CPQ for Administrators

Table of Contents

This section explains how to configure Conga Configuration & Pricing. This section describes the administrative tasks to be performed to enable the features in CPQ and covers the most common use cases for CPQ administration.

Topic	Description
What's Covered	This section walks the administrator through the process of CPQ administration. It provides conceptual information, step-by-step instructions, and use cases for CPQ administrative tasks. This section describes the workflows for configuring products, defining the price for configured products, defining product and pricing rules and so on.
Primary Audience	Administrator
IT Environment	Refer to the latest *Conga CPQ Release Notes* for information on System Requirements and Supported Platforms.
Updates	For a comprehensive list of updates to this section for each release, see the What's New in CPQ Documentation topic.
Other Resources	Refer to *Conga CPQ Release Notes* for information on system requirements and supported platforms, new features and enhancements, resolved issues, and known issues for a specific release.

This section describes the following tasks:

- Create Products
- Manage Bundles
- Create Categories and Hierarchy
- Manage Attributes
- Manage Rules
- Create Price Lists
- Manage Pricing Rules
- Configure Quote/Proposal
- Manage Promotions
- Manage Assets and Asset Line Item

Before using CPQ, you must be familiar with the following:

- Basic Salesforce administration
- Salesforce and Conga terms and definitions

Select one of the following topics for more information:

- Getting Started with CPQ
- Setting up Salesforce Org with Conga Cloud Instance for CPQ
- Managing Products
- Managing Product Rules
- Managing Pricing
- Managing Quotes or Proposals
- Managing Guided Selling
- Managing Promotions
- Configuring Assets
- Configuring Service CPQ
- Configuring the Cart
- Managing Translation of Fields and Values in Different Language
- Customizing CPQ Using Callbacks
- Configuring Admin Settings
- Frequently Asked Questions
- Conga Contact Support
- System Fields

Getting Started with CPQ

Select one of the following topics for more information about Conga CPQ.

- Before You Install CPQ
- Installing CPQ Packages
- Registering for Conga Upgrade Program
- Logging in to CPQ
- Navigating the CPQ Admin User Interface
- About Permission Sets
- CPQ Package Objects
- Configuring Post-Installation Settings
- Upgrading CPQ
- Creating an Opportunity in Salesforce
- Creating a Shopping Cart Experience
- Running Maintenance Jobs
- About TurboEngines
- Using Shield Platform Encryption
- Enabling CPQ for Partner Community Users

Before You Install CPQ

This section explains the tasks to be completed before you install CPQ.

To enable Salesforce CRM content

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

Installing 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 Conga 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 Advanced Approvals you do not need to install Conga Approvals or Conga Quote Approvals. 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 Conga products.

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

> ✅ The Conga Upgrade Program is an automated tool that upgrades packages available in your Salesforce org (Production or Sandbox) to the latest versions. In addition, It ensures all the Conga published managed packages are on the latest versions for the registered orgs. To register your org for upgrade program, see Registering for Conga Upgrade Program.

Install the packages in the following order.

> ⚠ The Conga Base Library provides a shared library of reusable components that other Conga packages may use. All CPQ and Contract Management customers must install this package in order to use Conga CPQ and Conga CLM products. In addition, beginning in the Spring '20 release, Conga is adding a User Experience Analysis component that collects product usage data. No action is required on your part and there is no impact to any existing product features, system functionality, or configured workflows. No personally identifiable information (PII), nor any other proprietary, confidential, or sensitive data will be collected. For more information, see CPQ on Salesforce Spring 2020 Release Notes.

Order	Package	Install Center tab to access the package	Required?	Install this package only if you also have...
1	Conga Base Library	Contract Management	Y	
2	Conga Contract Lifecycle Management	Contract Management	Y	
3	Conga Quote Management	CPQ	Y	
4	Conga Configuration & Pricing	CPQ	Y	
5	Conga Quote Configuration Integration	Integrations	Y	
6	Conga CPQ Api	CPQ	Y	
7	Conga CLM Configuration Integration	Integrations	N	Contract Management
8	Conga Quote CLM Integration	Integrations	N	Contract Management
9	Conga Quote Asset Integration	Integrations	Y	Renewals
10	Conga CPQ Setup	CPQ	Y	

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

Order	Package	Install Center tab to access the package	Required?
1	Conga Approvals	Advanced Approvals	N
2	Conga CPQ Approvals	Advanced Approvals	N

Order	Package	Install Center tab to access the package	Required?
3	Conga Quote Approvals	Advanced Approvals	N

You can additionally install the following packages if you are using the Deal Maximizer:

Order	Package	Install Center tab to access the package	Required?
1	Conga Deal Maximizer Setup	Deal Maximizer	N
2	Conga Deal Maximizer	Deal Maximizer	N
3	Conga CPQ Maximizer	Deal Maximizer	N

If you are using Adobe Sign to electronically sign any type of document associated with quotes, you must install the following packages:

Order	Package	Install Center tab to access the package	Required
1	Adobe Sign	NA	Y
2	Conga Quote EchoSign Integration	Echosign Integration	Y

If you are trying to integrate and synchronize your CPQ data with the Conga Platform, you must install the following:

Order	Package	Install Center tab to access the package	Required
1	Conga CPQ (RLS)	Conga CPQ (RLS)	Y

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

> If, during installation, you see an "Approve Third-Party Access" prompt, please select the **Yes, grant access to these third-party web sites** check box and click **Continue**.

To install the CPQ managed packages

1. Go to the **Get Started** > **Install Center** tab on the Conga Community Portal.
2. In the **My Packages** navigation link, click **CPQ**. A list of CPQ-related packages is displayed.
3. 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.
4. To install any specific version, click Details on the package tile > select the required version > 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 Conga.
7. Select the profile for which you want to install the package. Conga 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.

   > ⓘ If a permission is not enabled for a profile, you can apply Conga-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 About Permission Sets.

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.

Registering for Conga Upgrade Program

You can register your org for Conga Upgrade Program to auto-upgrade Conga managed packages. Conga Upgrade Program supports upgrading from the Spring '18 release or later to the latest major release (*n*) or previous two major releases (*n* – 1 or *n* – 2).

To register your org with Conga Upgrade Program for auto-upgrade

This section provides a high-level flow of the org registration process and early feedback programs. Refer to *Conga Upgrade Program documentation* for a comprehensive overview of features.

1. Launch the Conga Upgrade Program's registration page.
2. Select your Salesforce org type (**Production** or **Sandbox**). The Salesforce login page is displayed.
3. Enter your credentials and click **Login**. You are redirected to the connected app called **Conga Push Upgrade CA**, which prompts you to allow access to the org information.
4. Click **Allow**. The org registration form is displayed.
5. Click **Set Package Access Control** to set the accessibility for Conga packages to be upgraded with Conga Upgrade Program.
6. Click **Submit**.
    The **Early Feedback** pop-up is displayed. In addition, Conga sends an email notification to the system administrator on successful registration with a list of packages (including dependent packages, if any) handled in the upgrade process.

    > ⚠ The early feedback program executes a test suite to check if the customizations in your org are compatible with the upgrade. For more information, see the Conga Automation Early Feedback program.

7. Click **Try Now**. A **Conga Automation Dashboard** is displayed with a **Quick Guide** pop-up. Click **Yes** to go through the quick tour of the dashboard.
8. Click **Click to see the test t**o view a list of tests to be executed for your org, A **List of Tests** pop-up is displayed.

9. Enter your Email ID and Organization Name. You can enter multiple email ids.
10. Click **Trigger Welcome Email**. Conga sends a welcome email to the email ids entered in the previous step with a list of tests performed in the early feedback program.
11. Click **Proceed**. A **Conga Services Test Automation Execution Page** is displayed.
12. Click the checkbox next to the Job ID and click **Begin Test Execution** to run the predefined automated test suite against your org.

For the latest information on Conga Upgrade Program, see Conga Upgrade Program documentation.

Logging in to CPQ

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

> ⓘ 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 Conga.

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 CPQ (for example, if you are doing user acceptance testing), go to http://test.salesforce.com/ instead.
2. From 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**.
4. Navigate to **Conga Product Setup** or **Cona Pricing Setup** to access the Product or Pricing setup tabs, respectively.

   - In Salesforce Classic: Click the App Menu (Conga Product Setup ▼).
   - In Salesforce Lightning Experience: Click the App Launcher (⠿).

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

> ⓘ For using Lightning Experience in your Salesforce org, you must enable My Domain in your organization. To know how to enable My Domain in your org, refer to the Salesforce Help page My Domain.

Navigating the CPQ Admin User Interface

As an admin, you can use the CPQ Admin console to manage products and its related components such as options, attributes, and categories. The admin console helps you to create a product, associate the product with the categories, price lists, and price list items on a single page. You can define various settings in the Application Management section that are required to use CPQ.

> ⓘ Conga recommends you use CPQ Admin UI to create a product and associate the product with products, categories, price lists, and price list items. Alternatively, you can also use the classic Salesforce UI.

You can navigate to the **Conga CPQ Admin** app, and click the **CPQ Admin** tab to access the admin console. The homepage of the CPQ Admin user interface consists of the **Products**, **Catalog**, and **Pricing** tabs on the primary toolbar and the **search** and **filter** options on the secondary toolbar. You can configure the fields that are displayed on this page. For more information, see Configuring the CPQ Admin Landing Page Fields and Configuring the Product Creation Screen Fields.

A description of each of these user interface elements is provided as follows:

- The **Products**, **Catalog**, and **Pricing** tabs enable you to add and manage products, product associations, and configure pricing. After you have selected any feature from the primary toolbar, you see a secondary toolbar that enables you to configure subsequent flows.
- The **Products** homepage consists of a list of products configured for the organization. The following tasks are available on the secondary toolbar.
  - Search filters enable you to define search criteria to search products. For more information about searching configured products, see Configuring Product Visibility.
  - Click **New Product.** A product creation page is displayed. For more information about creating products, see Creating Products.

- From the menu icon (⋮) next to the **New Products** button, click **New from Clone** to create a new product by cloning an existing product configuration. For more information about cloning a product, see Cloning Products.
- You also search for the product on the homepage. See Navigating the Product Page for more information.

- The **Catalog** homepage consists of a list of categories.
  - Click field headers to sort the list on the Categories homepage to sort the column.
  - Enter the name of the product that you want to search in the search window. The category that matches the search is displayed.
  - Click **New Category** to navigate to the category creation page. For more information about creating categories, see Creating Categories.
- The **Pricing** homepage consists a list of price lists.
  - Click field headers to sort the list on the Pricing homepage to sort the column.
  - Enter the name of the price list that you want to search in the search window. The price list that matches the search is displayed.
  - Click New PriceList to navigate to the price list creation page. For more information about creating price lists, see Creating Price Lists.
- Click **Showing** *<number>* **Records** to select the number of products or category records that you want to display in the homepage. Click arrows **< >** to move to the next page.
- The **More** icon (⋮) on the primary toolbar enables you to execute maintenance jobs and configure various Application Management settings required to use CPQ.

Configuring 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 Product Creation page as well.

Configuring 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** column.
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, CPQ saves the data that you enter in the previous screen.

Configuring the Tooltips

You can display additional details of product records on the different pages of CPQ Admin. When you hover the cursor on the product name, a tooltip appears that displays more details about the product. By default, you can see the **Product Code**, **Product Family**, and **Product Description** fields on the tooltip. You can configure the tooltip to display different fields.

The tooltips are displayed on the following pages:

- Product Details page
- Product Groups Details page
- Price List Items Tab

To configure tooltips

1. Navigate to **Setup > Customize > Products > Field Sets**. This field set displays the fields on the tooltip.
2. Click **Edit** for **Product Compact Fields**.
3. Drag and drop the desired fields in the **In the Field Set** column. You can add a maximum of 3 fields only. You must remove the unwanted fields.
4. Click **Save**.

For more information on the Product Details page, refer to Navigating the Product Page.

Configuring Custom Fields on Price List Related Pages

You can add custom fields to price list related pages through the CPQ Admin UI. You can add and remove the fields displayed on the pages where you can configure and manage pricing records. You can also add custom fields with **Text**, **Number**, **Date**, **Picklist**, and **Lookup** datatype.

To configure the display of the pricing fields

Perform the following steps to configure the fields:

1. Navigate to **Setup > App Setup > Create > Objects**.
2. Selected one of the following objects where you want to add or remove pricing fields. Click **Edit** next to Field Set based on the page you want to configure.

Object	Field Sets	Description
Price List	Price List List Fields	Configure this field set to add or remove fields from the Manage Price Lists landing page.
	Price List Detail Fields	Configure this field set to add or remove fields from the Price List Detail page.
Price List Item	Price List Item Detail Fields	Configure this field set to add or remove fields from the Price List Item Detail page.
	Price List Item Product Detail Fields	Configure this field set to add or remove fields from the **Pricing** tab on the Product Detail page.
Price Matrix Entry	Price Matrix Entry Detail Fields	Configure this field set to add or remove fields from the **Matrices** tab page.

> ⓘ You can display fields that store IDs such as **External ID** or **Created By ID**, and fields like **Delete**. However, you cannot save any changes you make to them as they are not editable.

3. Drag and drop the required fields in the **In the Field Set** column.
4. Click **Save**.

For more information on how to configure product pricing, refer to Managing Pricing.

About Permission Sets

CPQ has a standard set of permissions that contain the access rights to fields and objects available in CPQ. These permission sets define whether the User can read, create, edit, and delete the records of an object and its fields. These permissions and the impact they have on the User are described below:

Permission	Description
Read	Allows the User to view the existing records created for the object.
Create	Allows the User to create a record for the object.
Edit	Allows the User to edit the existing record of the object.
Delete	Allows the User to delete a record of the object.
View All	Allows the User to read and create records for an object.
Modify All	Allows the User to edit and delete records for an object.

The following table lists the standard permission sets available in CPQ and their usage.

Permission Sets	Usage
Apttus CPQ Admin	This permission set must be assigned to system administrators.
Apttus CPQ User	This permission set can be assigned to any user who is not a system administrator.

There are different sets of permissions for an administrative role and a regular user role. You must assign the permission sets to the Users accordingly.

To view the permission sets

You can find the standard CPQ permission sets by following the steps below.

1. Go to **Setup** > **Administration Setup** > **Managed Users** > **Permission Sets**.
2. Click **Apttus CPQ Admin** or **Apttus CPQ User**.
3. Click **Object Settings** which contains a list of all the objects with their permissions.
4. Click the object names to view the fields of the objects and their permissions.

> ⚠ If you are using permission sets that are not standard to CPQ, you must update permissions for new fields manually based on **Apttus CPQ Admin** and **Apttus CPQ User** permission sets.

CPQ Package Objects

CPQ is comprised of multiple packages as mentioned in the installation section and this page lists out the objects inside each package.

Conga Base Library

Conga Base Library provides a shared library of reusable components that the other Conga packages can use. The following table list the objects in Conga Base Library.

Object	Purpose of the Object
Apttus UI Component Configuration	Object Records holds stored configurations for configurable higher-order components
Apttus UI Component Provider	The object records store the information about Providers of all types that are implemented by consuming applications
Configuration Action Provider (Deprecated)	Junction Object to Correlate Configurations to Action Provider(s)
Configuration Action Provider 2	Junction Object to Correlate Configurations to Action Provider(s)

Conga Configuration & Pricing

Conga Configuration & Pricing gives you complete visibility and control of the quote-to-order phase of your sales process. Beginning in an opportunity, a user can select products or services, configure and price them for simple or highly complex scenarios, and produce high-quality customer-facing quotes and proposals.

Object	Purpose of the Object
Account Billing Summary	Represents the billing summary for an account.
Account Location	Represents a location for an account.
Adjustment Line Item	Represents an adjustment to the price of a product/service.
Agreement Price Rule	Represents a special pricing rule associated with an agreement or an asset.
Agreement Price Tier	Represents a single entry in an agreement price tier.
Application Feature	An object to contain features in the Applications, where Applications are Conga products like CPQ, CLM etc.
Application Setting	Object to contain metadata about application settings.
Applied Expression Info	Represents expressions applied to the configuration.
Applied Rule Action Info	Rule actions that are applied to the configuration.
Applied Rule Info	Keeps track of rules applied to a configuration.
Asset Attribute Value	Represents the attribute values for a product class.
Asset Attribute Value History	Represents the attribute values for a product class.
Asset Line Item	Represents a product or service asset line item.
Asset Line Item History	Represents a product or service asset line item history.

Object	Purpose of the Object
Asset Transaction History	Represents a transaction that records the change to an asset.
Asset Usage Price Tier	Represents a single entry in an asset usage price tier.
Asset Usage Price Tier History	Represents a single entry in an asset usage price tier.
Attribute Group Translation	This object holds the translated values of attribute group object fields for all the languages.
Attribute Value Matrix	Represents the information about attribute value compatibilities or incompatibilities.
Attribute Value Matrix Entry	Represents a row (entry) in an attribute value matrix.
Batch Job (CPQ)	Represents an individual batch job.
Batch Job (CPQ) Parameter	Represents a batch job bind parameter.
Billing Plan	Represents a billing plan associated with a product or service that is part of a quote or order.
Billing Plan Item	Represents a billing entry in a billing plan of a quote or order.
Billing Plan Template	Represents the template to create a billing plan for a quote or order.
Billing Plan Template Item	Represents a billing plan template item associated with the quote or order.
Billing Preference	Represents a functional preference for billing purposes of a quote or order
Bundle Component View	Keeps the list references of component products related to the bundle and sub-bundle.
Cart Clone Info	
Category	Represents a category.

Object	Purpose of the Object
Category Hierarchy	Represents a category hierarchy.
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.
Charge Group	Represents a group of charge types associated with a product or service.
Charge Group Member	Represents the mapping between a charge group and the charge types associated with the group.
Charge Type	Represents a type of charge for a product or service.
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.
Constraint Rule	Captures information about configuration constraint rule.
Constraint Rule Action	Captures information about rule actions. Rule actions are applied only when the rule conditions are satisfied.
Constraint Rule Action Expression	
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.
Cost Adjustment	Represents an adjustment to the cost of a product/ service.
Cost Item	Represents a product entry in a Cost Model.
Cost Line Item	Represents cost associated with base price of line item.

Object	Purpose of the Object
Cost Model	Cost Model represents cost model of a FES or Factory.
Cost Type	Represents cost hierarchy definition.
Custom Message	Used to provide translation for messages used in custom objects.
External Order Adjustment Item	Represents an adjustment to the price of a product/service.
External Order Summary	Represents an external order summary associated with an account.
External Order Summary Item	Represents a product or service line item associated with an external order summary.
Favorite Configuration	
Feature	Represents a single feature in a feature set.
Feature Set	Represents a set of features shared by products.
Field Expression	Represents a field expression.
Footnote	Represents a single footnote in a document.
Formula Field (CPQ)	Represents a formula field used in price dimensions and rules.
Frequency Conversion Rate	Represents the conversion rates used to convert pricing from one frequency to another.
G/L Account Map Entry	Represents a single entry in a G/L account map rule.
G/L Account Map Rule	Represents a G/L account map rule to determine G/L account numbers for a product or service.
Guided Question	Represents a question in a guided interview.

Object	Purpose of the Object
Guided Search Rule Entry	Represents a single entry in a guided search rule.
Guided Search Rule Filter	Represents a single search filter in a guided search rule entry.
Guided Selling Rule	Represents a rule to search for products based on a guided interview.
Help Doc	Object holds help documentation.
Incentive	Represents an incentive associated with one or more products.
Incentive Adjustment Item	Represents an adjustment to the price of a product/service.
Incentive Benefit Data	Represents a single benefit data row associated with the incentive benefit.
Incentive Coupon	Represents a coupon associated with an incentive.
Incentive Group	Represents a group of incentives.
Incentive Limit	Represents a single limit associated with an incentive.
Incentive Limit Data	Represents a single limit data row associated with the incentive limit.
Incentive Loyalty Enrollment	Represents a single customer enrollment into an incentive loyalty program.
Line Item	Represents a product or service line item.
Line Item Rollup	Holds rollup summary custom fields.
Loyalty Point Accrual Adjustment Item	Represents loyalty points accrual adjustments for a product/service.

Object	Purpose of the Object
Loyalty Point Accrual Summary	Represents a summary of loyalty points accrued for the account or contact.
Milestone	Represents milestones associated with other business objects.
Milestone Adjustment Item	Represents an adjustment to the price of a product/service.
Milestone Template	Represents a template for creating milestones.
Notification Feed	A generic object to contain Notification Feeds generated by any object for any page.
Order	Represents an order associated with an account.
Order Adjustment Line Item	Represents an adjustment to the price of a product/service.
Order Fulfillment	Represents an order fulfillment associated with an account.
Order Fulfillment Line Item	Represents a fulfillment line item associated with one or more orders.
Order Incentive Benefit Data	Represents a single benefit data row associated with the incentive benefit for the order.
Order Incentive Limit Data	Represents a single limit data row associated with the incentive limit for the order.
Order Line Item	Represents a product or service line item associated with an order.
Order Loyalty Point Accrual Item	Represents loyalty points accrued for a product/service.
Order Pocket Adjustment Line Item	Represents a pocket price adjustment to the price of a product/service.

Object	Purpose of the Object
Order Product Attribute Value	Represents the attribute values for a product class in an order line item.
Order Rollup Data	Represents the dynamic rollup data for use in criteria.
Order Tax Breakup	Represents the tax breakup of a product/service
Order Usage Price Tier	Represents a single entry in an order usage price tier.
Payment Term	Represents a payment term associated with a quote, order or an agreement.
Permission Set Relationship	
Price Breakup	Represents the price breakup of a product/service.
Price Dimension	Represents a price criteria dimension.
Price List	Represents a price list that contains the list of products the organization sells.
Price List Category	Represents the categories associated with a price list.
Price List Item	Represents a product entry in a price list.
Price Matrix	Represents a price matrix associated with a price list entry.
Price Matrix Entry	Represents a single entry in a price matrix.
Price Rule	Represents a single rule in a price ruleset.
Price Rule Entry	Represents a single entry in a price rule.
Price Ruleset	Represents a rule set that contains the list of rules to modify prices of products the organization sells.

Object	Purpose of the Object
Product Attribute	Represents a single attribute in a product attribute group.
Product Attribute Group	Represents a product attribute group that contains attributes shared by products.
Product Attribute Group Member	Represents the association between a product and an attribute group.
Product Attribute Matrix View	Represents a denormalized (materialized) view of product attribute and matrices.
Product Attribute Rule	Represents the rule definition of a product attribute value.
Product Attribute Rule Action	Represents a single action in a product attribute rule.
Product Attribute Rule View	Represents a denormalized (materialized) view of product and attribute rule.
Product Attribute Value	Represents the attribute values for a product class.
Product Classification	Represents a product classification which is a member of the list of products associated with a classification.
Product Configuration	Represents a product configuration.
Product Constraint	Represents a set of constraints a product must satisfy before it can be added to the product configuration.
Product Constraint Entry	Represents a single entry in a product constraint.
Product Constraint View	Creates product and constraint rule view table based on the definition of product scope in constraint rule condition.
Product Default Rule	Represents a rule to default products, quantity, and term.
Product Default Rule Filter	Represents a single filter in a product default rule.

Object	Purpose of the Object
Product Default Value	Represents the default values for a product.
Product Feature	Represents the association between a product and a feature set.
Product Feature Value	Represents the feature values for a product and its associated feature set.
Product Filter View	Represents the filter values for each product in a category.
Product Footnote	Represents the footnote available in a product or option.
Product Group	Represents a group of products.
Product Group Member	Represents a single product in a product group.
Product Hierarchy View	Represents a denormalized (materialized) view of product and classification hierarchies.
Product Information	Represents information associated with a product or a classification.
Product Option Component	Represents a product option component which is a member of the list of options associated with a product option group.
Product Option Group	Represents a product option group which is a member of the list of option groups associated with a product.
Product Option Price	Represents the price for an option or component.
Product Translation	This object holds the translated values of product object fields for all the languages.
Published Favorite	A junction object between Favorite Configurations and Price Lists to support many to many relationships.

Object	Purpose of the Object
Related Account Location	Holds the relationships between account locations.
Related Asset Line Item	Holds the relationships between asset line items.
Related Incentive	Holds the relationships between incentives.
Related Price List Item	Holds the relationships between price list items.
Related Product	Related products are for up-sell, cross-sell or related products (search helper).
Revenue Recognition Policy	Represents a revenue recognition policy associated with a product or service.
Revenue Split Policy	Represents an order revenue split policy associated with a product or service.
Revenue Split Policy Entry	Represents a single entry in a revenue split policy.
Rollup Data	Represents the rollup data for use in criteria.
Saved Search	Represents a saved search criteria.
Search Attribute Value	Holds product search parameters and default values for line items.
Search Filter (CPQ)	Represents a search filter.
Service Location	Represents a service location.
Setting Group	Object to group settings into logical sets.
Smart Search Activity History	Contains a history of the actions occurring during the search process when connecting to the Azure system (activation, deactivation, sync).
Summary Group	Represents a summary of line items.
Tax Breakup	Represents the tax breakup of a product/service.

Object	Purpose of the Object
Tax Certificate	Represents a tax certificate.
Tax Code	Represents a tax code associated with a product or service.
Temp Display Column	Holds information about columns displayed in the cart view.
Temp Filter	Helps in the binding of search filter values.
Temp Incentive Benefit Data	Represents a single benefit data row associated with the incentive benefit for the product configuration.
Temp Incentive Limit Data	Represents a single limit data row associated with the incentive limit for the product configuration.
Temp Object (CPQ)	A container for temporary objects used by the application.
Temp Renew	Temporary object for capturing user input to create asset line item during the amendment/renewal process.
Temp Renew Asset Group	Temporary object to store renew asset group processed by RenewOpportunityJob.
Temp Renew Asset Line Item	Temporary object to store renew asset line item processed by RenewOpportunityJob.
Temp Rollup Data	Represents the dynamic rollup data for use in criteria.
Totaling Group	Represents a totaling group.
Transfer Price	Represents association of source price like factory price with source cost like FES source cost.
Usage Price Tier	Represents a single entry in a usage price tier.

Conga Contract Lifecycle Management

The Conga Contract Lifecycle Management Suite gives you complete visibility and control of your contract process. This suite is easy to deploy and use; you can immediately gain benefits such as process control, high-risk management, and immediate ROI.

Object	Purpose of the Object
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.
Admin	Store admin preferences and metadata.
Agreement	Represents an agreement in the system.
Agreement Action Condition	Represents conditions to enable agreement actions.
Agreement Clause	Represents a clause associated with an agreement.
Agreement Document	Links to agreement documents that are not stored in Conga/Salesforce repository. These could be URLs to files stored on a file server or document management system behind the client's firewall.
Agreement Document Output Format	Holds default output format preferences by user profile and agreement type for use in the agreement document generation process.
Agreement Explorer	A reusable, unique combination of Field Sets and Filter Sets.
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.
Agreement Line Item	Represents a product or service line item associated with an agreement.
Agreement Lock	Represents a lock on the agreement.
Agreement Protection	Specifies protection settings for agreements generated by the Conga Contract Wizard and maintained by the Conga Contract Author.

Object	Purpose of the Object
Agreement Rule	Specifies a rule used to evaluate various agreement actions such as submit request.
Agreement Rule Condition	Specifies a condition used to evaluate an agreement rule.
Agreement Term Exception	Term Exceptions associated with a given Agreement.
Async Merge Call	Holds asynchronous merge calls.
Content Event	Holds content events for subscribers to handle.
Cycle Time Field	Holds the fields monitored for cycle time data capture.
Cycle Time Field Data	Holds the data captured on fields for cycle time reporting.
Cycle Time Group	Holds the groups monitored for cycle time data capture.
Cycle Time Group Data	Holds the data captured on groups for cycle time reporting.
Doc Assembly Component	Represents a single component in a document assembly rule.
Doc Assembly Rule	Represents a single rule in a document assembly ruleset.
Doc Assembly Ruleset	Represents a rule set that contains a list of rules to determine the sections to include in a dynamic template.
Document Agreement Clause	Agreement clauses in an agreement document.
Document Version	Represents a document version.
Document Version Detail	Holds the version details of a document.
Formula Field (Comply)	Represents a formula field used in agreement rules.
Merge Event	Holds document merge events for subscribers to handle.
Merge Event Detail	Holds document merge event details.

Object	Purpose of the Object
Query Template	Query Template for building SOQL queries and filters.
Query Template Filter	Filter expression used in a query.
Query Template Qualifier	Filter used to select a query template.
Related Agreement	To identify the different types of relationships between agreements.
Retention Policy	Holds object retention policies.
Search Filter (Comply)	Represents a search filter.
Temp Object (Comply)	A container for temporary objects and attachments.
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.
Template Clause Reference	Represents a clause reference in a template or a library.
Template Clause Reference Version	Holds Template Clause References for this version.
Template Datasource Filter	Represents a data source filter associated with a template.
Template Dynamic Section	Represents a dynamic section in a template.
Template Dynamic Section Version	Holds the template dynamic section details for a given template version.
Template Version	Holds various versions of this template and is a Master-detail to Template.
Term Exception	Term Exceptions master table.
Wizard	A runtime instance of the wizard design.

Object	Purpose of the Object
Wizard Design	Holds all the wizard design records.
Wizard Input Control	The individual question, instruction statements, and associated field attributes.
Wizard Rule	Individual rules which are applied within a ruleset.
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.
Wizard Runtime Input	Input Control instance within a runtime wizard.
Wizard Step	The step that is designed and used in Wizard designs.

Conga Quote Management

Conga Quote Management allows you to create customized proposals including pricing and sales content. The right pricing and content is automatically selected and assembled based on business rules so that the sales representative can quickly get a compelling sales proposal to the prospect.

Object	Purpose of the Object
Document Collate Info	
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.
Proposal Line Item	Represents a product or service line item associated with a quote/proposal
Proposal Request	Approval requests for a proposal
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

Conga Quote Configuration Integration

Conga Quote and CPQ Integration integrates Conga Quote/Proposal Management with Conga CPQ. It enables the user to select products or services, configure and price them for simple or highly-complex scenarios, and produce high-quality customer-facing quotes and proposals.

Object	Purpose of the Object
Proposal Adjustment Line Item	Represents an adjustment to the price of a product/service.
Proposal Footnote	Represents the footnote available in a quote/proposal.
Proposal Incentive Benefit Data	Represents a single benefit data row associated with the incentive benefit for the quote/proposal.
Proposal Incentive Limit Data	Represents a single limit data row associated with the incentive limit for the quote/proposal.
Proposal Location	Represents the location available in a quote/proposal.
Proposal Product Attribute Value	Represents the attribute values for a product class in a proposal line item.
Proposal Rollup Data	Represents the dynamic rollup data for use in criteria.
Proposal Summary	Represents the summary of a product or service line item in a proposal.
Proposal Summary Group	Represents a summary of line items associated with a quote or proposal.
Proposal Tax Breakup	Represents the tax breakup of a product/service.
Proposal Usage Price Tier	Represents a single entry in a quote/proposal usage price tier.

Conga CLM Configuration Integration

Conga Contract and CPQ Integration integrates Conga Contract Lifecycle Management with Conga CPQ. It enables the user to select products or services, configure and price them

for simple or highly-complex scenarios, and produce high-quality customer-facing agreements.

Object	Purpose of the Object
Agreement Adjustment Line Item	Represents an adjustment to the price of a product/service.
Agreement Footnote	Represents the footnote available in an agreement.
Agreement Incentive Benefit Data	Represents a single benefit data row associated with the incentive benefit for the agreement.
Agreement Incentive Limit Data	Represents a single limit data row associated with the incentive limit for the agreement.
Agreement Product Attribute Value	Represents the attribute values for a product class in an agreement line item.
Agreement Rollup Data	Represents the dynamic rollup data for use in criteria.
Agreement Summary	Represents the summary of a product or service line item in an agreement.
Agreement Summary Group	Represents a summary of line items associated with an agreement.
Agreement Tax Breakup	Represents the tax breakup of a product/service.
Agreement Usage Price Tier	Represents a single entry in an agreement usage price tier.

Configuring Post-Installation Settings

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.

Basic Post-Installation Configuration

These settings are mandatory settings in order for CPQ app to run.

Configuring Proposal System

1. Go to **Setup > Custom Setting > Proposal System Properties > Manage> New**
2. Refer to the following image and enter the values in the System property fields. (The values may vary according to your CPQ instance.)
3. Click **Save.**

Configuring System Properties

1. Go to **All tabs> Config Setting > System Properties**
2. Enter values in the fields.
3. Click **Save.**

Configuring Flow

1. Go to **All tabs> Config Setting > Flow Settings**

2. Enter the values in the fields. Refer to the image below for an example.

> ⓘ Do not use special characters in the name.

3. Click **Save**.
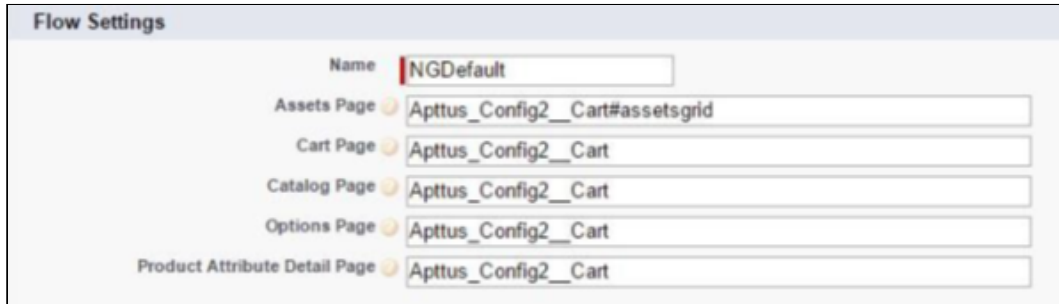
Configuring Custom Settings

1. Go to **Setup > Develop > Custom Setting > Config Select Products Settings > Manage**.
2. Find *Primary Settings*, Click **Edit**.
3. Enter a value in **Carousel Default Icon** ID (If you don't have any: create any notes & attachment, and get ID of that and enter here)
4. Click **Save.**

Running Custom Setting Maintenance Job

You must execute the Custom Setting Maintenance Job for all the flows after defining Custom Settings and when you define or update fields in Config Settings. For more information, refer to Running Maintenance Jobs.

Advance Post-Installation Configuration

- 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 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.

Configuring Application Management Settings

This section describes how you can configure Application Management settings.

The Application Management Settings 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 CPQ settings.

Navigate to **Admin Console** and from the ⋮ drop-down list, click **Application Management**. Click **Load Defaults** to bring the custom settings for the default flow in the Application Management.

Using the **Load Defaults** functionality on the application management will auto-create the system settings and you must go with this approach.
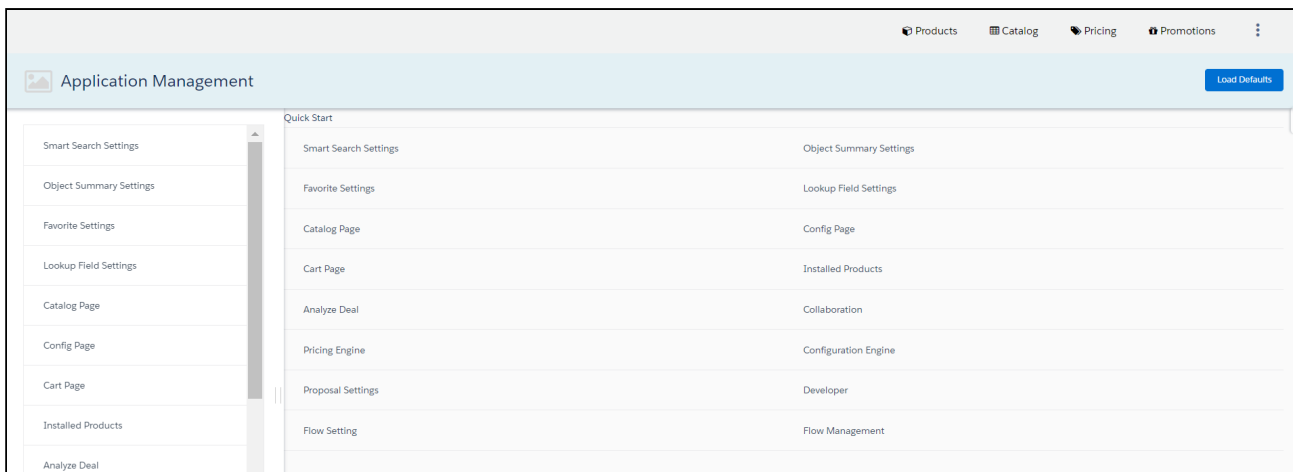
> ⚠ You must execute Custom Settings Maintenance Job after you update any Custom Settings. The changes you make are only effective if you execute the maintenance job. Otherwise, all the computations are processed with stale values. Refer to the Running Maintenance job topic.

> You can use the link given next to the **Save** and **Cancel** button on every setting page.

Favorites Settings under Application Management lets you manage the favorite product related settings including the default image, the columns in the favorite dialog, and enabling or disabling the favorite settings using the slider. When you disable this setting, the **Save as Favorite** option is not displayed in the cart. For more information on the configuration part, refer to Configuring Favorite Settings.

Lookup Field Settings under Application Management lets you manage the name of the record, an object of the lookup field, field name of lookup field, display columns, and filter criteria. For more information on the configuration part, refer to Configuring Lookup Field Settings.

For Quote, Product, and Favorites objects, you can define "Object summary" fields, which are shown when the user clicks on the link on the catalog such as Product Name or Favorites name. For more information on the configuration part, refer to Configuring Object Summary Settings.
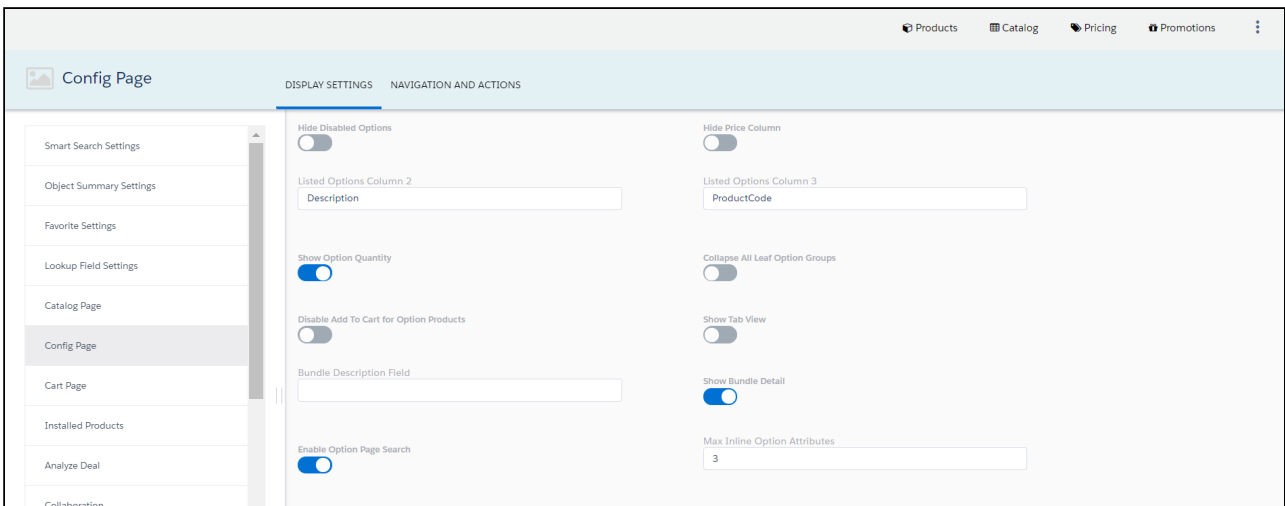
> ⓘ Disabled values are displayed with a dark gray background. CPQ does not allow you to access such values.

Note that the user with only System Administrator profile can access the Application Management functionality. If you log in through any other user, you will see this error message: *You do not have the level of access necessary to access the Application Management feature. Contact the administrator if access is necessary.*

> ⓘ The Lookup Field tab available at **CPQ Admin > Application Management >**
> **Developer tab** is not functional with the current release.

Optional Configurations

Adding the Application Settings Management Objects as Tabs

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

Object	Description
Application Feature	This is a parent tab.
	The application settings available in Config Settings and Quote/Proposal System Properties are 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 is a related list in the Application Feature Visualforce page. Using this page, you can create logical groupings for each of the settings as well as the Flow in which these settings are 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 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.

Object	Description
Application Setting	This is 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.
Application Feature	The name of the application feature, such as Cart Page.
Setting Group Name	The name you want to give to the Setting Group, such as Display Actions, Cart Views, and so on.
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.

Field	Description
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 records 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 Precision	This field specifies 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	Description
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 HideDefault Options, in Descriptionfield, 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.
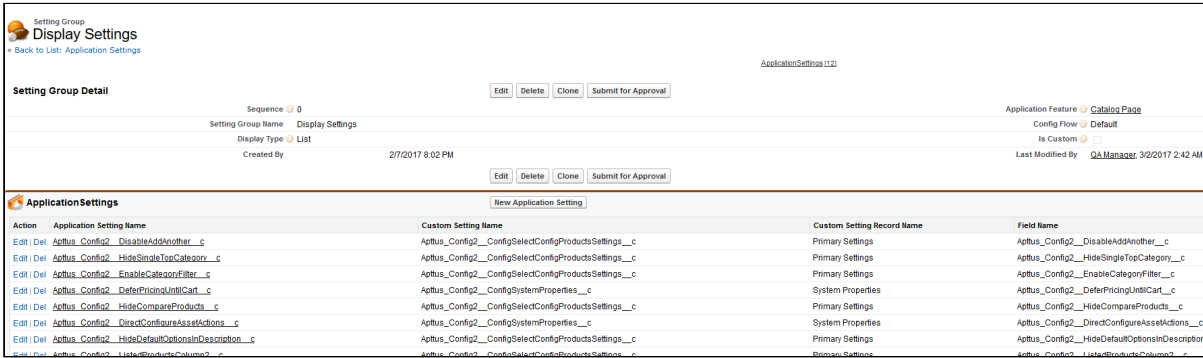
Configuring Smart Search Settings

1. On **App Menu**, click **Conga CPQ Admin** and then click the **CPQ Admin** tab. The new admin console is launched.
2. Click the more icon (⋮) on the top-right corner and select **Application Management**.
3. Click **Smart Search Settings**.
4. Configure the following details.

Setting	Description
SEARCH CONNECTION SETTINGS	
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.
PRODUCT FIELD WEIGHTAGE	
Flow	Select a flow and enter the following details. • Field Name • Typeahead • Weightage

Configuring Favorite Settings

1. On **App Menu**, click **Conga CPQ Admin** and then click the **CPQ Admin** tab. The new admin console is launched.
2. Click the more icon (⋮) on the top-right corner and select **Application Management**.
3. Click **Favorite Settings**.
4. Enter details in one or more of the following sections, based on your organization's requirement.

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 Configuration Column1	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.
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.
Favorites Category 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.
Listed Favorite Configuration Column2	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.

Configuring Catalog Page Settings

1. On **App Menu**, click **Conga CPQ Admin** and then click the **CPQ Admin** tab. The new admin console is launched.
2. Click the more icon (⋮) on the top-right corner and select **Application Management**.
3. Click **Catalog Page**.
4. Configure the following details.

Setting	Description
NAVIGATION AND ACTIONS	
Flow	Select a flow and enter the following details. • Display As: • Action Area: • Action Style Class: • Is Enabled: • Always Display:
DISPLAY SETTINGS	
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.
Hide Configure Action	Hides the Configure action for a Product on the catalog page.
Hide Listed Products Price Column	Hides the price information column in listed product section. By default, this is not selected.
Listed Products Column3	(optional) Type the API name of the product field to display in the listed products section. Example Value: ProductCode
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.

Setting	Description
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.
Hide Compare Products	Disables the Compare Products feature in the catalog page.
Hide Default Options In Description	Hides the default options listed below the product description.
Listed Products Column2	(optional) Type the API name of the product field to display in the listed products section. Example Value: ProductCode
Show Quantity Input	Displays the Quantity field on the catalog page.
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.

Configuring Cart Page Settings

1. On **App Menu**, click **Conga CPQ Admin** and then click the **CPQ Admin** tab. The new admin console is launched.
2. Click the more icon (⋮) on the top-right corner and select **Application Management**.
3. Click **Cart Page**.
4. Configure the following details.

Setting	Description
DISPLAY SETTINGS	

Setting	Description
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 Itemsisbased on the user selection sequence rather than the Bundle Options sequence.
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 length, the ellipsis is shown. The default value is 21.
Expand Bundles in Cart	Expands all attributes and options of a bundle product in the cart. ⓘ The pagination feature is not supported along with Expand Bundles in Cart.
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 dropdown 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. These fields are also used in the Cart View feature. For more information, see Viewing the Cart in Grid View.
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.
Show Recommended Products Cart View	Displays the recommended products component in the Cart Detail View page.
Show Attributes in Cart	⚠ This setting is deprecated.
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.

Setting	Description
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.
NAVIGATION AND ACTIONS	
Flow	Select a flow and enter the following details. • Display As: • Action Area: • Action Style Class: • Always Display:
CART LINE COLUMNS	
Flow	Select a flow and enter the following details. • Field Name ⓘ Select the required columns that should be displayed on the cart for the user. The **Renewal Adjustment Type** and **Renewal Adjustment Amount** adjustment types are applicable only for Asset Renewal action. • Is Editable • Style • Style Class • Header Style
TOTALS COLUMNS	
Flow	Select a flow and enter the following details. • Field Name • Is Editable • Style • Style Class • Header Style
PRICE RAMP	

Setting	Description
Flow	Select a flow and enter the following details. • Field Name • Is Editable • Style • Style Class • Header Style
USAGE PRICE TIER	
Flow	Select a flow and enter the following details. • Field Name • Is Editable • Style • Style Class • Header Style
TIERED PRICE	
Flow	Select a flow and enter the following details. • Field Name • Is Editable • Style • Style Class • Header Style
RELATED PRICE	
Flow	Select a flow and enter the following details. • Field Name • Is Editable • Style • Style Class • Header Style
ADJUSTMENT LINE ITEM	

Setting	Description
Flow	Select a flow and enter the following details. • Field Name • Is Editable • Style • Style Class • Header Style
MASS UPDATE	
Flow	Select a flow and enter the following details. • Field Name • Is Editable • Style • Style Class • Header Style
AGREEMENT PRICE RULE	
Flow	Select a flow and enter the following details. • Field Name • Is Editable • Style • Style Class • Header Style

Configuring Analyze Deal Settings

1. On **App Menu**, click **Conga CPQ Admin** and then click the **CPQ Admin** tab. The new admin console is launched.
2. Click the more icon (⋮) on the top-right corner and select **Application Management**.
3. Click **Analyze Deal**.
4. Configure the following details.

Setting	Description
NAVIGATION AND ACTIONS	

Setting	Description
Flow	Select a flow and enter the following details. • Action Label Name: • Display As: • Action Area: • Action Style Class: • Action Page: • Action Param: • Behavior: • Is Enabled: • Always Display:
COST LINE ITEM	
Flow	Select a flow and enter the following details. • Field Name: • Is Enabled: • Style: • Style Class: • Header Style:
LINE ITEM WATERFALL	
Flow	Select a flow and enter the following details. • Field Name: • Is Enabled: • Is Price Point:
LINE ITEM METRICS	
Flow	Select a flow and enter the following details. • Field Name: • Is Enabled: • Style: • Style Class: • Header Style:
SUMMARY METRICS	

Setting	Description
Flow	Select a flow and enter the following details. • Field Name: • Is Enabled: • Style: • Style Class: • Header Style:
SUMMARY WATERFALL	
Flow	Select a flow and enter the following details. • Field Name: • Is Enabled: • Is Price Point:

Configuring Flow Management 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.

1. On **App Menu**, click **Conga CPQ Admin** and then click the **CPQ Admin** tab. The new admin console is launched.
2. Click the more icon (⋮) on the top-right corner and select **Application Management**.
3. Click **Flow Management**.
4. Enter details in one or more of the following sections, based on your organization's requirement.

Setting	Description
Flow Name	Type a mandatory name for the flow.
AssetsPage	Type the Visualforce page name for assets. Valid values are: • SelectInstalledProducts • SelectInstalledProductsEn

Setting	Description
CartPage	Type the Visualforce page name for cart. Valid values are: • Apttus_Config2__CartDetailView • Apttus_Config2__CartView If you want to view the cart with the new Grid UI, name the cart page as *Apttus_Config2__Cart#cartgrid.*
CatalogPage	Type the Visualforce page name for product selection. Valid values are: • Apttus_Config2__SelectConfigProducts • Apttus_Config2__SelectConfigProductsFilterView
OptionsPage	Type the options page to capture the options for a bundle product. Valid values are: • Apttus_Config2__ConfigureBundle • Apttus_Config2__SelectConfigOptions • Apttus_Config2__SelectConfigOptionsTabView • Apttus_Config2__SelectConfigOptionsListView • Apttus_Config2__SelectConfigOptionsTreeView • Apttus_Config2__SelectConfigOptionsDetailView • Apttus_Config2__SelectBundleOptions ⚠ Do not use the *SelectBundleOptions* page from your drop-down list, it is no longer supported.
ProductAttributeDetailPage	Type the Visualforce page name to capture product attribute detail information. Valid value is: • Apttus_Config2__ProductAttributeDetail3

Configuring Configuration Engine Settings

1. On **App Menu**, click **Conga CPQ Admin** and then click the **CPQ Admin** tab. The new admin console is launched.

2. Click the more icon (⋮) on the top-right corner and select **Application Management**.
3. Click **Configuration Engine**.
4. Configure the following details.

Setting	Description
CONSTRAINT RULES	
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).
Defer Constraint on Option Selection	Select this setting to process the constraint rules on option selection on the Configuration Page. If the selected the constraint rules deferred until the Sales Rep clicks **Go to Pricing** or any other action button available on the Configuration page.
Defer Validation Check In Bundles	Defers the validation check in the bundle configuration step. This is selected by default.
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*.
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.

Setting	Description
Constraint Rule Execution Mode	Enter the mode to perform the constraint rule execution on the client-side or the server-side. By default, no value is specified and the system assumes the value as *Server*. Valid values are: • *Client* • *Server* • *CMS* This custom setting supports inclusion, exclusion, and validation rules on the client-side as well as the server-side. ⓘ In a Service CPQ flow, always execute constraint rules in Client mode.
ATTRIBUTES	
Cascade Shared Attribute Updates	Cascades the shared bundle attribute updates to options in the bundle. If this setting is unchecked, 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.
Populate Attribute Extensions	Select this setting to indicate that the attribute values are populated in the attribute extension objects.
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.
EXPRESSIONS	
Enable Field Expressions	Enables you to use Expression Builder for attributes, option groups, and options.

Configuring Developer Settings

1. On **App Menu**, click **Conga CPQ Admin** and then click the **CPQ Admin** tab. The new admin console is launched.
2. Click the more icon ([⋮]) on the top-right corner and select **Application Management**.
3. Click **Developer**.
4. Configure the following details.

Setting	Description
SYSTEM GLOBAL	
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.
CSS Override	Type the name of the static resource to override CSS in the catalog, attributes, options, and cart pages.
Auto Update Category View	Indicates whether auto incremental updates to category view are enabled.
Keep Abandoned Carts	Indicates whether abandoned carts should be kept. The default value is false (abandoned carts are moved to the recycle bin).
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.
FIELD PRECISION	
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.
Currency Field Precision	Type a number to specify decimal places for currency precision. The default currency precision is 2 decimal places.

Setting	Description
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.
FINALIZATION & APPROVALS	
Run Misc Finalization Task In Async Mode	Indicates whether the secondary finalization task (attributes, usage tiers, etc) should be run asynchronously
Auto Sync On Cart Approval	Automatically synchronizes the cart when approved and finalized.
Auto Finalize On Cart Approval	Automatically finalizes the cart when the quote gets approved.

Configuring Object Summary Settings

In order to show the summary information for the Proposal (cart header) and Product in the Salesforce Lightning Experience, a new custom setting named **Object Summary Settings** is now available.

1. On **App Menu**, click **Conga CPQ Admin** and then click the **CPQ Admin** tab. The new admin console is launched.
2. Click the more icon (⁝) on the top-right corner and select **Application Management**.
3. Click **Object Summary Settings**.
4. Configure the following details.

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,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*

Configuring Lookup Field Settings

1. On **App Menu**, click **Conga CPQ Admin** and then click the **CPQAdmin** tab. The new admin console is launched.
2. Click the more icon (⋮) on the top-right corner and select **Application Management**.
3. Click **Lookup Field Settings**.
4. Click **New Lookup Field**.
5. Configure the following details.

Setting	Description
LOOKUP FIELD SETTINGS	
Name	Specify the name for the custom setting.
Object Name	Select one from the dropdown menu. · Collaboration Request · Line Item · Product Attribute
Lookup Field Name	Select the name of the field from the dropdown menu. The values are populated based on the Object you select. This is the lookup field containing values such as *Enterprise*, *Standard*, and *Premium*.
Filter Criteria	Enter a valid expression to further filter down the search results for a lookup. The field size is 256 characters. 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.
Lookup Display Columns	Select the fields to be displayed on the lookup dialog box.
Lookup Record Limit	Click the ⚙ icon, enter the maximum number of records to query for the lookup field dropdown, and click **Save**. Default value is 200 to a maximum of 1000.

Setting	Description
Junction Object Name	Click the ⚙ icon, enter the name of the junction object which you would like to filter the lookup values, and click **Save**. This is a custom object that you can define to filter the values which do not exist on the Lookup Object itself.
Junction Field Name	Click the ⚙ icon, enter the name of a field on the junction object which you would like to filter the lookup field by, and click **Save**.
Junction Default Flag	Click the ⚙ icon, enter the name of a field API from the Junction object to determine the default value of a lookup field, and click **Save**. This field must be a checkbox.
Filter Criteria 2	❗ This field is not available in the CPQ Admin UI. You must use **Setup** > **Build** > **Develop** > **Custom Settings** > **Lookup Field Settings** to configure it.

Configuring Config 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.

1. On **App Menu**, click **Conga CPQ Admin** and then click the **CPQ Admin** tab. The new admin console is launched.
2. Click the more icon (⸽) on the top-right corner and select **Application Management**.
3. Click **Config Page**.
4. Configure the following details.

Setting	Description
DISPLAY SETTINGS	
Exclude Optional Products	Select this to exclude optional products when CPQ evaluated the bundle's structural requirements. When you enable this setting, CPQ excludes the optional products from the option group Min/Max criteria.

Setting	Description
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.
Listed Options Column 2	Type the API name of the product field to display in the listed options section.
Show Option Quantity	Displays the Quantity in the body of the options pages. This setting controls the SelectConfigOptionsDetailView and ConfigureBundle page.
Disable Add To Cart for Option Products	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.
Show Bundle Detail	Shows the bundle details in the options page.
Hide Price Column	Disables the price column from the options page.
Listed Options Column 3	Type the API name of the product field to display in the listed options section.
Collapse All Leaf Option Groups	Collapses all the leaf option groups in the Config Options Detail View page.
Show Tab View	Displays option groups as tabs instead of sections in the options page. You can enable the Show Tab View setting at a product level also. Select the check box on the product details page.
Bundle Description Field	Type the field names from the product object to be displayed as bundle description.
Enable Option Page Search	Select this setting to allow option search on the bundle configuration page. This setting enables a search bar on the Bundle Configuration page. The setting allows the user to search for options, attributes, and sub-bundles on the Configuration page.

Setting	Description
Max Inline Option Attributes	Specify the maximum number of inline option attributes to be displayed in the attribute group. The default value is 3.
NAVIGATION AND ACTIONS	
Flow	Select the flow you want to configure. • Display As: • Action Area: • Action Style Class: • Always Display:
Display Type	The value of this field is set to *Attribute Item*. This indicates that you are configuring the actions available on the Attribute page.
Action	The name of the action available on the Attribute page.
Actions Label Name	The label of the corresponding **Action** action that is displayed on the Attribute page
Display As	Select one of the following values: • *Action*: Select this to set the action as an action button • *Sidebar Action*: Select this to set the action as an action button on the sidebar of the Configuration page. You can define a maximum of 3 actions in the sidebar. Only the first 3 sidebar actions in the list are valid, CPQ does not display the remaining sidebar actions on the Configuration page at all. • *Task*: Select this to set the action as Task menu item in the Status bar.
Action Area	Select one of the following values to define the alignment of the buttons on the Attribute page: • *Center* - For center alignment • *Left* - For left alignment • *Right* - For right alignment • *More* - To add as an item under the More action button

Setting	Description
Action Style Class	Enter one of the following values: • *apt-selected-btn*: 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*: To display an arrow pointing left on the button. • *apt-right-btn*: To display an arrow pointing right on the button.
Is Enabled	Select this to enable the button.
Always Display	Select this to display the button on the Attribute page. The button is displayed even if you disabled the buttong using **Is Enabled** checkbox.

Configuring Installed Products Settings

1. On **App Menu**, click **Conga CPQ Admin** and then click the **CPQ Admin** tab. The new admin console is launched.
2. Click the more icon (⋮) on the top-right corner and select **Application Management**.
3. Click **Installed Products**.
4. Configure the following details.

Setting	Description
NAVIGATION AND ACTIONS	
Flow	Select a flow and enter the following details. • Display As: • Action Area: • Action Style Class: • Always Display:
COLUMNS	
Flow	Select a flow and enter the following details. • Field Name • Is Sortable • Style • Style Class • Header Style

Setting	Description
GENERAL SETTINGS	
Direct Configure Asset Actions	This setting is no longer supported. Instead, you can use the **Allowed Actions** setting to configure asset actions at each Asset Line Item. Also, you can use the **Hide Asset Actions** setting to hide the action buttons from users on the Installed Products page.
Submenu Actions	Enter comma-separated submenu actions that you want to display when the user clicks an action button on the Installed Products page. You can configure submenu actions only on the **Change** and **Relate** actions. The supported values are: • **Change:Configuration** displays the Configuration submenu under the Change action. This submenu enables users to configure an asset line item. • **Change:Quantity** displays the Quantity submenu under the Change action. This submenu enables users to initiate an action to add more licenses for an existing asset stream. • **Change:Split** displays the Split submenu under the Change action. This submenu enables users to split an asset into multiple split lines. • **Change:Merge** displays the Merge submenu under the Change action. This submenu enables users to merge multiple asset lines into a single asset. • **Relate:Component** displays the Component submenu as a drop-down under the Relate action. This submenu enables users to relate an asset component with a service.
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. Commonly used fields are **Start Date**, **End Date**, **Lead Time Expiration**, and **Asset Status**. Usually, all fields under Asset Line Item object are supported but you must enter *ExpirationLeadTimeInDays* value in order to display Lead Time Expiration field in the Search pane.

Setting	Description
Hide Co-Term	Hides the Coterminate with Increment panel on the Change Quantity (Increment Asset) page. The Coterminate with Increment panel is displayed by default.
Show Assets	Select the account hierarchy that the user must see on the Installed products page. Select from the following settings: **Parent and Children**: The entire account hierarchy including the parent and children are displayed. **Parent**: Only parent accounts are displayed. **Children**: Only children accounts are displayed.
Hide Asset Actions	Type comma-separated values of Increment, Amend, Renew, and Cancel actions. Listed actions will be hidden from the users. You can configure 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 20.
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.
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.
GroupBy Fields	Specify the API names of the asset line item fields that must be displayed under the **Group By** drop-down on the ASSET VIEWS tab of the Installed Products settings from the CPQ Admin UI. The same GroupBy API field values will reflect on the **GroupByFields** custom setting at **Setup > App Setup > Develop > Custom Settings > Installed Product Settings**. For more information, see Installed Products Settings.

Setting	Description
RENEWAL	
Renewal Lead Time	Indicates when the renewal quote will be created after an order is activated. By default, the value is 0, which indicates that the renewal quote is created immediately after the order is activated, with the same number of Line Items in the order. If you specify 30, the renewal quote 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.
Renewal Default Price Book	Enter a Renewal Price Book name which you want to associate with renewals.
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. If the implementation teams want to group the renewal opportunities by other parameters on the Asset Line Item (for example, end date, product family, or any custom fields), 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.

Setting	Description

Renewal Execution Mode	Indicates if the Renewal of Asset Line Items must happen automatically or based on 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*.

Renewal Business Object Type	Enter the Business object for which this renewal is taking place. Currently, *Proposal* and *Agreement* are supported.

Setting	Description
Cotermination Preference During Renewal	Select one or more of the following options to allow the user to determine renewal end dates. • UseProposalOrAgreementEndDate: This option enables users to renew an asset using the proposal or agreement end date. CPQ displays the **Use Proposal End Date** option if the user renews assets using the quoting flow and the Renewal Business Object Type is Proposal. CPQ displays **Use Agreement End Date** if the user renews assets using the agreement flow and the Renewal Business Object Type is Agreement. ⓘ If the user performs asset renewal through the Asset Manager flow or CSR flow, the **Use the Proposal End Date** or **Use Agreement End Date** option is not displayed on the Confirm Renewal intermediate page. • RetainCurrentAssetEndDate: This option enables users to retain the current asset end date while renewing an asset. • UseFarthestAssetEndDate: This option enables users to use the farthest asset end date while renewing assets. • UseRenewalDate: This option enables users to renew assets with a custom renewal date. You must select at least one value in this setting. If you have more than one value, you must also select a default value in the **Default Renewal Cotermination Option** setting. If you do not select any default value, the first of the selected values is considered as the default value. Depending on the selected setting, users can now bypass the Confirm Renewal intermediate page altogether. • If there is only selection for co-termination (apart from *UseRenewalDate*), the Confirm Renewal intermediate page is completely bypassed. On clicking Renew, CPQ creates the renewal line and refreshes the Installed Products page to show the detail in the Mini Cart. • If there is more than one selection for co-termination, CPQ displays the Confirm Renewal intermediate page to allow the user to make a selection. • If the selected option is *UseRenewalDate*, CPQ displays the Confirm Renewal pop-up to collect the user input.

Setting	Description
Default Renewal Cotermination Option	Select the default end date preference for renewal cotermination. If there is only one value in the **Cotermination Preferences During Renewal** setting, that option is selected by default. If there are more than one options, you can select a value as a default renewal cotermination option. Based on the default option, CPQ loads the selected renewal transactions on the Installed Products page, Confirm Renewal popup, or Confirm Renewal intermediate page.
MISCELLANEOUS	
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 are 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.
	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.
Allow Mass Change	You can enable or disable mass changes for *must configure* assets with the help of this setting.
Create Renewal Opportunity	Indicates if you want to create renewal opportunities along with renewal quotes. By default, this check box is not selected. This is to prevent the system from running into locking issues when an account has a large number of assets and the batch process creates the renewal opportunities for all these assets.
	When set to true, the Create Renewal Opportunity scheduled job runs and creates the renewal opportunities along with renewal quotes.
	When set to false, the Create Renewal Opportunity scheduled job runs and does not create the renewal opportunities and creates only the renewal quote.

Setting	Description
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.
Account Hierarchy Batch Size	Indicates the number of Account Hierarchy records processed for each batch execution. Recommended value is 500. Valid values range from 1 to 2000.
COTERMINATION PREFERENCES	
End Date preferences for Cotermination	Enables you to decide what End Date preferences the user should see while coterminating incremental licenses using the **Change:Quantity** action. For information on coterminating assets, see Managing Asset Increment with Coterminate Lines. The supported values are: • RetainCurrentAssetEndDate: This option enables the user to retain the current asset end date for cotermination. • UseCustomDate: This option enables the user to enter a custom date for cotermination. • UseProposalEndDate: This option enables the user to use the proposal end date for cotermination. > ⓘ If the user performs asset increment through the Asset Manager flow, the **Use the Proposal End Date** option is not displayed on the Change Quantity intermediate page.
Default Cotermination Option	From the **Default Cotermination Option** drop-down list, select the cotermination option that you want to set as default.

Setting	Description
RELATE CRITERIA	
Relate Action Criteria Fields	Select the asset line item fields to enable relate asset action. This setting shows all fields of Asset Line Item and Price Dimensions defined on asset line item. When the user selects two or more assets and clicks **Relate**, CPQ validates the selected asset line items against the Relate Action Criteria Fields. In case of any discrepancies, CPQ displays a validation error message to the user.
ASSET SOURCE	
Asset Source	Allows you to filter the assets to be displayed on the Installed Products page based on the **Ship** To or **Sold To** or **Bill To** fields of an account. By default, all the assets filtered with the **Sold To** field are displayed. Specify the API name of the Account field that you want to use as a source for displaying assets. For example, to filter the assets based on **Ship To** field, enter Apttus_Config2__ShipToAccountId__c. You can specify all the three account fields and the system considers an OR condition between these fields while filtering. This is useful when an account has different locations for shipping and billing and you want to display only those assets that have been shipped to a certain location.
PURCHASE ID CRITERIA	

Setting	Description
Purchase Id Criteria	Allows user to define a criteria to get coterminate related asset lines in the Coternimate with Increment section on the Change Quantity page during asset increment. The default purchase identification criteria are: Product, Charge Type, Option, and Account. From the **Available Purchase Identification Criteria** list, add the required criteria to the **Selected Purchase Identification Criteria** list. CPQ currently supports only Purchase Identifier and alphanumeric custom fields. For example, if you select Purchase Identifier, during increment of a selected asset, all other assets with the same Purchase Identifier are displayed on the Coternimate with Increment section. ⚠ If you add *Quantity*, *Selling Term*, or *End Date* to the **Selected Purchase Identification Criteria**, the user will receive an error on order creation or activation. Based on the selected purchase identification criteria, the total quantity of an asset spans multiple streams during the increment scenarios, on the Installed Products page (of a renewal quote).
SPLIT ASSET ACTIONS	
Split Asset Actions	Enables the user to perform split asset actions on the Define Split <asset name> page. Add the required split asset action from the **Available Split Asset Actions** to **Selected Split Asset Actions** list. Supported options are: • **Renew**: Enables users to perform the Split and Renew asset action. • **Swap**: Enables users to perform the Split and Swap asset action.
ASSET VIEWS	
You can create a view to be displayed as a view on the Assets grid. Users can only select the view that you create from the Admin UI, but they cannot edit or delete it. Only you can edit or delete the view as an administrator.	
Flow	Select a flow for the view.

Setting	Description
View	Select **New** to create a view.
Name	Enter a name for the view.
Group By	Select a value from the drop-down. The default field is *Product*. ⓘ The values on the **Group By** drop-down depend on the fields you have added in the **GroupByFields** custom setting in the Installed Products settings. For more information, see Installed Products Settings.
Default View	Turn on this button to make this view as the default view.
Make Public	Turn on this button to make this view public.
User Types	If you do not make the view public, you can share it with selected users. Select one of the following user types: • User Profiles • User Roles • Users After selecting the user type, move the respective users from the **Available** list to the **Selected** list.
Field Name	Add the fields to be displayed as columns in the view for users on the Assets grid.
ASSET TERMINATION FIELDS	
Asset Termination Fields	🛑 Do not use this setting. Instead use the *Asset Termination* option from the **Display Type** picklist. For more information, see Configuring Display Column Settings.
MERGE ACTION CRITERIA	
Merge Action Criteria Fields	Select the fields that must be common among the asset line items to be merged. The default fields are product, charge type, selling term, and price type.

Setting	Description
EDITABLE FIELDS FOR CANCELLED LINES	
Editable Fields for Cancelled Lines	Select the fields that you want to make editable for cancelled lines on the cart (during asset termination).
	You can select a custom currency field that is not based on a formula. All out-of-the-box currency fields such as base price and base extended price are not visible in the list of editable fields.
	Sensitive managed package fields that the user cannot edit on a cancelled line item are not visible in the multi-select picklist to the user.
	You can allow Sales Representatives to edit custom asset line item field values on the Confirm Termination page instead of waiting to go to the Cart page to edit these values. For example, they may want to capture the reason for terminating assets. CPQ carries these edited field values from the Confirm Termination page to the Cart page where Sales Representatives can make further changes to those field values, if required.
	ⓘ You must add the same fields to the *Asset Termination* **Display Type** of a specific flow in **Display Colum Settings**. For more information, see Configuring Display Column Settings.

Configuring Collaboration Settings

1. On **App Menu**, click **Conga CPQ Admin** and then click the **CPQ Admin** tab. The new admin console is launched.
2. Click the more icon (⋮) on the top-right corner and select **Application Management**.
3. Click **Collaboration**.
4. Configure the following details.

Setting	Description
PARENT LINE ITEM COLUMNS	

Setting	Description
Flow	Select a flow and enter the following details. • Field Name • Is Sortable • Style • Style Class • Header Style
CHILD LINE ITEM COLUMNS	
Flow	Select a flow and enter the following details. • Field Name • Is Sortable • Style • Style Class • Header Style

Configuring Pricing Engine Settings

1. On **App Menu**, click **Conga CPQ Admin** and then click the **CPQ Admin** tab. The new admin console is launched.
2. Click the more icon (⋮) on the top-right corner and select **Application Management**.
3. Click **Pricing Engine**.
4. Configure the following details.

Setting	Description
GLOBAL PRICING	
Enable External Pricing	Indicates whether pricing will be done externally. The default value is false (disabled).

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.
	The Sales rep can use the **Update Price** button on the Configuration page to calculate pricing after updating fields.
	CPQ allows you to selectively turn on or off Defer Pricing at a flow level. You must set the **deferPricingUntilCart** parameter to True in the URL that you have configured on the **Configure Products** field of the respective quote. For more information, see Creating Custom Buttons for Different Flows.
Pricing Profile	Type Basic, Advanced, or External.
	A pricing profile is Basic, when there are none of the following used:
	- Pricing Rules - Price Matrices - Related Pricing Setup - Bundles Note: If the Pricing Profile field is left blank, the default value is Advanced.
	ⓘ You can configure this setting based on a flow. In such case, the flow-based setting overrides the generic setting.

TOTALING AND SUBTOTALING

ⓘ CPQ allows you to selectively skip totaling at a flow level. You must set the **isCartTotalingDisabled** parameter to True in the URL that you have configured on the **Configure Products** field of the respective quote. For more information, see Creating Custom Buttons for Different Flows.

Setting	Description
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.
Adhoc/Product Totaling Hierarchy	Enter the list of product fields that represent the product totaling hierarchy. Each product field name must be separated by a new line. For Ad-hoc totaling, enter the line item fields that represent a totaling group. If empty, then CPQ uses Bundle Family by default.
Disable Charge Type Totaling	Disables the totaling of charge types.
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.
ADVANCED PRICING	
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 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).

Setting	Description
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.
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.
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.

Configuring Proposal Settings

1. On **App Menu**, click **Conga CPQ Admin** and then click the **CPQ Admin** tab. The new admin console is launched.
2. Click the more icon (⠿) on the top-right corner and select **Application Management**.
3. Click **Proposal Settings**.
4. Configure the following details.

Setting	Description
DEFAULT	
Default Opportunity Quote Owner	Type the owner name to set the default owner for the quote/proposal created from an opportunity. Valid values are: • Opportunity Owner • Current User If not set, Opportunity Owner becomes the owner of the new quote/proposal.
Disable Opportunity Products Copy	Disables copying of opportunity products to the quote/proposal.

Setting	Description
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.
SYNCING	
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 Conga Quote Configuration Integration package installed.
Sync Option Products	Synchronizes options along with bundles when you click Sync with Opportunity for your proposal. The default is to synchronize bundle only. This setting appears only if you have Conga Quote Configuration Integration package installed.
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 Conga Quote Configuration Integration package installed.
MERGE SERVICE	
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 Webservice Endpoint	Type the Conga merge webservice endpoint.
Merge Call Timeout Millis	Type a number to indicate timeout in milliseconds for the merge request. For example: 60,000

Setting	Description
Max Child Level	Type the maximum level to generate the merge data for the proposal.
ESIGN	
Auto Select Attachment?	Select this setting to enable CPQ to select the most recent attachment automatically.
Auto-Select multiple recipients?	Select this setting to enable projects to specify a list of recipients that will be defaulted for eSignature. To use this option, projects will add the **DocuSign Default Recipients** related list on proposals and user will select the list for each proposal.
Auto Select Recipient Field Name	Enter the API name of the field which holds the value for the recipient to select automatically.
SECURITY	
PDF Owner Password	Type the password required to change permissions of the PDF document like printing or editing.
Enable PDF Security	Enabling PDF security lets users apply security settings to PDF documents and protect them with a password.

Configuring Custom Settings

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 > App Setup > Develop > Custom Settings.** Click **Manage** next to the setting you want to configure. In the case of Config System Properties and Installed Product Settings, you can configure certain fields differently for different flows, refer to Configuring Custom Settings for Different Flows.

Custom settings are executed in an order of precedence. For more information, refer to About Custom Setting Order of Precedence.

After you define or modify any custom setting, execute Custom Settings Maintenance Job.

> ⬤ You must execute Custom Settings Maintenance Job after you update any Custom Settings. The changes you make are only effective if you execute the maintenance job. Otherwise, all the computations are processed with stale values. Refer to the Running Maintenance Jobs topic.

Some of the custom settings listed in this section may require you to contact Conga Support or 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).

- Base Private Properties
- Comply System Properties
- Config Asset Pricing Criteria Fields
- Config Asset Pricing Default
- Config Custom Classes
- Config Custom Display Actions
- Config Custom Display Columns
- Config Data Cache
- Config Field Set
- Config Flow
- Config Line Item Custom Fields
- Config Select Options Settings
- Config Select Products Settings
- Config System Properties
- Config User Preferences
- Installed Products Settings
- Lookup Field Settings
- Proposal System Properties

Base Private Properties

This custom setting holds the list of all the base library properties.

Setting	Description
Enable Usage Data	Select this to enable data usage.
Usage Data Endpoint	Enter the usage data endpoint.

Setting	Description
Usage Data Key	Enter the usage data key
Usage Data Provider Class	Enter a list of comma-separated class names of Usage Data Providers specifically the class that implements Apttus_Base2.IUsageData interface.
Usage Event Key	Enter the usage event key.

Comply System Properties

This custom setting holds the 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.
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, use Apttus__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, see **Agreement Number/Header Configuration** section in Configuring Offline Agreement Window.

Setting	Description
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, see **Agreement Number/Header Configuration** section in Configuring Offline Agreement Window.
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 Conga 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.

Setting	Description
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, see Configuring Document Finder.
Document Naming Convention	Specify a value to apply a custom naming convention for all proposal 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. The following attributes permitted when formulating a document naming convention are: • %checkintype% • %action% • %templatename% • %user% • %timestamp% • Proposal attributes such as %:Name%. Note: any variable prefixed by ':' represents a field on the Proposal object. Example proposal document name using the default naming convention: *SOW_Regenerated_SOW ABC_2015-08-07*

Setting	Description
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.
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 Conga 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, see Enabling Contract Document Versioning.
Enable Merge Call Debug	Enable Merge Service debugging.

Setting	Description
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. The **Submit** and **Submit for Generate (Async)** 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, see Configuring Template Versioning.
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. For more information, see **Setting the Date and Time Format for the Footer** section in Configuring Offline Agreement Window.
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.

Setting	Description
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 Conga merge webservice endpoint. For example: https://mergews.apttus.net/cgi-bin/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.
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 Conga Contract-Configuration Integration package installed.
Sync Option Products	Synchronizes options along with bundles. The default is to synchronize bundle only. This setting appears only if you have the Conga Contract-Configuration Integration package installed.

Setting	Description
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, see CLM for Administrators.
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.

Config Asset Pricing Criteria Fields

This custom setting holds the list of API names of fields from Asset Line Item object used in setting Asset Pricing Indicator. Criteria with a specific Line Status can also be defined. You can create multiple data sets as well.

Setting	Description
Criteria Field Names	Enter the list of API names of fields from the line item object used in setting the asset pricing indicator. Each field must be separated by a comma or a new line.
Line Status	Enter the list of line statuses to make the criteria valid only for the line item with line status mentioned in this field. Examples of valid values are *Renewed*, *Amended*. ⓘ If you want to define a Line Status specific criteria, you must define two criteria: one with Line Status and another with blank Line Status for the same criteria fields.

Config Asset Pricing Default

This custom setting holds the default value for the asset pricing indicator associated with a line status. The value defined in the setting is only applicable on the line status defined. Create different data sets for different line statuses. Click **Manage** then click **New** to create a new record. Specify the exact value of line status in **Name**. Examples of valid values are *Renewed, Amended.* The list of valid line status values is defined in the Line Item object. Define the field mentioned in the table as desired and click **Save**.

Setting	Description
Default Asset Pricing Indicator?	Select this checkbox to enable asset pricing by default for Asset Related Line items associated with the given line status.

ⓘ CPQ applies Price Escalator on new ramp lines created during ABO operations regardless of the **Default Asset Pricing Indicator?** setting (true or false). A new ramp line always gets its value from the price list item (PLI) pricing. For more information, see Use Case: Applying Price Escalator on New Ramp Lines During ABO Operations.

Config Custom Classes

This custom setting registers all custom classes that implement Callback classes for customization. Click **Manage** and click **Edit** next to *Config Custom Classes*. Enter API names of the custom classes that you created next to the Callback class you want to implement. The following table lists the Callback classes and the global interfaces the Callback class implements.

Callback Class	Interface
Action Callback Class	CustomClass.IActionCallback3
Action Invoker Callback Class	CustomClass.IActionInvokerCallback
Action Params Callback Class	CustomClass.IActionParamsCallback
Adjustment Line Item Callback Class	CustomClass.IAdjustmentLineItemCallback
Adjustment Spread Callback	CustomClass.IAdjustmentSpreadCallback

Callback Class	Interface
Advanced Approval Callback Class	CustomClass.IAdvancedApprovalCallback
Asset Line Item Callback Class	Class CustomClass.IAssetLineItemCallback3
Asset Renewal Custom Callback Class	CustomClass.IAssetRenewalCustomCallback
Bulk Loyalty Point Callback Class	CustomClass.IBulkLoyaltyPointCallback
Cart Approval Callback Class	CustomClass.ICartApprovalCallback2
Deal Optimizer Callback Class	CustomClass.IDealOptimizerCallback2
Display Action Callback Class	CustomClass.IDisplayActionCallback
Formula Callback Class	CustomClass.IFormulaCallback ⓘ This callback has been deprecated.
Loyalty Cycle Callback Class	CustomClass.ILoyaltyCycleCallback
Loyalty Point Callback Class	CustomClass.ILoyaltyPointCallback
Metadata Callback Class	CustomClass.IMetadataCallback. ⓘ This callback has been deprecated.
Option Filter Callback Class	CustomClass.IOptionFilterCallback
Pricing Callback Class	CustomClass.IPricingCallback2
Pricing Extension Callback Class	CustomClass.IPricingExtensionCallback
Product Attribute Callback Class	CustomClass.IProductAttributeCallback
Product Filter Callback Class	CustomClass.IProductFilterCallback
Recommendation Callback Class	CustomClass.IRecommendationCallback

Callback Class	Interface
Related Pricing Callback Class	CustomClass.IRelatedPricingCallback
Revalidation Callback Class	CustomClass.IRevalidationCallback
Shipping Callback Class	CustomClass.IShippingCallback
Tax Callback Class	CustomClass.ITaxCallback
Validation Callback Class	CustomClass.IValidationCallback

Config Custom Display Actions

This custom setting holds information about custom actions displayed in, for example, Installed Products view, Cart view.

Setting	Description
Action Area	Enter a value to select the action display area. Valid values are Left, Right, Center, and More. The value *More* indicates that the action is displayed under More button.
Action Label Name	Enter the name of the custom label for the action.
Action Name	Enter the API name for the action.
Action Page	Enter the name of the custom action page. You can implement a custom action either as a Visualforce page or as a custom class.
Action Params	Enter the URL parameters for the action. You can pass multiple parameters by using the ampersand character to separate each parameter. Each parameter is expressed in the form param-name=param-value. For example, templateName=XYZ&draft=true
Action Style Class	Enter the style class name for the action.
Action Type	Enter the action type.

Setting	Description
Always Display	Select this setting to display the action (grayed out) even when it is not enabled.
Behavior	Enter the behavior that the open page action takes.
Display As	Enter the action, task, or both to display the action on a page as action button or task.
Display Type	Enter the action display type. Valid values are: • Cart Line Item • Asset Line Item
Flow	Enter the flow associated with the action.
Is Enabled	Select this setting to enable the action for the display.
Sequence	Enter a number for the action display sequence.

Config Custom Display Columns

This custom setting holds information about fields displayed in, for example, Installed Products view, Cart view.

Setting	Description
Display Type	Enter the column display type. Valid values are: • Cart LineItem • Installed Product
Field Name	Enter the field API name for the column.
Flow	Enter the flow associated with the column.
Header Style	Enter the style used to display the header of a column, which is used primarily for adding inline CSS styles.
Is Editable	Select this setting to enable column value to be edited.

Setting	Description
Is Price Point	Select this setting to determine type of a line item for use in Cost & Profitability Price Waterfall. If you select the setting, item is a Price Point. If you deselect the setting, item is an Adjustment.
Is Sortable	Select this setting to enable column sorting.
Sequence	Enter a number for the column display sequence.
Style	Enter the style used to display the column, which is used primarily for adding inline CSS styles.
Style Class	Enter the style classes that must be applied to dynamic columns.

Config Data Cache

Config Data Cache is used to store the list of line item fields that are used by various custom settings, like **Config Display Columns**, **View Cart Custom Fields**, and **Constraint Rule Criteria**. These fields are used in line item queries constructed for CPQ's internal use. This optimizes the queries to fetch specific fields instead of fetching all the fields. The administrator must not update this custom setting manually.

Setting	Description
Data	The delimited data.
Is Last	The last value indicator

Config Field Set

Config Field Set is used to store list of fields for special purpose.

Setting	Description
Data	Enter delimited data.

Setting	Description
Is Last	Enter the last value indicator

Config Flow

This custom setting allows you to define a user interface flow for CPQ by using the default pages available. Click **Manage** then click **New** to create a new flow. Define the field mentioned in the table as desired and click **Save**.

Setting	Description
Assets Page	Enter the name of the page you want to display as Asset page. Supported value is: • Apttus_Config2__Cart#assetsgrid
Cart Page	Enter the name of the page you want to display as Cart page. Supported value is: • Apttus_Config2__Cart#cartgrid
Catalog Page	Enter the name of the page you want to display as Catalog page. Supported value is: • Apttus_Config2__Cart
Option Page	Enter the name of the page you want to display as Option page. Supported value is: • Apttus_Config2__Cart
Product Attribute Detail Page	Enter the name of the page you want to display as Product Attribute Detail page. Supported value is: • Apttus_Config2__Cart

Config Line Item Custom Fields

The Config Line Item Custom Fields requires you to add these fields.

• Apttus_Config2__Comments__c
• Apttus_Config2__OrderLineItemId__c

- Apttus_Config2__OrderLineStatus__c
- Apttus_Config2__AutoRenew__c
- Apttus_Config2__AutoRenewalType__c
- Apttus_Config2__AutoRenewalTerm__c
- apttus_config2__paymenttermid__c
- apttus_config2__netunitprice__c

Config Select Options Settings

Configuration settings for option selection pages. The record should be named "Primary Settings".

Setting	Description
Bundle Description Field	Enter the field names from the product object to be displayed as bundle description.
Enable Option Page Search	Select this setting to allow option search on the bundle configuration page.
	This setting enables a search bar on the Bundle Configuration page. The setting allows the user to search for options, attributes, and sub-bundles on the Configuration page.
Hide Disabled Options	Select this setting to hide the options that are 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.
Hide Price Column	Select this setting to hide the price information column from the options page.
	Be careful in hiding this critical information.
Listed Options Column 2	Enter the API name of the product field to be displayed in the listed options section. A valid value could be like ProductCode.
Listed Options Column 3	Enter the API name of the product field to be displayed in the listed options section. A valid value could be like ProductCode.

Setting	Description
Main Section Ratio	Enter the ratio to divide the options page. The ratio of the main sections in the page must be separated by a colon. The default value is 0:70:30. This is applicable to the SelectConfigOptionsDetail and ConfigureBundle pages.
Max Inline Option Attributes	Specify the maximum number of inline option attributes to be displayed in the attribute group. The default value is 3.
Option Field In Summary	Enter the Option field to be displayed on right hand side of configuration page.
Show Bundle Detail	Select this setting to show bundle price and quantity fields on options pages.
Show Option Quantity	Select this setting to display the Quantity in the body of the options pages. This setting controls the SelectConfigOptionsDetailView and ConfigureBundle page.

Config Select Products Settings

Configuration settings for product selection pages. The record should be named as "Primary Settings".

Setting	Description
Always Display Leaf Products	Select this setting to load and display all leaf nodes for a specific category. These leaf nodes have products associated with them. With this the setting, the user needs to just click the main category or any subcategory to load all sub-categories (and products) at one click.
Cache All Products	Select this setting to cache all products to avoid SOQL queries. This setting must be enabled when the number of product records is small. This is a Beta feature.

Setting	Description
Carousel Category Level	Enter a number for the maximum depth allowed to be browsed in the Product Carousel. No value in this field allows infinite browse depth. A value of zero will restrict you to the root level.
Carousel Default Icon	Enter the Default Icon to be used for the categories in the Carousel.
Cart List Item Description Field	Enter the API field name of the **Long Description** field to be used in the Cart Detail View page.
Collapse All Leaf Option Groups	Select this setting to collapse all leaf option groups on the Config Options Detail View page.
Custom Action Label Name	Enter the name of the custom label for the action.
Custom Action Page	Enter the name of the custom action page.
Disable Add Another	Select this setting to disable the user from adding multiple instances of the same product.
Disable Add To Cart for Option Products	Select this setting to hide **Add to Cart** and **Configure** buttons for the option products on catalog pages. This is a global setting to hide these buttons.
Disable Add To Cart On Click	Select this setting to disable the **Add To Cart** button on click until the server confirms the addition of the product.
Disable Favorites	Select this setting to disable saving the favorite, favorites category, and other favorites-related UI items.

Setting	Description
Enable Category Filter	Filters the categories based on search results and displays them 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.
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. ⓘ **Deprecated setting** Favorites Display Label Settings have been deprecated. Now you need to rely on overriding custom labels for renaming 'Favorites' with your choices such as Quick Buys, Most Used, and so on etc. For example, if you want to use the term "Quick Buy" for favorites, CPQ contains two custom labels whose values you must override. Go to Setup > Custom Labels. • Favorites > Click "New Local Translations/ Overrides" > Choose Language as "English" > Write "Quick Buys" in translated value area. • SaveAsFavorite > Click "New Local Translations/Overrides" > Choose Language as "English" > Write "Save as Quick Buy" in translated value area.
Favorites Icon Attachment Id	Enter the ID to the attachment of the Favorites icon.

Setting	Description
Favorites Upload Image	You can choose an image to upload for your group of favorite configuration categories. This image is displayed under your favorites (as entered in the **Favorites Display Label**) category on the catalog page. By default, no image is set
Hide Breadcrumb	Hides the breadcrumb navigation element.
Hide Cart Header	Select this to hide the Cart Header for selected pages.
Hide Compare Products	Disables the Compare Products feature in the catalog page.
Hide Configure Action	Hides the Configure action for a Product on the catalog page.
Hide Default Options In Description	Hides the default options listed below the product description.
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.
Hide Line Item Attribute Details	Hides the Line Item Details section on the Product Attribute Detail page.
Hide Listed Products Price Column	Hides the price information column in the listed product section. By default, this is not selected.
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.
Hide Options in Quick Add Mode	Select this to hide option products in Quick Add mode
Hide Product Image	Hides the product image from the catalog page.
Hide Selected Product Charge Types	Select this to hide charge types for selected products from the tree view options page.

Setting	Description
Hide Selected Products Column1	Hides the column that displays the product name.
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.
Listed Favorite Configurations Column1	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. If you have a requirement of displaying one more field, use **Listed Favorite Configurations Column 2** custom setting.
Listed Favorite Configurations Column2	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.
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
Listed Products Default Icon	Enter the ID to the attachment of the default Product Icon used for the Listed Products.
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.

Setting	Description
Minimize Cart	Displays the cart icon and hides the cart sidebar on the right-hand side of the catalog page.
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.
Product Cache Fields1	Enter the first list of custom fields required for the product cache.
Product Cache Fields2	Enter the second list of custom fields required for the product cache.
Product Cache Fields3	Enter the third list of custom fields required for product cache.
Product Cache Fields4	Enter the fourth list of custom fields required for product cache.
Read Only Location	Makes the Location field read-only.
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.
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.
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
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*.
Show Favorite as First Category	Select this setting to display the favorites category on top in the list of product categories.
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 Quantity Input	Displays the Quantity field on the catalog page.
Show Recommended Products Cart View	Displays the recommended products component in the Cart Detail View page.
Show Selected Product All Charges	Displays all the charge types for selected products in the tree view options page.
Show Selected Products in Config Options	Displays the Selected Products widget in the Config Options Detail View page.

Config System Properties

This custom setting holds the Configuration & Pricing System properties. The record *System Properties* should already exist. If not, create a record named *System Properties.* Click **Manage** then click **New** to create a new record. Define the fields mentioned in the table as desired and click **Save**.

Setting	Description
Actions Column Position	Enter the position of the actions column in the cart page. Valid values are: • Left • Right
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. Select the custom setting **Totaling Group Type**.
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>.
Auto Execute Post-Pricing Step	Select this setting to enable CPQ to execute the post-pricing logic automatically after pricing. The setting is applicable only after the pricing logic is defined in the pricing extension callback.
Auto Execute Pre-Pricing Step	Select this setting to enable CPQ to execute the pre-pricing logic automatically before pricing. The setting is applicable only before the pricing logic is defined in the pricing extension callback.
Auto Finalize On Cart Approval	Select this setting to automatically finalize the cart when it is approved and ready for finalization.
Auto Refresh Usage Tier	Select this setting to enable modifiable usage tiers to automatically re-calculate tier price upon PLI change.
Auto Sync On Cart Approval	Select this setting to automatically synchronize the cart when it is approved and ready for finalization.
Auto Update Category View	Select this setting to enable auto incremental updates to category view.

Setting	Description
Base Product Relation Field	Enter the API name of the product field that associates a child product with the base product.
Bypass Sharing	Select this setting to enable apex code to bypass record sharing.
Bypass Shopping Cart	Select this setting to display the **Update Price** and **Finalize** buttons and hide 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.
Cart Edit Access Idle Timeout in Minutes	Enter a number to specify the number of minutes of idle time a configuration can be kept in edit mode.
Cart Theme	Cart Themes are deprecated. No changes are executed even if the values are defined in the setting.
Cascade Shared Attribute Updates	Select this setting to cascade the shared bundle attribute updates to options in the bundle. If this setting is not selected, 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.
Check Many Options	Select this to enable mass selection of options on the Configuration and Service Bundle page. This setting allows the Sales rep to select, clear the selection, and update options, and attributes before processing pricing and expression, and invoking rules.
Cleanup Invalid Rule Prompt	Select this setting to enable removal of line items prompted by currently invalid rules.
Compute Totals In Separate Step	Select this setting to enable CPQ to perform the totaling in a separate step or combine it 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.

Setting	Description
Constraint Rule Execution Mode	Enter the mode to perform the constraint rule execution on the client-side or the server-side. By default, no value is specified and the system assumes the value as *Server*. Valid values are: • *Client* • *Server* • *CMS:* For TurboConfig flow. For more information, see Configuring TurboConfig Settings. • *Turbo:* For a Unified Cart Enabled org. For more information, see Enabling TurboEngines in an Org. This custom setting supports inclusion, exclusion, and validation rules on the client-side as well as the server-side. ⚠ The Salesforce Validation Rules are not supported with Conga TurboEngines. ⓘ In a Service CPQ flow, always execute constraint rules in Client mode.
Column Render Threshold	Enter the number of columns to load when Cart Grid renders. Use this setting when you have defined a style for the columns in the Display Column settings. This improves scrolling performance when you increase the column width. The default value is 10. Enter a value based on the number of columns in the cart.
CR Maintenance Governor Limits Threshold	Enter the percentage of governor limit to be used as a threshold for constraint rule maintenance job. When the maintenance job fulfils the threshold, CPQ divides the maintenance job in multiple batches and then continue to execute the job. The value must be in range of 10 to 100. By default, the threshold will be 100% of the governor limit.

Setting	Description
CSS Override	Enter the name of the static resource to override CSS in the catalog, attributes, options, and cart pages. For more information, see Configuring CSS Override. ⓘ To define CSS for a specific flow, define the field in the dataset of that flow and leave the field blank in **System Properties**. If you define the **CSS Override** in both the datasets, only the value in **System Properties** is considered.
Currency Field Precision	Enter a number to specify decimal places for currency precision. The default currency precision is 2 decimal places. ⓘ Currency Field Precision only applies to **Net Price**, irrespective of the currency. You can create a formula field to define the precision of the values of the amount fields based on different currencies.
Custom Asset Action Label Name	Enter a label name for a custom asset action.
Custom Asset Action Page	Enter the Visualforce name for the custom asset action page.
Custom Deal Guidance Page	Enter the name of the custom Deal Guidance page.
Custom Option Attribute Page	Enter the name of the custom Option Attribute page.
Custom Pre-Pricing Fields	Enter the list of custom fields from the line item object that determine the pre-pricing indicator.
Custom Pricing fields	Enter 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 you makes any changes to these custom fields, the system is set to reprice before clicking **Review and Finalize**.

Setting	Description
Default Asset Pricing Indicator?	Select this setting to make asset pricing as default for asset line items. However, you can override this global setting for certain ABO actions and the corresponding line statuses on the cart. In that case, CPQ applies price from price list items. For more information, see Overriding the Default Asset Pricing for Line Statuses Based on ABO Actions. CPQ applies Price Escalator on new ramp lines created during ABO operations regardless of the **Default Asset Pricing Indicator?** setting (true or false). A new ramp line always gets its value from the price list item (PLI) pricing. For more information, see Use Case: Applying Price Escalator on New Ramp Lines During ABO Operations.
Default Catalog Page	Enter the Visualforce page name for the default catalog page. Valid values are: • SelectConfigProducts • SelectConfigProductsFilterView
Defer Constraint on Option Selection	Select this setting to process the constraint rules on option selection on the Configuration Page. If the selected the constraint rules deferred until the Sales Rep clicks **Go to Pricing** or any other action button available on the Configuration page.
Defer Pricing	Select this setting to price the products before displaying the cart or after displaying the cart. The default is true (defer until cart is displayed). The Sales rep can use the **Update Price** button on the Configuration page to calculate pricing after updating the fields.
Defer Validation Check in Bundles	Select this setting to defer the validation check in the bundle configuration step. This is setting is selected by default.

Setting	Description
Direct Configure Asset Actions	Select this setting to open 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	Select this setting to disable the totaling of charge types.
Disable Constraint Rules	Select this setting to disable constraint rules. This may be used to optimize performance when constraints are not used in the organization. The default value is false (enabled).
Disable Existing Asset Pricing	This setting is deprecated.
Disable Pricing Query Optimization	Select this setting to disable pricing query optimization for the cart. This setting allows the system to load a 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.
Disable SOSL Text Search	Select this setting to disable the SOSL catalog search. When disabled, CPQ uses SOQL catalog search.
Display Cart Actions As Dropdown	Select this setting to display the line item action buttons as a drop-down menu on the cart page.
Enable Adjustment Buckets	Select this setting to enable adjustments applied in the Mass Adjustments dialog box for a line item to be grouped in buckets, to be used as targets for applying further adjustments. This setting works only if you have set a finite value in the **Max Adjustment Lines** custom setting.
Enable Adjustment Spread	Select this setting to enable the adjustment on a bundle to spread to its options.

Setting	Description
Enable Aggregate Pricing	Select this setting to enable aggregate pricing. This may be used to optimize performance when aggregate price rules are not used in the organization. The default value is false (disabled).
Enable Auto Reprice	Select this setting to automatically reprice the cart when a price component is modified, without applying any pricing rules. This setting provides client-side computation facility. • On Cart Grid with custom setting **Enable Auto Reprice** set to ON, when you change any Quantity on cart line item and click **Reprice**, CPQ updates the pricing fields such as Extended Price, Net Price for the specific row. • Calculates totals (by charges and frequency) and grand totals • Calculates subtotals for category As a sales rep you can see estimated pricing without any delay using this setting. Estimated pricing means when quantity is changed and adjustment or discounts are provided, pricing calculation is done at the client-side.
Enable Auto Sequencing For Options	Select this setting to enable auto sequencing for options. If you do not select this option, the Option Item Sequence in the shopping cart and the Line Itemsisbased on the user selection sequence rather than the Bundle Options sequence.
Enable Base Price Adjustment	Select this setting to enable unit level price adjustment for the Base Price of the product in the cart.
Enable Base Price Rounding	Select this setting to enable CPQ to round list price and base price fields in the line item using the currency field precision setting.
Enable CartLocking for Concurrent Access	Select this setting to enable your cart for concurrent access to other sales representatives. For more information, see Enabling Cart Locking for Concurrent Access.

Setting	Description
Enable Contextual Totals	Select this setting to display subtotals and totals section at the bottom of the cart page for Cart Grid UI. By default, this setting is not selected.
Enable Custom Rounding	Select this setting to enable rounding of selling term and price in each price calculation step. • Rounding of Selling Term: This setting considers the precision of selling term defined in the **Term Field Precision** setting and rounds the selling term consistently both on the UI and in backend calculation. • Rounding of Pricing: This setting rounds the price in each price calculation step considering the precision defined for each currency, both on the UI and in backend calculations. The amount in price calculation steps (for list price, base price, adjusted price, and net price only after each step such as price matrix, price ruleset, line level manual adjustment, group adjustment) is calculated considering the precision of the currency. ⓘ Price rounding at backend occurs only for the base price, adjusted price, and net price when this setting is enabled.
Enable Defaulting For Products	Select this setting to enable defaulting 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 Default Quantity For ABO Item	Select this setting to enable the default quantity rules for ABO items. When you enable this setting, CPQ supports only attribute-based default quantity during ABO operations.
Enable External Pricing	Select this setting to enable external pricing. The default value is false (disabled).
Enable Fast Cloning	Select this setting to enable fast cloning.

Setting	Description
Enable Field Expressions	Select this setting to enable users to use Expression Builder for attributes, option groups, and options. This setting is enabled by default.
Enable File	Select this setting to enable use of Files (Salesforce). This setting also enables preview on the Doc Gen page. ⓘ After you select this setting, it is mandatory to run Migration, which moves the documents in Notes & Attachments from CPQ to Salesforce Files.
Enable Keyed Matrix Pricing	Select this if you have option products with a large number of dimensions of type discrete. This also helps in reducing SOQL queries.
Enable Location	Select this setting to enable users to select a location for a cart. The line items will have the current cart location assigned to them.
Enable Matrix Pricing For Options	Select this setting to enable matrix pricing 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 Notification Feed	Select this setting to enable push notifications. The notifications are enabled for the following features: • Quote Collaboration notifications. The notification pop-up appears when any collaboration request is submitted, completed, accepted, and merged in the parent cart. • To enable push notifications for async CPQ operations. For more information, see Configuring Progress Tracker for Async Operation.
Enable Optional Items	Select this setting to enable optional items in the cart.
Enable Paginated Grid	Select this setting to enable pagination for cart grid.

Setting	Description
Enable Price Matrix Audit Trail	Select this setting to enable CPQ to create audit trail entries when adjustments are applied on the Line Item through Price Matrices. The audit trail entries are created on the Adjustment Line Item object. ⓘ This feature does not work if you use adjustment buckets.
Enable Price Rule Audit Trail	Select this setting to enable CPQ to create audit trail entries when promotions are applied on the Line Item through Price Rules. The audit trails entries are created on the Adjustment Line Item object.
Enable Price Rule For ABO Item	Select this setting to indicate whether price rule adjustments should be applied to ABO items.
Enable Total Level Incentive	Select this setting to enable incentives at the total level.
Enhanced Search URL	Select this setting to enhance search URL endpoint.
Exclude Optional Products	Select this to exclude optional products when CPQ evaluated the bundle's structural requirements. When you enable this setting, CPQ excludes the optional products from the option group Min/Max criteria.
Expand Bundles in Cart	Select this setting to expand all attributes and options of a bundle product in the cart. ⓘ Pagination feature is not supported along with Expand Bundles in Cart.

Setting	Description
FavoriteFilters	Enter the API names of fields of type picklist and multi-select picklist from favorite configuration object. These fields define the filters that can be used to categorize favorite configuration on catalog page. If you define default values for the picklists, they are displayed on the catalog page as well. Only the first five fields are displayed on the catalog page. Enter the API Names of the fields separated by a comma or by a new line.
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.
Fixed Button Bar	Select this setting to display a fixed bar for actions button on the catalog, attributes, options, cart, and installed products page.
Generate Relative Url For Sites	Select this setting to enable CPQ to generate relative URL for links for a Communities user. This custom setting is deprecated.
Grid Render Threshold	This setting helps to improve the grid scroll performance, with a default value of 20. You must adjust the value based on the number of lines displayed on the Cart. ⓘ Earlier the scroll performance was managed using **APTS_VirtualizationThreshold**, which is now deprecated. You must use the Grid Render Threshold instead of **APTS_VirtualizationThreshold**.

Setting	Description
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. For more information, see Viewing the Cart in Grid View. ⓘ API names with relationship fields are not supported. For example, *Apttus_Config2_ProductIdr.name* is not supported. However, the regular Line Item Field API Names like *Apttus_Config2ProductId_c* are supported
Guide Page Default	Enter the Visualforce page name for the default guide page for product selection.
Hide Asset Actions	Enter the actions to hide on the Installed Products page. Use a comma to separate values. For example, Increment, Amend, Renew, Cancel. These actions are tied with your Flow settings. For more information, see Defining Allowed Actions for Assets.
Hide Cart Views	Select this setting to hide Cart Views on the cart.
Hide Cart Views Creation	Select this setting to hide the **Create New View** menu option in the Cart View.
Hide Copy Action	Select this setting to hide the copy action in the cart. The default value is false (copy icon is displayed).
Hide Dates For One Time Charge	Select this setting to hide the dates for price list items of type One-Time Charge.
Hide Evergreen End Date	Select this setting to hide the end date for Evergreen Proposals.
Hide Grand Total	Select this setting to hide the grand total line on the cart page.

Setting	Description
Hide Resolve Config Link	Select this setting to hide the link to the resolve configuration error page and instead display the error along with the constraint rule message.
Hide Subtotals In Cart	Select this setting to hide subtotals in the cart view. The default value is false (show subtotals in the cart). When selected, the system hides subtotal lines in the Summary section and still shows Category Total, One Time Total, and Grand Total.
Instance Url	Enter 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	Select this setting to keep abandoned carts. The default value is false (abandoned carts are moved to the recycle bin).
Large Cart Spilt Criteria Fields	This setting is not functional. You must use the Split Cart Criteria Fields setting to define criteria for the Large Cart flow.
Large Cart Threshold	Enter the value that defines whether it is a Large Cart or not and the mode of the cart finalization from synchronous or asynchronous. For example, If you specify the value 499 and your line item in the Cart is greater than or equal to 500, then the quote finalization happens in asynchronous mode or else in the synchronous mode.

Setting	Description
Max Adjustment Lines	Enter the maximum number of adjustment lines allowed for a line item. This setting is applicable only for manual adjustments, not for auto-applied adjustments (such as promotions, price matrices, and price rulesets). For example, if Max Adjustment Lines = 3, the sales representative can apply a maximum of three manual adjustments on the multiple adjustments pop-up. After that, CPQ disables the **+ Add another adjustment** button and displays a message. • If the value of this setting is greater than or equal to 1, CPQ displays auto-applied adjustments (in read-only mode) on the multiple adjustments pop-up. Sales reps can create as many adjustments on the pop-up as the value defined in this setting. After reaching the limit, CPQ disables the **+ Add another adjustment** button and displays a message. • If the value of this setting is 0, CPQ displays only auto-applied adjustments (in read-only mode) on the pop-up. Sales reps cannot add any manual adjustments. CPQ does not display the **+ Add another adjustment** button, but displays a message.
Max Allowed Lines For Mass Actions	Enter the maximum number of line items to be selected for mass actions on the cart. The default value for selecting the line items for mass action is 10. **Max Allowed Lines For Mass Actions** custom setting is overridden when **Perform Mass Actions in Parallel** custom setting is enabled by the user for selecting a large number of line items on the cart for mass action in one go.
Max Constraint Rules Round Trip	Enter 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 picklist values when adding miscellaneous items to the cart. The default values are *Sales Tax* and *Shipping & Handling*.

Setting	Description
Multi Currency Management	Enter a value to define the currency precision in a multi-currency org. Valid values are *Platform* and *None*. • *Apttus:* This is the default value. If you choose this value, the system considers the currency 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 considered. If you do not specify any value in the **Decimal Places**, the system considers the default value as 2.
Option Line Item Columns	Enter the field name from the line item object displayed on the Option Hierarchy page. To add more than one field name separate the names by a new line or a comma.
Option Pricing Chunk Size	Enter a number for the option pricing chunk size for bundles with a large number of options. The default value is 100. The SFDC governor limit affects the chunk size.
Option Product Columns	Enter the field name from the product object displayed on the Option Hierarchy page. To add more than one field name separate the names by a new line or a comma.
Options Page	Enter the Visualforce page name for the default options page. If no value is specified, the options page defaults to the icon view. Valid values are: • SelectConfigOptions • SelectConfigOptionsTabView • SelectConfigOptionsListView
Percentage Field Precision	Enter a number to specify decimal places for percentage precision. The default percentage precision is 2 decimal places.
Perform Mass Actions in Parallel	Select this setting to perform mass actions on the cart. You can copy and remove multiple line items from the cart by mass action in one go.

Setting	Description
Populate Attribute Extensions	Select this setting to indicate that the attribute values are populated in the attribute extension objects.
Pricing Batch Size	Enter 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. ⓘ You can configure this setting based on a flow. In such a case, the flow-based setting overrides the generic setting.
Pricing Profile	Enter *Basic*, *Advanced*, or *External*. A pricing profile is *Basic*, when there are none of the following used: • Pricing Rules • Price Matrices • Related Pricing Setup • Bundles If the Pricing Profile field is left blank, the default value is *Advanced*. ⓘ You can configure this setting based on a flow. In such a case, the flow-based setting overrides the generic setting.
Product Attribute Detail Page	Type the Visualforce page name for default attributes page. This is the custom page to capture product attribute details.
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 the **Product Attribute Value** object and their corresponding 4 lookup relationships, you must specify 4 API names in this field.

Setting	Description
Product Display Max Length	Enter the maximum number of characters to be displayed for the product name on the cart page. If the product name exceeds the maximum length, the ellipsis is shown. The default value is 21.
Product Option Price Order	Enter the order of execution of option pricing (with adjustments) in conjunction with the price rule set and matrices. The valid values are: *First* and *Last.* By default, the value is *Last.* • *First* - The system applies the adjustments before the price rulesets and after price matrices. • *Last* (Default) - The system applies the adjustments after the price rulesets.
Product Sorting Fields	Enter the list of API names of fields from the Product object to be displayed in the sorting drop-down on the Catalog page. The Sales rep can use the **Sort** drop-down to select a field from the list to sort the products on the Catalog page.
Quantity Field Precision	Enter a number to specify decimal places for quantity precision. The default quantity precision is 2 decimal places.
Refine Search Checkbox Display Limit	Enter a number to specify the limit after which the Refine Your Search options are displayed as a picklist. By default, the options are displayed as checkboxes. The minimum and the maximum values are 0 and 20 respectively. Any number less than 0 is considered 0 and any number greater than 20 is considered 20.

Setting	Description
Related Price Scope	Enter a value 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 another Bundle. • If value = *Bundle*, the Option price is based on the related product instances within the same Bundle.
Remove Invalid Rule Products	Select this setting to automatically remove line items auto-included by invalid rules on cart launch. If the setting is disabled, the user is prompted to take appropriate action.
Resolve Configuration Page	Enter the default Resolve Configure page name. On Resolve Configuration Errors/Warnings, this page is used. • *ResolveConfig* • *ResolveConfigProducts*
Revalidation Product Columns (D)	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. ⓘ This setting is deprecated.

Setting	Description
Rounding Mode	Enables currency rounding in pricing calculations. This setting rounds adjustments before calculating the base price. Enter one of the following values: • UP: Rounds the currency to the next number. For example, 21.2 is rounded to 22. • DOWN: Rounds the currency to the previous number. For example, 21.8 is rounded to 21. • HALF_UP: Rounds the currency to the next number if the decimal is equal to or greater than 5. For example, 21.5 to 21.9 is rounded to 22. • HALF_DOWN: Rounds the currency to the previous number if the decimal is equal to or smaller than 5. For example, 21.1 to 21.5 is rounded to 21. • HALF_EVEN: Rounds the currency to the nearest even number. For example, 23.5 is rounded to 24 and 22.5 is rounded to 22.
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. 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.
Run Post Finalize Trigger In Async Mode	Indicates whether the finalization task should be run asynchronously.
Run Validation Callback On Add	Select this to invoke the Validation Callback upon upon clicking **View Cart** on the Mini Cart.

Setting	Description
Same Day Cancellation	Select this option if you want the Asset Cancellation applicable on the same day. For example, while performing Termination on an Asset you enter the Termination Date as 5/11/2016. If you select Same Day Cancellation, the cancellation will be effective from 5/11/2016. Otherwise, the cancellation will be in effect a day later which is 5/12/2016.
	Same Day Cancellation is applicable by default. Clear the checkbox to apply cancellations after a day.
Save On All Actions	Saves the configuration when you click any action on the cart page.
	ⓘ This setting is deprecated.
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.

Setting	Description
Selling Term Calculation Method	Indicates the method to calculate the selling term for line items on the Cart page.
	The valid values are:
	• Calendar: The selling term is calculated based on the calendar. This is the default method. • Billing Preference: The selling term is calculated based on the billing preferences. You can price transactions based on the term calculated according to the user's preferred start day instead of it always being the first of the month. When the Conga Billing package is installed, you must set the **Selling Term Calculation Method** setting to Billing Preference. CPQ then calculates the selling term based on the billing preferences. For more information, see Billing Preferences.
	❗ The old values supported for this field were *Normal* or *Precise*. Whenever you upgrade CPQ packages, the last value of *Calendar* or *Billing Preference* (added before upgrade) must be retained. If the value of this field changes to *Normal* or *Precise* (old values) after an upgrade, manually change it to *Calendar* or *Billing Preference*. The old values of this field (Normal and Precise) are not automatically removed from existing installations.
Service Line Split Criteria	Defines the criteria to clone the service line.

Setting	Description
Service Price Distribution Method	Enter a value to define the mode used for service pricing. The valid values are: • **Rollup:** Considers all entries of asset components and service line items and the rolls up the final value. Enter this value when the customer has a need to pool and price related equipment with different service prices. • **Allocate:** If values for different services are different and user clicks **Relate** and adds another option, CPQ takes the price of the matrix entry and then distributes to the entire service price. Enter this value when the customer has a need to create a service line only for similarly priced related equipment line items. In this case, the dimensions from the first set of the related line items are used to price the service.

Setting	Description
Show Admin Sidebar	Hides the sidebar for the following Admin pages: • 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
Show Attributes in Cart	⚠ This setting is deprecated.
Show Header	Displays the header tabs and the sidebar on the custom pages.
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).
Show Radio Buttons for Asset	Enable to display radio buttons for asset actions Renew, Increment, Amend, and Cancel.

Setting	Description
Show Tab View	Displays option groups as tabs instead of sections in the options page. You can enable the Show Tab View setting at a product level also. Select the checkbox on the product details page.
Skip Constraint Decision Field	Indicates that fields on the line item skips the Constraint Rule check. Enables bypassing constraint and finalize quote even if there are constraint error.
Skip Review	Skips the review step for items in the cart.
Split Cart Criteria Fields	Defines the criteria to break the summarized view of items added during a large cart process. List the API names of the line item fields separated by a comma or new line. Enter the field names whose values do not change after the records are created for a cart.
Split Cart Threshold	Indicates the number of line items, when exceeded a cart is created. When you define the threshold as 20 and the cart has 30 line items, a cart is created for the remaining 10 line items. This setting is applicable for *Split* quotes. The default batch size for order line item creation and order activation is 100 unique bundle lines. This means 100 unique products on the cart (each product can have multiple line items).
Static Criteria Fields	Defines the criteria to break the summarized view of items added during a large cart process. List the API names of the line item fields separated by a comma or a new line. Type the field names whose value does not change after the records are created for a cart.
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.

Setting	Description
Totaling Group Type	The totaling group preference. The valid values are the following: • *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.
Update View Category Batch Size	Defines the number of categories to update views in a single transaction. Enter a number to define the batch size.
Update View Product Batch Size	Defines the size of the batch to update views of products in a single transaction. Enter a number to define the batch size.
Update View Use Dml Limit	Indicates whether the DML limit is used to determine the workload of a batch to update views of products.
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.
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 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 Page	Type the Visualforce page name for default cart page.

Setting	Description
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.

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.

The behavior listed below is true when you split the bundle ramps.

Scenario	Flags selected	Result
If **Price Type** is *One Time* on Line Item	Hide Dates For One Time (If selected)	Hide **Start Date** and **End Date** on Line Item.
	Hide Dates For One Time (If not selected)	**Start Date** and **End Date** are visible.
If **Price Type** is *Recurring* or *Usage*	Hide Evergreen End Date (If selected) and AutoRenewalType = Evergreen on Line Item	**End Date** is hidden.
	Hide Evergreen End Date (If not selected) and AutoRenewalType = Evergreen on Line Item	**End Date** should behave the same way it behaves for recurring.

Config User Preferences

This custom setting holds Configuration and Pricing user preferences.

Setting	Description
Catalog Products Per Page	Enter the number of catalog products that you want to display on the Catalog page. Valid values are: 10, 20, 50, and 100.
Category Preference	Enter the Category preferences for the user or profile. Enter the list of category names separated by comma.

Setting	Description
Collapse Error Message	Select this setting to collapse error messages.
Collapse Info Message	Select this setting to collapse info messages.
Collapse Quick Add Filter	Select this setting to collapse the quick add filter by default.
Collapse Warning Message	Select this setting to collapse warning messages.
Flow	Enter the name of the default flow associated with the user.
Groups Per Page	Number -
Items Per Page	Enter the number of browsed or searched products to be displayed on the following pages: • On the product selection page • On the cart grid for showing line items (if pagination is enabled) • On the promotions pop-up • On the Mini-Cart • On the assets grid
Logging Level	Enter the logging level for JavaScript code. Valid values are: Debug, Info, and Error.
Option Items Per Page	Enter the number of displayed option products to show in the cart page. Valid values are 5, 10, 15, 20, and 25.
Pin Option Group by Default	Select this setting to control the initial state of option group tabs, either expanded by default when enabled or collapsed by default when disabled.
Selected Comparison Products	Enter values to maintain selection of compared products, between the Catalog and the Compare Features page.
Selected Products Per Page	Enter the number of selected products that you want to display on the configuration pages. Valid values are 5, 10, 15, 20, and 25.

Installed Products Settings

Custom settings for the Installed Products page in CPQ.

Setting	Description
Account Hierarchy Batch Size	Enter the number of Account Hierarchy records processed for each batch execution. Recommended value is 500. Valid values range from 1 to 2000.
Alert Asset Related To Renewal Cart	Select this setting to display a warning on the Installed Products page when the user tries to manually renew an asset associated with a system-generated renewal quote or renewal agreement. The setting is not selected by default.
Allow Backdated Termination	This setting is disabled by default, that means, user cannot 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. Select this setting to enable user to specify a termination date earlier than today. This may have additional impact on billing because an invoice for the period may have already been generated.
Allow Mass Change	Select this setting to enable user to perform mass changes for *must configure* assets.
Amend Change Fields	Enter 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.
Apply Adj To Current Contract Term	Select this setting to keep the discounting limited to the current contract term. When you enable this setting, CPQ applies adjustments and calculates the net price and net unit price of the assets based on the current contract term only. When you disable this setting, CPQ applies adjustments and calculates the net price and net unit price of the assets based on the original start date of the asset.

Setting	Description
Asset Currency Field Precision	Enter the number of decimal places for the asset currency fields. (used by the formatfields function)
Asset Restore Fields	Enter comma-separated API names of the asset fields that CPQ must restore from the original asset on reselection of option line items on the Configuration page during ABO operations. When the sales representative deselects an option and reselects it on the Configuration page during any ABO operation, CPQ restores asset values for the fields added in the **Asset Restore Fields** setting, for the option line item on the Cart page. However, there is an exception to restoring start and end dates during the Renew operation. CPQ restores the renewal start date and end date on reselection of an option (not the asset start and end dates) during the Renew operation. However, for this feature to work: • This setting must have at least one asset field. • The same fields must be available on the asset line items and cart line items. ⓘ Due to high volume of attributes, CPQ does not restore attribute and extension attribute asset fields from the original asset on reselection of option line items during ABO operations.

Setting	Description
Asset Source	Enter the account source which the asset line items will be retrieved from.
	This setting allows you to filter the assets to be displayed on the Installed Products page based on the **Ship To** or **Sold To** or **Bill To** fields of an account. By default, all the assets filtered with the Sold To field are displayed. Specify the API name of the Account field that you want to use as a source for displaying assets. For example, to filter the assets based on **Ship To** field, enter Apttus_Config2__ShipToAccountId__c. You can specify all the three account fields and the system considers an OR condition between these fields while filtering. This is useful when an account has different locations for shipping and billing and you want to display only those assets that have been shipped to a certain location.
Asset Termination Fields	❗ Do not use this setting. Instead use the *Asset Termination* option from the **Display Type** picklist. For more information, see Configuring Display Columns Settings.

Setting	Description
Base Price Defaulting Method For Renewal	Enter a value to define how CPQ must default the base price when an asset is renewed. Valid Values are: • Net Unit Price (default if not specified): CPQ renews the asset line with the net unit price. If no value is specified for this setting, this is the default behavior of CPQ during renewal. • Higher Of Contract Unit Price / Net Unit Price: CPQ renews the asset with the higher value of contract unit price of net unit price of the asset. • Lower Of Contract Unit Price / Net Unit Price: CPQ renews the asset with the lower value of contract unit price of net unit price of the asset. ⓘ Other asset operations still use the net unit price for loading the asset line items. CPQ applies this setting to bundle and its options similarly (that means, a bundle cannot have a higher price while its options have a lower price). You can define this setting at a flow level; however, the flow-level setting overrides the setting defined at **Primary Settings**.

Setting	Description

Cotermination Preferences During Renewal	Enter one or more of the following values (comma separated values, without space) to allow the user to determine renewal end dates.
	
	• UseProposalOrAgreementEndDate: This option enables users to renew an asset using the proposal or agreement end date. CPQ displays the **Use Proposal End Date** option if the user renews assets using the quoting flow and the **Renewal Business Object Type** is Proposal. CPQ displays the **Use Agreement End Date** option if the user renews assets using the agreement flow and the **Renewal Business Object Type** is Agreement.
	
	  ⓘ If the user performs asset renewal through the Asset Manager flow or CSR flow, the **Use the Proposal End Date** or **Use Agreement End Date** option is not displayed on the Confirm Renewal intermediate page.
	  If the user performs asset increment through the Asset Manager flow, the **Use Agreement End Date** option is not displayed on the Change Quantity intermediate page.
	
	• RetainCurrentAssetEndDate: This option enables users to retain the current asset end date while renewing an asset.
	• UseFarthestAssetEndDate: This option enables users to use the farthest asset end date while renewing assets.

Setting	Description
	• UseRenewalDate: This option enables users to renew assets with a custom renewal date.
	You must enter at least one value in this setting. If you have more than one value, you must also select a default value in the **Default Renewal Cotermination Option** setting. If you do not select any default value, the first of the selected values is considered as the default value.
	Depending on the selected setting, users can now bypass the Confirm Renewal intermediate page altogether.
	• If there is only selection for co-termination (apart from *UseRenewalDate*), the Confirm Renewal intermediate page is completely bypassed. On clicking **Renew**, CPQ creates the renewal line and refreshes the Installed Products page to show the detail in the Mini Cart. • If there is more than one selection for co-termination, CPQ displays the Confirm Renewal intermediate page to allow the user to make a selection. • If the selected option is *UseRenewalDate*, CPQ displays the Confirm Renewal pop-up to collect the user input.
Create Renewal Opportunity	Select this setting to enable CPQ to create a renewal opportunity for system generated asset renewal quotes.
	This setting indicates if you want to create renewal opportunities along with renewal quotes. By default, this check box is not selected. This is to prevent the system from running into locking issues when an account has a large number of assets and the batch process creates the renewal opportunities for all these assets.
	When this setting is true, the Create Renewal Opportunity scheduled job runs and creates the renewal opportunities along with renewal quotes.
	When this setting is false, the Create Renewal Opportunity scheduled job runs and does not create the renewal opportunities and creates only the renewal quote.
Default Cotermination Option	Enter the default end date preference for cotermination.

Setting	Description
Default Renewal Cotermination Option	Enter the default end date preference for renewal cotermination. If there is only one value in the **Cotermination Preferences During Renewal** setting, that option is selected by default. If there are more than one options, you can select a value as a default renewal cotermination option. Based on the default option, CPQ loads the selected renewal transactions on the Installed Products page, Confirm Renewal popup, or Confirm Renewal intermediate page.
Default Renewal Price List	Enter the name of the Price List which is a mandatory field for Quote creation.
Default Renewal Term	Enter 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*.
Editable Fields for Cancelled Lines	Enter comma-separated API names of the fields that you want to make editable for cancelled lines on the cart (during asset termination). You can enter a custom currency field that is not based on a formula. You cannot enter out-of-the-box currency fields such as base price and base extended price as editable fields.

Setting	Description
Enable One Time Change	Select this setting 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 are 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. 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.
End Date preferences for Cotermination	Enter comma separated values (without space) to enable the user to select one of the following end dates while coterminating incremental licenses. • RetainCurrentAssetEndDate • UseCustomDate • UseProposalEndDate ⓘ If the user performs asset increment through the Asset Manager flow, the **Use the Proposal End Date** option is not displayed on the Change Quantity intermediate page.
Filter Fields	Enter 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. Commonly used fields are **Start Date**, **End Date**, **Lead Time Expiration**, and **Asset Status**. Usually, all fields under Asset Line Item object are supported but you must enter *ExpirationLeadTimeInDays* value in order to display Lead Time Expiration field in the Search pane.

Setting	Description
GroupByFields	Specify the API names of the asset line item fields that must be displayed under the **Group By** drop-down on the Create New View pop-up while Sales Representatives are creating views on the Installed Products page. The same GroupBy API field values will reflect on the **GroupBy Fields** field on the GENERAL SETTINGS tab of the Installed Products settings from the CPQ Admin UI. For more information, see Configuring Installed Products Settings.
Hide Co-Term	Select this setting to hide the co-term end date on renew. This setting hides the Coterminate with Increment panel on the Change Quantity (Increment Asset) page. The Coterminate with Increment panel is displayed by default.
Max Renewal Group Per Job	Enter the number of renewal groups to be processed by one batch job. This setting limits the number of renewal quotes to be processed in each batch job to avoid too many running batch jobs.
Max Renewal Line Item Per Cart	Enter the number of renewal line items to be processed in one cart. This setting limits the size of product configuration to avoid exceeding CPU and SOQL limits. This setting determines the number of line items that renewal quotes can have in the cart. You can enter a maximum value of 1000. For example, if a cart has 5000 line items and you set **Max Renewal Line Item Per Cart** as 500, there will be 10 renewal quotes.
Max Renews Per Trip	Enter the maximum number of renewal batch jobs per trip. Use a smaller number to prevent CPU timeout. The default is 20.
Merge Action Criteria Fields	Enter comma-separated asset line item fields that must be common among the asset line items to be merged. The default fields are product, selling term, charge type, and end date.

Setting	Description
Purchase Identification Criteria	Enter comma separated asset line item fields that determine the eligibility for cotermination and quantity cumulation.
	This setting allows user to define a criteria to get coterminate related asset lines in the Coternimate with Increment section on the Change Quantity page during asset increment. The default purchase identification criteria are: Product, Charge Type, Option, and Account. CPQ currently supports only Purchase Identifier and alphanumeric custom fields.
	For example, if you select Purchase Identifier, during increment of a selected asset, all other assets with the same Purchase Identifier are displayed on the Coternimate with Increment section.
	Based on the selected purchase identification criteria, the total quantity of an asset spans multiple streams during the increment scenarios, on the Installed Products page (of a renewal quote).
Relate Action Criteria Fields	Enter comma separated asset line item fields to enable relate asset action. This custom setting shows all fields of Asset Line Item and Price Dimensions defined on asset line item. When the user selects two or more assets and clicks **Relate**, CPQ validates the selected asset line items against the Relate Action Criteria Fields. In case of any discrepancies, CPQ displays a validation error message to the user.
Renewal Business Object Type	Enter the Business object for which this renewal is taking place.
	Valid values are: *Proposal* and *Agreement*
Renewal Default Price Book	Enter a Renewal Price Book name which you want to associate with renewals.

Setting	Description
Renewal Execution Mode	This setting indicates if the Renewal of Asset Line Items must happen automatically or based on 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*.
Renewal Group Fields	Enter the API names of the fields you want to use for grouping. This setting indicates how to group the Asset Line Items when a Renewal Opportunity is created. 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	Enter a value when the renewal quote must be created after an order is activated. By default, the value is 0, which indicates that the renewal quote is created immediately after the order is activated, with the same number of Line Items in the order. If you specify 30, the renewal quote is created 30 days before the Asset End Date.

Setting	Description
Renew One Ramp	Select this setting 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. 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.
Show Accounts Filter	Select this setting to display accounts filter in the Installed Products page. This is currently not supported.
Show Assets	This setting determines whether current, parent, or child assets of the current asset's account are displayed. Enter comma separated values of Parents and/or Children. If empty, only context account assets are shown. Examples: • parents • parents,children
Show Service Coverage	Select this setting to display service coverage for a primary service.
Split Asset Actions	Enter the actions that can be performed on a split asset. This setting enables the user to perform split asset actions on the Define Split <asset name> page. Valid values are: • Renew: Enables users to perform the Split and Renew asset action. • Swap: Enables users to perform the Split and Swap asset action.

Setting	Description
Submenu Actions	Enter comma-separated submenu actions that you want to display when the user clicks an action button on the Installed Products page. You can configure submenu actions only on the **Change** and **Relate** actions. The supported values are: • **Change:Configuration** displays the Configuration submenu under the Change action. This submenu enables users to configure an asset line item. • **Change:Quantity** displays the Quantity submenu under the Change action. This submenu enables users to initiate an action to add more licenses for an existing asset stream. • **Change:Split** displays the Split submenu under the Change action. This submenu enables users to split an asset into multiple split lines. • **Change:Merge** displays the Merge submenu under the Change action. This submenu enables users to merge multiple asset lines into a single asset. • **Relate:Component** displays the Component submenu as a drop-down under the Relate action. This submenu enables users to relate an asset component with a service.

Lookup Field Settings

Controls lookup field behavior in CPQ. Lookup field settings provide an ability to filter lookup records through filter criteria based on the immediate parent on which the lookup field is defined.

Setting	Description
Enable Quick View	Select this to enable CPQ to pre-populate the values in lookup fields on the Cart page and Mass Update popup. If you disable the setting, the Sales Rep needs to search the values. CPQ performance is optimized if you disable this setting as the lookup values are retrieved on the Cart page or Mass Update popup only when the Sales Rep clicks the search icon. Conga recommends disabling the setting for lookup fields with a large number of lookup records and also only enable the setting for a maximum of 2 lookup fields that are present on the cart.

Setting	Description
Filter Criteria	Enter a formula to filter lookup values. The formula must be a valid expression to further filter down the search results for a lookup. The field size is 256 characters. Formulas are built with left as a lookup or junction and right as the object. Example: Product__c=Apttus_Config2__ProductAttributeValue__c.Product_Id__c This formula fetches only such attributes whose IDs match with the selected products. Here, the ProductId is a custom formula field that fetches the ProductId of the Line Item object.
Filter Criteria 2	Enter a formula to be used as an extension to **Filter Criteria**. Contents of this field are concatenated to the content of **Filter Criteria** without any validation while evaluating. ⓘ There is no space between the expressions when they are concatenated because the leading and the trailing spaces are trimmed. Avoid splitting the expression where space is required. Split the expression mid-way between a word to avoid trimming the space at the beginning or end of a word. For example, **Filter Criteria** = *AccountId_c = '00137000002ttoF' AN* **Filter Criteria 2** = *D TaxExempt_c = 'Yes'*. After the fields are concatenated the expression becomes *"AccountId_c = '00137000002ttoF' AND TaxExempt_c = 'Yes'"*. *Otherwise, when Filter Criteria = AccountId_c = '00137000002ttoF' AND* **Filter Criteria 2** = *TaxExempt_c = 'Yes'*. After the fields are concatenated the expression becomes *"AccountId_c = '00137000002ttoF' ANDTaxExempt_c = 'Yes'"* and results in a syntax error. Note that there is no space between *'AND'* and *'TaxExempt_c'*
Junction Default Flag	Enter the name of a field API from the Junction object to determine the default value of a lookup field. This field must be a checkbox.
Junction Field Name	Enter the name of a field on the junction object which you would like to filter the lookup field by.
Junction Object Name	Enter the name of the junction object which you would like to filter the lookup values. This is a custom object that you can define to filter the values which do not exist on the Lookup Object itself.

Setting	Description
Lookup Display Columns	Enter the names of fields to be displayed on the lookup dialog box. Field names must be separated by comma without spaces. The columns of the Lookup popup fields from the Lookup Object will be displayed by populating with the fields APIs from the Lookup Object such as Name__c.
Lookup Field Name	Enter the name for the lookup field. This is the API name of the Lookup field on the Line Item or Product Attribute Value Object defined under the Object Name.
Lookup Record Limit	Enter the maximum number of records to query for the lookup field drop-down. Default value is 200 to a maximum of 1000.
Object Name	Enter the API name of the object where the lookup field exists. Valid values are: • *Apttus_Config2__LineItem__c* • *Apttus_Config2__ProductAttributeValue__c* • *Apttus_Config2__ProductConfiguration__c.* ⓘ You can define lookup fields with filter criteria from Product Configuration object in Lookup Fields Settings. This allows you to use lookup fields with filter criteria in features like Mass Update

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** checkbox for Product Id and 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 a valid expression to further filter down the search results for a lookup. The field size is 256 characters. When the expression exceeds the size, use **Filter Criteria 2**.
	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
Filter Criteria 2	This field is used as an extension to Filter Criteria. Contents of this field are concatenated to the content of "Filter Criteria" without any validation while evaluating. ⚠ There is no space between the expressions when they are concatenated because the leading and the trailing spaces are trimmed. Avoid splitting the expression where space is required. Split the expression mid-way between a word to avoid trimming the space at the beginning or end of a word. For example, **Filter Criteria =** *AccountId_c = '00137000002ttoF' AN* **Filter Criteria 2 =** *D TaxExempt_c = 'Yes'*. After the fields are concatenated the expression becomes "*AccountId_c = '00137000002ttoF' AND TaxExempt_c = 'Yes'*". Otherwise, when **Filter Criteria =** *AccountId_c = '00137000002ttoF' AND* **Filter Criteria 2 =** *TaxExempt_c = 'Yes'*. After the fields are concatenated the expression becomes "*AccountId_c = '00137000002ttoF' ANDTaxExempt_c = 'Yes'*" and results in a syntax error. Note that there is no space between '*AND*' and '*TaxExempt_c*'
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

Proposal System Properties

Quote/Proposal System properties.

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 in the following format: Format = first name, last name
Auto Create Cart Version After Finalize	Select this setting to enable CPQ to create a new version of the cart automatically after it is finalized.
Auto Create Order	Select this setting to enable CPQ to create an order and assets automatically when the proposal is accepted.
Auto Select Attachment?	Select this setting to enable CPQ to select the most recent attachment automatically.
Auto-Select multiple recipients?	Select this setting to enable projects to specify a list of recipients that will be defaulted for eSignature. To use this option, projects will add the **DocuSign Default Recipients** related list on proposals and user will select the list for each proposal.
Auto Select Recipient Field Name	Enter the API name of the field which holds the value for the recipient to select automatically.
Auto Sync With Opportunity	Select this setting to enable CPQ to synchronize the proposal automatically with the opportunity when the proposal is accepted. This setting automatically synchronizes the quote line items with the opportunity when the proposal is finalized. This setting appears only if you haveConga Quote Configuration Integration package installed.
Bypass Sharing	Select this setting to enable apex code to bypass record sharing.
Clone With Approval Status	Select this setting to enable CPQ to retain approval status on the cart and line items associated with the cloned quote/proposal.

Setting	Description
Create Agreement With Approval Status	Select this setting to enable CPQ to retain the approval status on the quote. When an agreement is created from the quote, the approval status on the quote is transferred to agreement line items associated with the quote.
CSS Override	Enter the name of the static resource to override CSS on the Proposal document generation page.
Custom Doc Gen URL	Enter the custom URL that is accessed when using the custom button on the DocGen page.
Default Opportunity Quote Owner	Enter the owner name to set the default owner for the quote/proposal created from an opportunity. Valid values are: Opportunity Owner Current User If not set, *Opportunity Owner* becomes the owner of the new quote/proposal.
Default Quote Agreement Owner	Enter the owner name to set the default owner for the agreement created from a quote/proposal. Valid values are Quote Owner Current User If not set, *Quote Owner* becomes the owner of the new agreement.
Default Template Name	Enter the name of the default email template that must be used when sending emails.
Disable Opportunity Products Copy	Select this setting to disable copying of opportunity products to the quote/proposal.

Setting	Description
Document Naming Convention	Enter a value to apply a custom naming convention for all proposal documents at generation. The following attributes are permitted when formulating a document naming convention: • %templatename% • %user% • %timestamp% • Proposal attributes such as %:Name%. ⓘ Any variable prefixed by ':' represents a field on the Proposal object. If the property contains a null value (i.e., if left blank), the current default naming convention will be used: "%:Proposal_Name_c%_%templatename%_%timestamp%". Example of a proposal document name using the default naming convention: SOW_Regenerated_SOW ABC_2019-08-07
Email Template For Presenting Proposals	Enter the name of the email template that must be used for presenting proposals.
Enable Document Preview in DocuSign App	Select this checkbox to enable document preview in DocuSign application.
Enable Fast Doc Gen	Select this setting to enable the automatic generation of the proposal document upon the click of the **Send Proposal** button on the Quote Details page.
Enable File	Select this setting to enable use of Files (Salesforce). This setting also enables preview on the Doc Gen page. ⓘ After you select this setting, it is mandatory to run Migration, which moves the documents in Notes & Attachments from CPQ to Salesforce Files.

Setting	Description
Enable PDF Security	Select this setting to enable users to apply security settings to PDF documents and protect them with a password.
Enable Proposal Email Editing	Select this setting to enable editing of the Proposal email template.
Enable Proposal Summary	Select this setting to enable the proposal summary object.
Enable Submit Merge Call	Select this setting to submit merge calls for processing. If this setting is enabled, the **Submit** button is displayed during document generation.
ESignatureStrictMode	RESERVED Select this setting to disable buttons for Select Attachments / Recipients.
Instance Url	Enter the Salesforce instance URL.
Large Doc Process Batch Size	Indicates the number of repeating child items to process in a single batch transaction. This setting is applicable for *Enterprise* quotes. The default batch size for order line item creation and order activation is 100 unique bundle lines. This means 100 unique products on the cart (each product can have multiple line items).
Large Doc Threshold	Enter a number of the line items in the proposal when exceeded indicates that the proposal is a large document. Enter *-1* to deactivate the threshold.
Max Child Level	Enter the maximum level to generate the merge data for the proposal.
Merge Call Timeout Millis	Enter the timeout in milliseconds for the merge request. For example: 60,000
Merge Webservice Endpoint	Enter the Conga merge webservice endpoint.

Setting	Description
PDF Owner Password	Enter the password required to change permissions of the PDF document like printing or editing.
Sort Documents Descending	Select this setting to sort the generated documents on the Doc Gen page in descending order of Created Date. Enabling this, CPQ previews the document that was generated last. ⓘ If you are upgrading from the May '22 or September '22 to February '23 Preview release of CPQ, you must remove the **APTS_SortDocumentsDesc** admin setting. The admin setting is no longer supported. For more information, see Configuring Admin Settings.
Sync Bundle Using Line Items	Select this setting to use proposal line items to synchronize bundle products. By default, CPQ uses proposal summary objects. This setting appears only if you have Conga Quote Configuration Integration package installed. ⓘ This field is not supported for the quotes where **QTC Profile** is set to *Split*. Regardless of the value of the field, proposal line items are used to sync Spilt Quote with Opportunity.
Sync Option Products	Select this setting to enable CPQ to synchronize the option products with the bundle. This setting synchronizes options with bundles when you click **Sync with Opportunity** for your proposal. This setting appears only if you have Conga Quote Configuration Integration package installed.

Configuring Custom Settings for Different Flows

You can create different data set of Config System Properties and Installed Product Settings for different flows in the org. When you create such a data set, the definition of

these properties in the data set is only valid in the Quotes configured using that flow. You must create the data set with the same name as the flow.

The following fields can be defined differently for different flow:

- Split Cart Threshold
- Split Cart Criteria Fields
- CSS Override
- Service Line Split Criteria
- Constraint Rule Execution Mode
- Pricing Execution Mode
- Custom Pre Pricing Fields
- Base Price Defaulting Method For Renewal
- Hide Asset Actions
- Check Many Options
- Cart Line Items Retrieve Chunk Size

To create a new data set for Config System Properties

1. Go to **Setup > App Setup > Develop > Custom Settings.**
2. Click **Manage** next to Config System Property.
3. Click **New**.
4. In the **Name** field, enter the name of the flow for which you want to create the data set.
5. Configure the settings according to your needs. For information on the settings, refer to the topic Configuring Custom Settings.
6. Click **Save**.

For more information about configuring Installed Product Settings for a different flow, refer to Defining Allowed Actions for Assets.

About Custom Setting Order of Precedence

There are different 3 ways you can define a custom setting, from default custom setting dataset, flow-based custom setting dataset, and URL parameters of the **Configure Products** button. There is an order of precedence based on where the custom settings are defined that CPQ follows while executing them. The order of precedence of custom settings execution is described below.

- Settings defined in the default custom settings dataset are executed in absence of flow-based dataset or URL parameters. They are also executed if an invalid definition of the setting is configured in flow-based dataset or URL parameters.

- Settings defined in the flow-based custom setting dataset overrides the settings in the default custom settings dataset. These are executed in the flow they are defined for.
- Settings defined in the URL parameters in the **Configure Products** button override both default and flow-based custom settings dataset.

Configuring Field Set Settings

The Field Set Settings page allows the System Administrator to configure field sets.

To view or edit field set settings:
- Go to Config Settings > Field Set Settings. A Config Field Set Edit page is displayed.

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

Setting	Description
Name	Select a Field Set from the drop-down list. The available Field Sets are as follows: • Collab Create Fields • Collab Merge Fields • Line Item Fields to Revalidate Invalid Price List Items • Line Item Fields to Revalidate Product Structures • Related Line Item Fields • Asset Line Fields For Selected Assets • View Cart Total Custom Fields
FieldName	Select a field that you want to add to a Field Set. For example, you may want to add Record ID, Created Date to the Collab Create Fields. These fields are populated in the drop-down lists from the corresponding objects. For example, Asset Line Item object for the asset field set. Click + symbol to add a field and - symbol to remove a field from the field set. Click Up and Down arrows in front of each field to set a sequence in which fields to be displayed in the field set list. Click Save to set your configuration.

Each field set is meant for a specific business scenario.

Field Set	Scenario
Collab Create Fields	In a multi-tier collaboration scenario, specify the additional fields whose value you want to copy over from parent line item to child line item when creating a collaboration request. For more information, see About Quote Lifecycle Collaboration.
Collab Merge Fields	In a multi-tier collaboration scenario, specify the additional fields whose value you want to copy over from child line item to parent line item when merging a collaboration request. For more information, see About Quote Lifecycle Collaboration.
Line Item Fields to Revalidate Invalid Price List Items	When you select the Price List Item field for the Line Item Fields to Revalidate Product Structures set, then you can have a validation based on the price list item. For more information, see About Quote Lifecycle Collaboration.
Line Item Fields to Revalidate Product Structures	When you select the Price List Item field for the Line Item Fields to Revalidate Product Structures set, then you can have a validation based on the price list item. For more information, see Revalidating the Product Configuration.

Field Set	Scenario
Related Line Item Fields	Select the fields that you want to display on the Related Line Items pop-up. This pop-up is displayed when the user hovers the mouse on a service name (bundle or option), clicks the more icon (⋮), and selects Related Line Items on the Cart. For example, when you select the Weightage Type field for the Related Line Item Fields set, then you can define the Percentage and Amount values for the Related Asset Line Items pop-up. The Related Asset Line Items are associated with a Service Product. For more information on the Service CPQ, see Managing Services.
Asset Line Fields For Selected Assets	Select the fields from Asset Line Item object that you want to display on the Selected Assets or Manage <Service_Option_Name> pop-up. You can display a maximum of 10 fields in the pop-up. This pop-up is displayed when the user clicks the Selected Assets link on the Service Catalog or when the user selects a service option and clicks the 🔲 icon on the Service Config page. ⓘ The asset name is displayed next the checkbox on the Manage <Service_Option_Name> pop-up by default. You need not add the Asset Name field in the Asset Line Fields For Selected Assets setting. For example, when you select the Location field for the Asset Line Fields for Selected Assets setting, then the user can view the location at which the product was sold to the customer. For more information on the Service CPQ, see Managing Services.

Field Set	Scenario
View Cart Total Custom Fields	When you select the View Cart Total Custom Fields, then this setting helps you for the addition of more custom fields that were to be added in View Cart Total Custom Fields 2 under system properties. 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. For more information, see Configuring System Properties.

Smart Search Setup

You can use the Smart Search feature to search for products in large and complex product catalogs with multiple levels of categories and stricter rules. The smart search typeahead mechanism and faceted search help narrow the search results and still honor product visibility and availability rules.

You must setup remote site before you configure settings for Smart Search.

Setting up Remote Site Settings

Navigate to **Setup > Administration Setup > Security Controls > Remote Site Settings** and click **New Remote Site** to create the following records:

Record number	Remote Site Name	Remote Site URL
1	SmartSearch	https://<SMART_SEARCH_URL> Smart Search URL can be any of the following: • AZURE PREVIEW: smartsearch-prod.congacloud.com/ (Nov 2016 SP1 GA - Version 9.1168 and above versions) For any fresh package installation, Azure URL defaults.
2	SFDC	• For production environment, enter https://login.salesforce.com • For sandbox environment, enter https://test.salesforce.com

Record number	Remote Site Name	Remote Site URL
3	OauthToken	• For production environment, enter https://login.salesforce.com/services/oauth2/token • For sandbox environment, enter https://test.salesforce.com/services/oauth2/token

Creating a Certificate

Navigate to **Setup > Administration Setup > Security Controls > Certificate and Key Management** and click **Create Self-Signed Certificate** to create a new certificate with the following details:

1. Enter *JWT* as the label for the certificate. The Unique Name defaults to *JWT*.
2. Select the **Exportable Private Key** checkbox.
3. Choose Key Size as *2048*.
4. Select the **Active** checkbox.
5. Click **Save**.

Once you have created a certificate, go to your certificate record and click Download Certificate to store the certificate on your local system. You will need the certificate while creating a connected app.

Creating a Connected App

1. Navigate to **Setup > App Setup > Create > Apps.**
2. Scroll down and search for **Connected Apps** related list. Click **New** to create a new app.
3. Enter the basic information, **Connected App Name** and **API Name** as *CongaSmartSearch.*
4. In the API (Enable OAuth Settings) section, select **Enable OAuth Settings** checkbox.
5. In the Callback URL, enter https://*<login OR test>*.salesforce.com/services/authcallback/*<org-id>*/CongaSmartSearch
   In *<login OR test>* enter login or test depending on your environment. In **<org-id>**, enter your org ID.
6. Select the **Use digital signatures** check box. Choose the file from generated certificate.
7. For **Selected OAuth Scopes**, select the following:

- Access and manage your data (api)
- Access and manage your Chatter data (chatter_api)
- Perform requests on your behalf at any time (refresh_token, offline_access)

8. Click **Save**.

Approving your Connected App

Before approving your connected app, you must decide which Oauth policy you would like to have. This policy decides which permitted users can work on Smart Search's admin page. Admin approved users are pre-authorized. You can also authorize other users and for such users, the steps mentioned in Approving your Connected App procedure are mandatory.

Step 1: Managing Profiles for Connected App

1. Navigate to **Setup > Administration setup > Manage Apps > Connected Apps**.
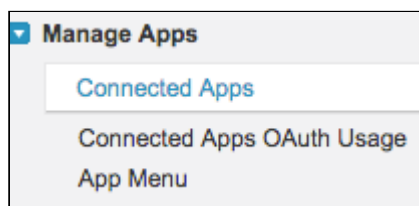
2. Next to the CongaSmartSearch connected app record, click **Edit.**
3. Under the OAuth Policies section, in Permitted Users drop-down list, choose *Admin Approved Users are Preauthorized* and click **Save**.
4. Scroll down and search for Profiles section. Click **Manage Profiles** and select the profiles that you want to give access to the Smart Search App.
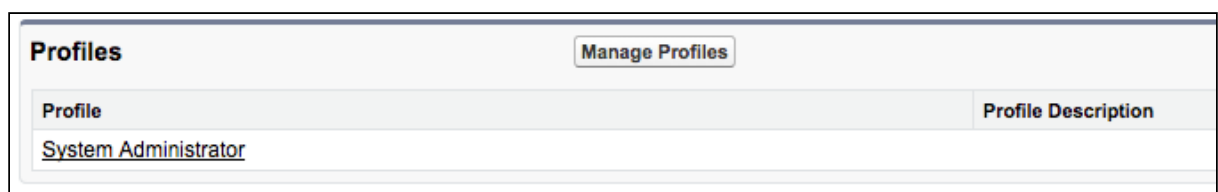
5. Click **Save.**

Step 2: Approving your Connected App

1. Navigate to **Setup > App Setup > Create > Apps.**

2. Scroll down and search for **Connected Apps** related list.
   Click your *CongaSmartSearch* record and save the values of the Consumer Key, Consumer Secret, and Callback URL fields in a text file. You will need these keys to set up the last step for smart search.
3. Go to your browser URL, paste the following URL and click **Enter**:
   https://*<login OR test>*.salesforce.com/services/oauth2/authorize?
   response_type=code&client_id=**<consumer-key>**&redirect_uri=**<callback-url>**

   Use the correct domain for **<login or test>**, enter your consumer key and callback URL which you have fetched in the above step in **<consumer-key>** and **<callback-url>**.
   Note that `<callback-url>` needs to be encoded.

4. Click **Allow** to give permission to your connected app.
5. You will see the following page, which is the expected behavior of the app. You can now close the browser tab.
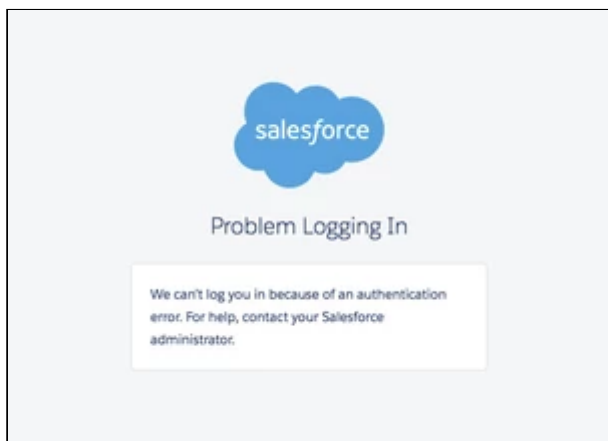
Smart Search Settings

You must follow the steps below to configure Smart Search.

> ⚠ Contact Conga Product Support team to enable the Smart Search capability for your org.

Prerequisite

Setup the remote site for Smart Search. Refer to Smart Search Setup.

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 Conga Intelligent Cloud Search Connection Settings

Please contact Conga 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.

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%. You can provide 100% weightage for a single product field 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 there is a match for a search text 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, you must 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. You can add product fields such as Product Name, Description, Product Code, and more.

3. Select the desired weightage for the product field in percentage using the **Weightage** drop down.

4. Select the **Typeahead** checkbox to apply typeahead for the selected field. For example, if you define typeahead on product name, when the sales representatives enter a keyword in the Search bar, they can see the typeahead results if the product name matches.

> ⓘ If you add multiple fields for product field weightage, you can apply **Typeahead** only for one product field.

5. Click **Add Row**, if you need to add an additional field.

> ⓘ   • Ensure that the total weightage sum of all the fields equals to 100%.
>   • CPQ supports fields of type text, picklist, rich text, and number. It can also be a mix of all these types of fields.

6. 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.*

> ⚠ 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.

Migrating a Workflow Rule to Flow

As Salesforce is starting the process of moving away from Workflow Rules and transitioning to Flow, Conga recommends you temporarily migrate Workflow Rules in CPQ to Flow.

You must migrate the following Workflow Rules to Flow.

- Set Original Order Line Number
- Set Original Order Number
- Search Field Update
- Set Incentive Code
- Set Incentive Number

To migrate Workflow Rule to Flow using the Salesforce Migration Tool

Salesforce introduced a new migration tool for the hassle-free migration of all your workflows to flows.

1. Navigate to the Workflow Rules.
   - Salesforce Classic users, **Setup** > **Create** > **Workflow & Approvals** > **Workflow Rules.**
   - Salesforce Lightning Users, **Setup** > **Platform Tools** > **Process Automation** > **Workflow Rules**.
     Click **Continue** on the *Understanding Workflow* page. A list of workflow rules is displayed.
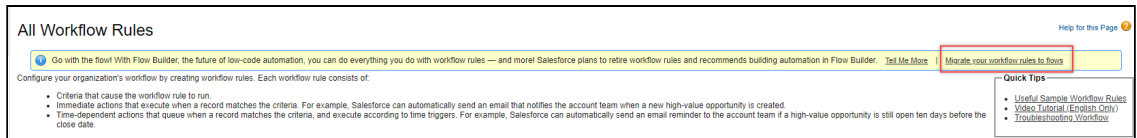
2. Click **Migrate your workflow rules to flows**. A Migrate to Flow page is displayed with a list of workflow rules.
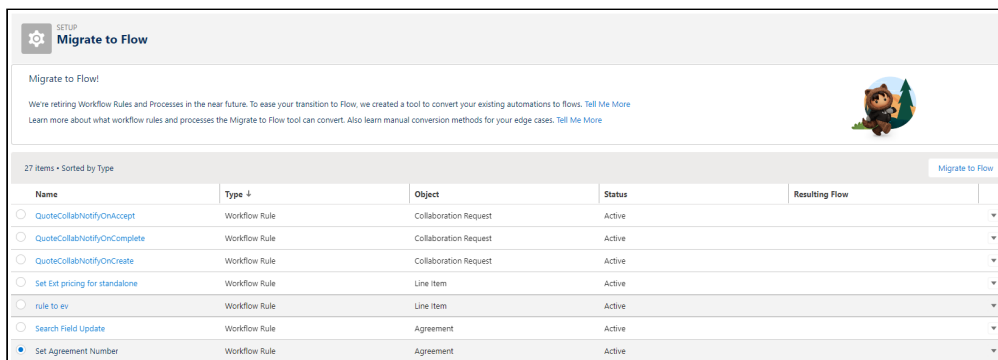
3. Click the radio button to select the workflow rule that you want to migrate.
4. Click **Migrate to Flow**. This action automatically deactivates the workflow rule and activates the flow in your system.
5. To identify the custom flows in CPQ, Conga recommends you add *Custom* as a prefix to the custom flow name. For example, *Custom Set Incentive Code*.

> ⓘ
> - Ensure that the workflow rule is deactivated successfully. To verify whether the *Active* checkbox is unchecked for the migrated workflow rule,
>   - Salesforce Classic users, **Setup** > **Create** > **Workflow & Approvals** > **Workflow Rules**.
>   - Salesforce Lighting Users, **Setup** > **Platform Tools** > **Process Automation** > **Workflow Rules.**

Configuring the Flows

To ensure that the flow operates similarly to the workflow rule, complete the following configuration steps for the migrated flows.

1. Navigate to the Flows.
   - Salesforce Classic users, **Setup** > **Create** > **Workflow & Approvals** > **Flows**.
   - Salesforce Lightning Users, **Setup** > **Platform Tools** > **Process Automation** > **Flows**.
2. Click the flow name. A Flow Builder page is displayed.
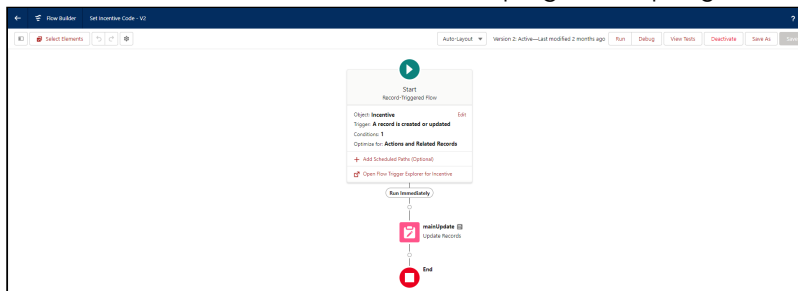
3. Click **Edit** beside the *Object* Name. A Configure Start popup window is displayed.
4. In the *Optimize the Flow for:* section, select **Actions and Related Records** and click **Done**.
5. Click **Save As** and **Save**.
   The changes are saved as a new version.

Working with Flow Builder

To create a Flow using FLow Builder

1. Navigate to **Setup** > **Create** > **Workflow & Approvals** > **Flows**.
2. Click **New Flow**. A *New Flow* popup window is displayed.
3. Select the type of flow and click **Create**. A Flow Builder page is displayed. You can create a flow using one of the following methods.
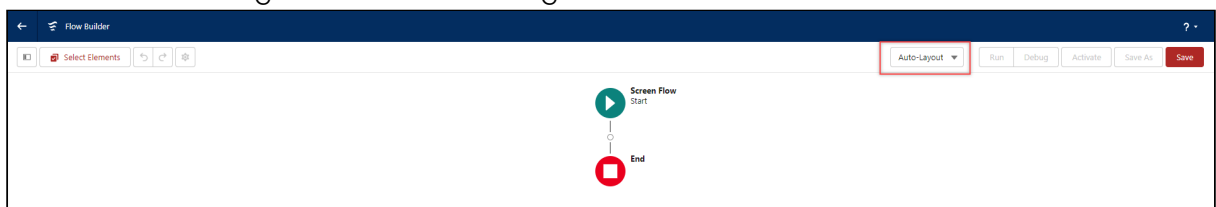
   a. **Auto-Layout**: By default, Auto-Layout is selected where you can add elements to the flow by clicking the **+** icon between the start and End icons on the canvas

to configure them. For more information, see https://help.salesforce.com/s/articleView?language=en_US&id=sf.flow_builder.htm&type=5.

b. **Free-Form**: This option opens the *Toolbox* where you can view screen elements that can be added to the canvas.

4. For Free-Form,

   a. Click and drag a Screen element from the Palette in the left pane onto the canvas to configure it.

   b. Click and drag a Record Update element from the Palette in the left pane onto the canvas to configure it.

   c. Click **Save**.

   d. In **Flow Properties**, enter a mandatory **Name** and press TAB. The Unique Name is auto-populated.

   e. 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

1. Navigate to **Setup** > **Create** > **Workflow & Approvals** > **Flows**.
2. Open the flow to be updated with a new variable.
3. Go to the **Manager** tab in the left palette and click **New Resource**. A *New Resource* pop-up window is displayed.
4. Select the **Variable** from the *Resource Type* dropdown list.
5. Enter the **API Name**, Description, and DataType.
6. Click the *Allow multiple values (collection)* checkbox if required.
7. Click **Done**. A new variable is created successfully.

To add a screen element

1. Navigate to **Setup** > **Create** > **Workflow & Approvals** > **Flows**.
2. Click **Create Flow** or open an existing flow.
3. Select the *Screen* from the palette of available elements on the left pane and drag it onto the flow builder canvas. A New Screen popup window is displayed.
4. In the Screen Properties, enter the Lable name. The Unique API Name field is auto-populated.
5. Enter the description.
6. Configure *Header* and *Footer* as required.
7. To add components to the new screen, click the required component in the left pane and configure them based on your business requirement.
8. Click **Finish**.

The screen element is added.

Upgrading CPQ

This section provides information on upgrading CPQ to the latest version from the previous two releases.

⚠️ Before you upgrade the CPQ packages in your org to the February '24 release, please note the following points:

- If the CPQ package in your org is on any of the following releases, you must upgrade it to the Winter '20 patch (Winter20.02.26) before upgrading it to February '24 release:

Release	Package
Spring '20 or earlier release	Packages older than **12.0.1719 | 12.1719**
Summer '20 patch older than Summer20.02.26	Packages older than **12.1.1787.99 | 12.1787.99**
Winter '20 patch older than Winter20.02.26	Packages older than **12.2.1839.48 | 12.1839.48**

- If the CPQ package in your org is on any of the following releases, you can directly upgrade it to the February '24 release:

Release	Package
Latest Summer '20 patch (Summer20.02.26 or later)	**12.0.1719 | 12.1719** or later
Latest Winter '20 patch (Winter20.02.26 or later)	**12.2.1839.48 | 12.1839.48** or later
Spring '21 and any Spring '21 patches	**13.0.1882 | 13.1882** or later
Summer '21 and any Summer '21 patches	**13.1.1921 | 13.1921** or later
December '21 and any December '21 patches	**13.2.1969 | 13.1969** or later
May '22 and any May '22 patches	**14.0.1995 | 14.1995** or later
September '22 and any September '22 patches	**14.1.2038 | 14.2038** or later

Release	Package
February '23 and any February '23 patches	**15.0.2061 | 15.2061** or later
June '23 and any June '23 patches	**15.1.2090 | 15.2090** or later
October '23 and any October '23 patches	**15.2.2106 | 15.2106** or later

- Preparing for Upgrade
- Upgrading to CPQ February '24 Release
- Performing the Post-Upgrade Tasks
- Preparing for Upgrade (UDJS)
- Upgrade Assistant

Preparing for Upgrade

Before you upgrade to CPQ February '24 release, you must ensure the following:

> ✅ The Conga Upgrade Program is an automated tool that upgrades packages available in your Salesforce org (Production or Sandbox) to the latest versions. In addition, It ensures all the Conga published managed packages are on the latest versions for the registered orgs. To register your org for upgrade program, see Registering for Conga Upgrade Program.

- You go through CPQ Features by Release to know about the new features, enhancements, and deprecated features in CPQ since your existing release. After you upgrade CPQ to February '24 release, you cannot roll back to any previous release.
- You have the supported platforms and system requirements.
- You have access to the **Install Center** on the Conga Community Portal.
- You have administrator privileges to your Salesforce org.
- You need not back up your configurations. All configurations you performed since you installed your existing release will remain intact after the upgrade.
- You must delete the UsageDataJobScheduler job because it is no longer supported. You must delete this job once before upgrading to the February '24 release or any succeeding release.

To delete the UsageDataJobScheduler job

Before you upgrade **Conga Base Library** for the February '24 release or any succeeding release, perform the following steps.

1. Go to **Setup** > **Administration Setup** > **Monitoring** > **Apex Jobs.**
2. Search for the **UsageDataJobScheduler** job, CPQ displays the status of the job as *Queued*.
3. Open another instance of the org in a different browser tab, and go to **Setup** > **Administration Setup** > **Monitoring** > **Scheduled Jobs**.
4. Search and delete the **UsageDataJobScheduler** job from Scheduled Jobs. On the Apex Jobs page, the status of the **UsageDataJobScheduler** job now changes to *Aborted*.
5. Install the package version **4.0.242 | 4.242** or higher of the **Conga Base Library** package.
6. After the latest package is installed, check if the **UsageDataJobScheduler** job gets scheduled again on both the Apex Jobs and Scheduled Jobs pages. If not, the latest package has successfully disabled the scheduling of this job and has deprecated it.

To updating the help text of Billing Start Date and Billing End Date fields

You must update the **Billing Start Date** and **Billing End Date** fields on the **Billing Plan** object.

1. Go to **Setup** > **Create** > **Objects** > **Billing Plan** object.
2. Under **Custom Fields & Relationships** search for **Billing Start Date** field:
   a. Click **Edit** next to the button.
   b. Update the **Help Text** field description as follows:
      Determines the 'From' date when the billing should commence. If not populated by the user on the "Define Billing Plan" page, the engine will auto-populate it for the first line on the billing plan based on criteria. Users can attempt to overwrite the date. Also, it considers the no. of days in the "Payment Term" and helps to validate the "Ready for Invoice Date" populated by the user for the current installment.
   c. Click **Save**.
3. Under **Custom Fields & Relationships** search for **Billing End Date** field:
   a. Click **Edit** next to the button.
   b. Update the **Help Text** field description as follows:
      Determines the 'To' date when billing should end. If not populated by the user on the "Define Billing Plan" page, the engine will auto-populate this field for the

last line on the billing plan based on criteria. Users can attempt to overwrite the date. Also, it helps to validate if the "Ready for Invoice Date" for the next installment is clashing with the current installment or not.

c. Click **Save**.

Upgrading to CPQ February '24 Release

This section describes step-by-step instructions to upgrade from June '23 and October '23 to February '24 Release.

> ✅ The Conga Upgrade Program is an automated tool that upgrades packages available in your Salesforce org (Production or Sandbox) to the latest versions. In addition, It ensures all the Conga published managed packages are on the latest versions for the registered orgs. To register your org for upgrade program, see Registering for Conga Upgrade Program.

Upgrading CPQ from October '23 to February '24 Release

1. Go to **Setup** > **Installed Packages** and ensure that your current Salesforce org has the packages of CPQ October '23 release or later, which are the *minimum* required versions for upgrading to the February '24 release. Refer to the latest October '23 release or respective patch Release Notes.

2. Ensure that you have the following packages and dependent packages to upgrade to the February '24 release. These packages are required to utilize the new features and enhancements of the February '24 release. These are the *minimum* required versions; later versions are also supported.

Package	Latest Certified Version Name	Latest Certified Version Number
Conga Approvals **(New)** (Required if you are using Approvals)	15.0.388	15.388

Package	Latest Certified Version Name	Latest Certified Version Number
Conga Base Library **(New)** (Required if you are using Conga Configuration & Pricing)	5.0.255	5.255
Conga Billing **(New)** (Required if you are using Conga Billing)	10.0.396	10.396
Conga CLM Configuration Integration **(New)** (Required if you are using CLM)	16.0.0187	16.187
Conga Configuration & Pricing **(New)**	16.0.2122	16.2122
Conga Contract Lifecycle Management **(New)**	15.0.740	15.740
Conga CPQ Api **(New)**	16.0.0136	16.136
Conga CPQ (RLS) **(New)**	1.2.60	1.60
Conga CPQ Approvals (Required if you are using Approvals on CPQ objects)	13.2.25	13.25
Conga CPQ Maximizer **(New)**	10.0.0034	10.34
Conga CPQ Setup **(New)** (Required if you are using CPQ Admin Console)	16.0.145	16.145
Conga Deal Maximizer Setup (Required if you are using Deal Maximizer)	8.0.0053	8.53
Conga Deal Maximizer	7.0.0007	7.7

Package	Latest Certified Version Name	Latest Certified Version Number
Conga DocuSign Api	8.1.142.7	8.142.7
Conga Order Management	1.0.0000	1.0
Conga Promotions (Required if you are using Promotions)	1.0.0000	1.0
Conga Quote Approvals	7.1.10	7.10
Conga Quote Asset Integration	6.5.0015	6.15
Conga Quote CLM Integration	12.2.0085	12.85
Conga Quote Configuration Integration **(New)** (Required if you are using CPQ and Proposal Management)	16.0.0407	16.407
Conga Quote DocuSign Integration	4.1.23	4.23
Conga Quote Echosign Integration (You must install the latest Adobe Sign package to use this package)	3.1.17	3.17
Conga Quote Management (Required if you are using Proposal Management)	13.2.0267	13.267

3. Perform the upgrade. The upgrade procedure is the same as the installation procedure. For detailed information on installing CPQ managed packages, see Installing CPQ Packages.
4. After the upgrade is complete, perform the post-upgrade tasks. For more information, see Performing the Post-Upgrade Tasks.

Upgrading CPQ from June '23 to February '24 Release

1. Go to **Setup** > **Installed Packages** and ensure that your current Salesforce org has the packages of CPQ June '23 release or later, which are the *minimum* required versions for upgrading to February '24 release. Refer to the latest June '23 release or respective patch Release Notes.
2. Ensure that you have the following packages and dependent packages to upgrade to the February '24 release. These packages are required to utilize the new features and enhancements of the February '24 release. These are the *minimum* required versions; later versions are also supported.

Package	Latest Certified Version Name	Latest Certified Version Number
Conga Approvals **(New)** (Required if you are using Approvals)	15.0.388	15.388
Conga Base Library **(New)** (Required if you are using Conga Configuration & Pricing)	5.0.255	5.255
Conga Billing **(New)** (Required if you are using Conga Billing)	10.0.396	10.396
Conga CLM Configuration Integration **(New)** (Required if you are using CLM)	16.0.0187	16.187
Conga Configuration & Pricing **(New)**	16.0.2122	16.2122
Conga Contract Lifecycle Management **(New)**	15.0.740	15.740
Conga CPQ Api **(New)**	16.0.0136	16.136
Conga CPQ (RLS) **(New)**	1.2.60	1.60

Package	Latest Certified Version Name	Latest Certified Version Number
Conga CPQ Approvals (Required if you are using Approvals on CPQ objects)	13.2.25	13.25
Conga CPQ Maximizer **(New)**	10.0.0034	10.34
Conga CPQ Setup **(New)** (Required if you are using CPQ Admin Console)	16.0.145	16.145
Conga Deal Maximizer Setup (Required if you are using Deal Maximizer)	8.0.0053	8.53
Conga Deal Maximizer	7.0.0007	7.7
Conga DocuSign Api	8.1.142	8.142.8
Conga Order Management	1.0.0000	1.0
Conga Promotions (Required if you are using Promotions)	1.0.0000	1.0
Conga Quote Approvals	7.1.10	7.10
Conga Quote Asset Integration	6.5.0015	6.15
Conga Quote CLM Integration	12.2.0085	12.85
Conga Quote Configuration Integration **(New)** (Required if you are using CPQ and Proposal Management)	16.0.0407	16.407
Conga Quote DocuSign Integration	4.1.23	4.23

Package	Latest Certified Version Name	Latest Certified Version Number
Conga Quote Echosign Integration (You must install the latest Adobe Sign package to use this package)	3.1.17	3.17
Conga Quote Management (Required if you are using Proposal Management)	13.2.0267	13.267

3. Perform the upgrade. The upgrade procedure is the same as the installation procedure. For detailed information on installing CPQ managed packages, see Installing CPQ Packages.

4. After the upgrade is complete, perform the post-upgrade tasks. For more information, see Performing the Post-Upgrade Tasks.

Performing the Post-Upgrade Tasks

After you upgrade to this CPQ release, consider the following options and requirements:

Object Maintenance

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

To access the maintenance pages, click the **All Tabs** icon (➕) 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:

> ⚠ Do not run multiple maintenance jobs at a time. You must complete one after another in sequential order.

1. **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.
2. **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.

3. **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.

4. **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.

5. **Criteria Maintenance** - After any pricing change (rules, dimensions or any criteria change), you must run a Criteria Maintenance job in the following order.

    a. **Update Expression Fields**: To update all Expression Criteria fields.

    b. **Update Pricing Fields**: To update all Pricing Criteria fields.

    c. **Update Constraint Fields**: To update all Constraint Criteria fields.

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

6. **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.

7. **Custom Settings Maintenance:** This batch job generates static resources of custom settings for CPQ to use in computations for optimized processing. You must execute Custom Settings Maintenance after you make any changes in custom settings. Otherwise, changes are not reflected in CPQ. Earlier, you only had to save the custom settings for the changes to reflect on CPQ. Now, CPQ uses the static resources that are generated by the callback for computation rather than accessing the Custom Settings itself. For more information, see Running Maintenance Jobs.

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 field for this.

> ⓘ Do not append namespace before the field name in case you are entering the API names of product type fields in any of the Custom Field Name fields. The following table shows the correct and incorrect format of the API names.
>
> | Correct | Incorrect |
> | --- | --- |
> | Apttus_Config2_ProductIdr.ProductType_c | Apttus_Config2_ProductIdr.Apttus_Config2ProductType_c |
> | Apttus_Config2_OptionIdr.ProductType_c | Apttus_Config2_OptionIdr.Apttus_Config2ProductType_c |

Custom Labels for Custom Actions

After upgrading, it is recommended that you create custom labels for any custom actions. Note that you must do it for each flow or action combination, otherwise none of the buttons are displayed on the cart.

For more information, see Configuring Display Action Settings.

Removing the **APTS_SortDocumentsDesc** Admin Setting

If you are upgrading from the May '22 or September '22 to February '23 or later releases of CPQ, you must remove the **APTS_SortDocumentsDesc** admin setting. The admin setting is no longer supported. For more information, see Configuring Admin Settings.

Preparing for Upgrade (UDJS)

Before you upgrade to CPQ May '22 release, you must ensure the following:

> ✅ The Conga Push Upgrade is an automated tool that upgrades packages available in your Salesforce org (Production or Sandbox) to the latest versions. In addition, It ensures all the Conga published managed packages are on the latest versions for the registered orgs. To register your org for push upgrade, see Registering for Conga Push Upgrade.

* You go through CPQ Features by Release to know about the new features, enhancements, and deprecated features in CPQ since your existing release. After you upgrade CPQ to May '22 release, you cannot roll back to any previous release.
* You have the supported platforms and system requirements.
* You have access to the **Install Center** on the Conga Community Portal.
* You have administrator privileges to your Salesforce org.
* You need not back up your configurations. All configurations you performed since you installed your existing release will remain intact after the upgrade.
* You must delete the **UsageDataJobScheduler** job because it is no longer supported and may cause an error.

To delete the UsageDataJobScheduler job

The Apex Jobs page displays that the **UsageDataJobScheduler** job failed with an error. This happens because the **UsageDataJobScheduler** job is no longer supported. Before you upgrade **Conga Base Library** for this release, perform the following steps to avoid the error.

1. Go to **Setup** > **Administration Setup** > **Monitoring** > **Apex Jobs.**
2. Search for the **UsageDataJobScheduler** job, CPQ displays the status of the job as *Queued*.
3. Open another instance of the org in a different browser tab, and go to **Setup** > **Administration Setup** > **Monitoring** > **Scheduled Jobs**.
4. Search and delete the **UsageDataJobScheduler** job from Scheduled Jobs. On the Apex Jobs page, the status of the **UsageDataJobScheduler** job should now change to *Aborted*.
5. Install the latest package of **Conga Base Library**. For more information on the package version name and number, see Upgrading CPQ.

6. After the latest package is installed, check if the **UsageDataJobScheduler** job gets scheduled again on both the Apex Jobs and Scheduled Jobs pages. If not, the latest package has successfully disabled the scheduling of this job and has deprecated it.

Upgrade Assistant

Prerequisite

- Install the Conga Upgrade Assistant package from the Install Center.

Overview

Conga Upgrade Assistant is a upgrade management tool to help the administrators to upgrade Conga packages. It helps to upgrade the major releases, patch releases, and hotfixes.

It has three tabs:

1. Installed Packages
2. Instant Upgrade
3. Schedule Upgrade

Auto registration

1. Install the package (Conga Upgrade Assist).
2. Go to Conga upgrade assist app .
3. You will land on first page i.e. Installed Packages.
4. The process for auto registration starts automatically.
5. Click **Allow Access**.
6. The process starts.
7. Once the process is complete, you receive a message "The Org has been successfully registered with Conga Upgrade. Thank you."
8. The window closes automatically in few seconds or you can close it manually.
9. Message: Success! Org Registered Successfully

You will receive an email once the auto registration is complete.

Installed Packages

Installed packages lists all the packages that has been installed to your org.

Fields	Description
Name	The name of the package.
Installed Release Name	The name of the release installed in the org.
Installed Version	The current version number of the package that is installed in this org.
Latest GA Release Name	The latest name of the GA release. ⚠ If the Installed Version and the Latest GA Version are different, the this field will be highlighted with (✅) icon.
Latest GA Version	The latest version number of the package that are available for this org.

Auto Upgrade

Auto Upgrade feature is a provision in the installed packages tab that allows Conga to auto-upgrade the org when a new GA release is available. To enable auto upgrade feature:

1. Toggle the button to **enabled** in the Auto Upgrade section.
2. Click **Confirm** in the Auto Upgrade confirmation pop up.

The auto upgrade feature is turned on.

To disable this feature:

1. Toggle the button as **disabled** in the Auto Upgrade section.
2. Click **Confirm** in the Auto Upgrade confirmation pop up.

Package Access

Package access feature allows you to set the accessibility for Conga packages which shall be upgraded using Conga Upgrade Program.

1. Click Set Package Access button in the installed Products Page.
2. A Conga Package Access pop up displays. It has the following fields:

Fields	Description
Package Name	The name of the package displays.
Install For Admin Only	Toggle this on if you want to grant access only to the admins for all the packages.
Install for All Users	Toggle this on if you want to grant access to all the users for all the packages.
Install For Specific Profiles	Toggle this on if you want to select the profiles that you want to grant access to.

> ⚠ If you want to select a particular package for specific profile, you can choose the profile for the package.

3. After configuring access permissions, click **Save**.
4. Click **Confirm** in the Conga Package access confirmation pop up.

You get a success message.

Instant Upgrade

Instant Upgrade is a feature that allows you to start the upgrade instantly. The page consists the records of all the Upgrade Jobs performed till date.

The list of Upgrade Job Records has the following details:

Fields	Description
Job Id	The unique Id for all the schedule jobs performed in this org. ⓘ Click on the Job Id to see more details for the particular job.
Organization Name	The name of the organization will be displayed here.

Fields	Description
Job Start Date	The day and date when the the job was initiated.
Job Duration	Duration required to complete the process.
Release Name	The name of the release installed in the org.
Job Status	The status of the job whether it is success, or skipped.

This page has following features:

Upgrade to latest GA package

In this section, a button with the latest major release version is present. To upgrade the org to latest GA package:

1. Click the latest major release version button.
2. A pop up displays with the Name, Installed Version and the Target Version columns for all the packages. You can see if the package needs to be upgraded, if it is highlighted by the ( ) icon on the Target Version.
3. Click **OK**.
4. Click **Confirm** in the Release upgrade confirmation pop up.

Upgrade package

The Upgrade package section allows you to upgrade the packages from the latest patches, hotfixes, or major release. To upgrade the org to latest release package:

1. Select the Release version, you want to install packages for.
2. Click the **Upgrade Now** button.
3. A pop up displays with the Name, Installed Version and the Target Version columns for all the packages. You can see if the package needs to be upgraded, if it is highlighted by the ( ) icon on the Target Version.
4. Click **OK**.
5. Click **Confirm** in the Release upgrade confirmation pop up.

Schedule Upgrade

Schedule Upgrade lets you schedule any upgrade on a specified date and time. It displays the list of scheduled jobs. The details present on this tab are:

Fields	Description
Schedule Id	The unique schedule Id for all the schedule jobs performed in this org.
Schedule Name	The name of the scheduled job.
Release	The name of the release that is set to upgrade for the org.
Schedule Time	The date and the time scheduled for the org.
Job Status	The status of the job whether it is allocated, scheduled, cancelled, or completed.

To schedule the upgrade:

1. Go to Conga Upgrade Assist app > Schedule Upgrade tab.
2. Click the **Schedule Upgrade** button.
3. New schedule upgrade box pops up. Enter the following details.

Fields	Description
Schedule Name	Enter the name of the Schedule.
Date	Select the date that you want to schedule the upgrade on.
Select Release	Select the release name from the dropdown.
Time	Select the time on the date when you want to upgrade your org with the packages.

4. Click **Create.**
5. A pop up displays with the Name, Installed Version and the Target Version columns for all the packages. You can see if the package needs to be upgraded, if it is highlighted by the ( ) icon on the Target Version.

6. Click **OK**.
7. Click **Confirm** in the Release upgrade confirmation pop up.
8. The Job Status will be set to Scheduled.
9. You will receive an email once the upgrade is scheduled.

> ⚠ You cannot create another schedule upgrade until the previous upgrade is completed.

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 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.

Creating a Shopping Cart Experience

A CPQ administrator configures CPQ to create a shopping cart experience for the user.

Consider that ABC company sells products related to computers. Products including monitor, keyboard, mouse, Operating System, tools are sold by the ABC company. Each of these products can be sold as a stand-alone product or together as a bundle. For example, all of these products can be integrated as a bundle called computer where each of these products are options. Furthermore, each product has attributes associated with it. Attributes are features for a product. For example, attributes for a laptop are color, RAM, Operating System used and so on.You can configure attributes to derive pricing or enhance search options.

Each of these products has a price associated with it. Pricing can be different for each of these products sold as a stand-alone or when you buy a computer.

A sales representative can simply select a product when it is categorized rather than to search for the product from the huge array of products. A shopping cart experience for a user is enhanced when the user gets to search for his products based on some search criteria. The search criteria can be based on features of a products.

The sales of a product can be improved if user is advised to buy another product when he selects one product. For example, if a user buys a monitor, he will also be interested to buy a keyboard.

Consider that there is a holiday and the ABC company decides to promote a new product. User get discount of x% for all products from October to December.

All of this is configured by using CPQ.

Perform the following to configure CPQ:

1. Defining the products and services
2. Configure price for products
3. Configure rules that define pricing discounts and offers
4. Configure rules that guide user to buy products
5. Define search mechanism to search based on attributes
6. Run the maintenance jobs to update all changes made to products, bundle, category, and pricing.

Running Maintenance Jobs

As an administrator, you can run the following maintenance jobs.

Job	Definition
Custom Setting Maintenance	This job generates static resources for custom settings.
Category Maintenance	This job creates a de-normalized view for Categories with related products.
Criteria Maintenance • Update Pricing Fields • Update Expression Fields • Update Constraint Fields	These jobs optimize the query time by creating indexes for the fields that are referred to in the criteria or expressions.
Bundle Maintenance	This job synchronizes all the bundles and options.

You must execute an appropriate maintenance job after applying structural changes to the definition of categories, products, or criteria to ensure that the Sales rep does not see

stale definition in CPQ application. A stale definition may cause errors in the console or application.

Prerequisites

- 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**.
- Define **Update View Product Batch Size** and **Update View Category Batch Size** in Config System Properties.

To run a Custom Settings Maintenance job

CPQ uses custom settings to process computation on various CPQ pages. The Custom Settings Maintenance job generates static resources of the custom settings to reduce the time to fetch the values. Because the values are static, you must run the Custom Settings Maintenance Job every time you make any modification in the custom settings. Otherwise, the changes may not reflect in CPQ and the computations may get processed with stale values, resulting in incorrect configuration and pricing.

1. Go to All Tabs ( ) > **Custom Settings Maintenance**.
2. Select the flow you want to run the maintenance for or select *All* to run maintenance for all the flows.
3. Click **Update**.

To run a Category Maintenance job

1. On **App Menu**, click **Conga CPQ Admin** and then click the **CPQ Admin** tab. The new admin console is launched.
2. Click the more icon ( ) on the top-right corner and select **Category Maintenance**. The Category Maintenance window is displayed.
3. From the **Hierarchy** drop-down, select a particular hierarchy or *All* and click **Update View**. After the maintenance job is 100% complete, the **Status** column under **Batch Jobs** changes to Completed.
4. Close the Category Maintenance window.

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.

To use manual incremental update for category views

After making changes to the existing category definition you can run the Category Maintenance batch job only for changes rather than running for the entire definition. For example, if your category has 10 products and you added 5 more, you can run the Category Maintenance batch job only for the 5 additional products. To execute the Category Maintenance batch job for the changes, you must define an Admin Property that is used by CPQ to isolate the changes in the category.

Perform the following steps to add an admin entry:

1. Click the All tabs icon ( ) and click **Admin**.
2. Click New to create a new Admin entry.
3. In the **Name** field, enter *APTS_UpdateViewProductClasses*. Leave the fields **Value** and **Code** blank.
4. Click **Save**.

Now, if you change the category definition, the **Code** field in the Admin property is populated with the Product Classification records of the newly added products. These records are created after you associate products to the category. When you execute the Category Maintenance batch job, the maintenance batch job is only executed for the changes and not the entire definition. In case, you want to execute a maintenance batch job for the entire definition, clear the **Code** field in the Admin Entry and then execute the Category Maintenance batch Job.

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

To specify the batch size for the Category Maintenance batch job

You can use the *APTS_UpdateViewApiBatchSize* admin entry to specify the batch size for the Category Maintenance batch job when the job is executed using the exposed global Apex method *CPQWebService.updateHierarchyViews()*. The default batch size for the Category Maintenance batch job set in this global method is 2000, which is used when you have not defined the *APTS_UpdateViewApiBatchSize* admin entry. The behavior of the Category Maintenance batch job is same when the job is executed from the UI.

You must specify a batch size value only when you notice limit exceptions in batch jobs because of complex categories with a lot of hierarchy levels. In such cases, you must decrease the batch size to execute the batch jobs successfully.

> ⓘ The maximum batch size for the Category Maintenance batch job set in the *CPQWebService.updateHierarchyViews()* global method is 2000. You can specify any number between 1 to 2000 in the *APTS_UpdateViewApiBatchSize* admin entry depending on the number of categories in the org. If you specify a value greater than 2000, CPQ will use 2000 as default. The actual batch size value to be used depends on the complexity of the categories and hence is specific to your requirements. You must arrive at the exact number through the trial and error method.

Perform the following steps to add an admin entry:

1. Click the All tabs icon (➕) and click **Admin**.
2. Click **New** to create an admin entry.
3. In the **Name** field, enter *APTS_UpdateViewApiBatchSize*.
4. In the **Value** field, enter a value between 1 and 2000.
5. Leave the **Code** field blank.
6. Click **Save**.

To run a Criteria Maintenance job

1. On **App Menu**, click **Conga CPQ Admin** and then click the **CPQ Admin** tab. The new admin console is launched.
2. Click the more icon (⋮) on the top-right corner and select **Criteria Maintenance**. The Criteria Maintenance window is displayed.
3. Click one of the following buttons.
   - **Update Pricing Fields**: Use this command to update all Pricing Criteria fields.
   - **Update Expression Fields**: Use this command to update all Expression Criteria fields.
   - **Update Constraint Fields**: Use this command to update all Constraint Criteria fields.

   After the maintenance job is 100% complete, the **Status** column under **Batch Jobs** changes to Completed.
4. Close the Criteria Maintenance window.

To run a Bundle Maintenance Job

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

Bundle Products			Update in Fast Mode	Update in Batch Mode	Update All	Refresh	performance	Search
[0 Records Selected]								
☐	**Product Name**							
☐	740e xDrive iPerformance							
☐	DB Performance Cloud - Group							

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**.

   > ⓘ Run Bundle Maintenance in Batch Mode instead of Fast Mode if the bundle has a large number of options and sub-bundles. Bundle Maintenance in Fast mode can only be used for a limited number of options and sub-bundles. If that limit is exceeded an error is displayed.

   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 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.

To run Attribute Maintenance Job

In the Classic UI, you must run the Attribute Maintenance job after you update attributes. You must execute this job after you associate attribute groups to a category or define a Product Attribute Rule (PAR) or Attribute Value Matrix (AVM) with criteria. It is not required to execute the job if you use CPQ Admin UI; CPQ executes the batch job automatically.

Follow the steps below in Classic UI:

1. Click All Tabs icon ( ) and click **Attribute Maintenance**.
2. Click one of the following buttons:
   - **Update Attributes**: Use this command to update the attribute groups.
   - **Update Attribute Rule Views**: Use this command to update PAR and AVM.

About TurboEngines

Conga TurboEngines is a concurrent processing engine provided by Conga that comprises various microservices that process product configurations (**TurboConfig**), pricing calculations (**TurboPricing**), and other product-related business data, such as promotions. Conga TurboEngines offload the computation workload from the Salesforce platform to the **Conga Flexible Compute Platform** to reduce the processing time on the cart. Processing the computation workload in the Conga Flexible Compute Platform reduces the interaction costs and the quote turnaround time specifically during peak load or large transactions.

TurboEngines scale on the following dimensions:

- Number of users
- Size of transaction
- The complexity of the product and rules

TurboEngines also provide a critical component called **TurboEngines Data Sync** services that provide a high-performance mechanism to sync pricing and config master data at regular, scheduled intervals (or on-demand) between Salesforce and the Conga Flexible Compute Platform. Data is pushed to TurboPricing and TurboConfig consumer endpoints and made available for processing to take advantage of the performance improvements offered by the TurboEngines platform.
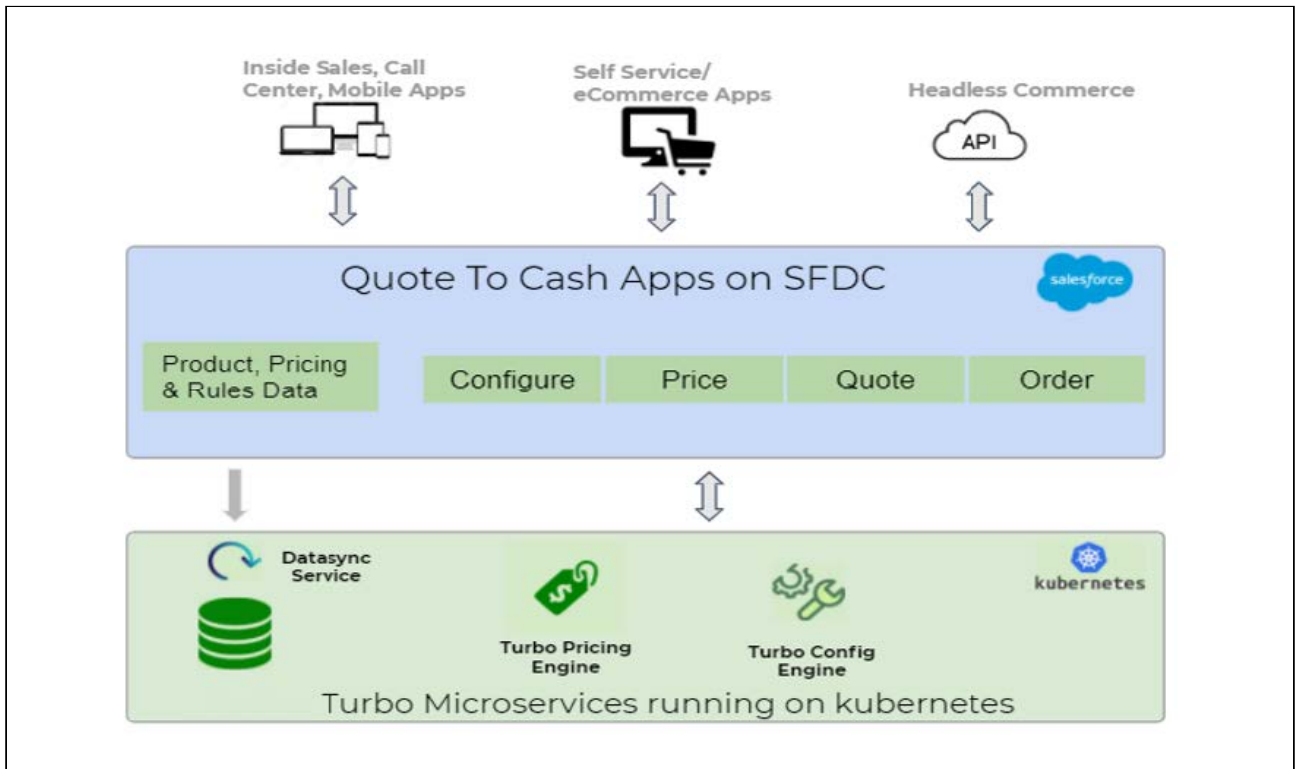
For more information on TurboEngines, refer to TurboEngines documentation.

In this section following topics are described:

- About TurboConfig
- About TurboPricing

About TurboConfig

TurboConfig is a configuration engine created to process product configuration rules when products and bundles are configured on a cart and when finalizing the quote. TurboConfig offloads the computation workload from the Salesforce platform to the Conga Flexible Compute Platform built using microservices to reduce the processing time of the configuration rules. Computation workload includes the processing of rules defined on the products. For example, in a TurboConfig enabled flow, when the Sales rep adds the product or the favorite configuration to the cart, the constraint rules associated with them are offloaded to the Conga Flexible Compute Platform to process. TurboConfig engine executes the rules, maintains rule states, and avoids unnecessary line item processing.

TurboConfig is recommended when you have a large number of rules or highly complex configuration rules to be applied while selecting a product or configuring a bundle.

About TurboPricing

TurboPricing is a pricing engine built using microservices to reduce the processing time on the cart. You can enable TurboPricing to offload complex pricing computation workload from the Salesforce platform to the Conga Flexible Compute Platform. It reduces time to submit prices to customers, improves user experience, and improves user adoption with a more responsive user interface.

The following diagram shows how data flows between a Salesforce org and Conga Flexible Compute Platform:
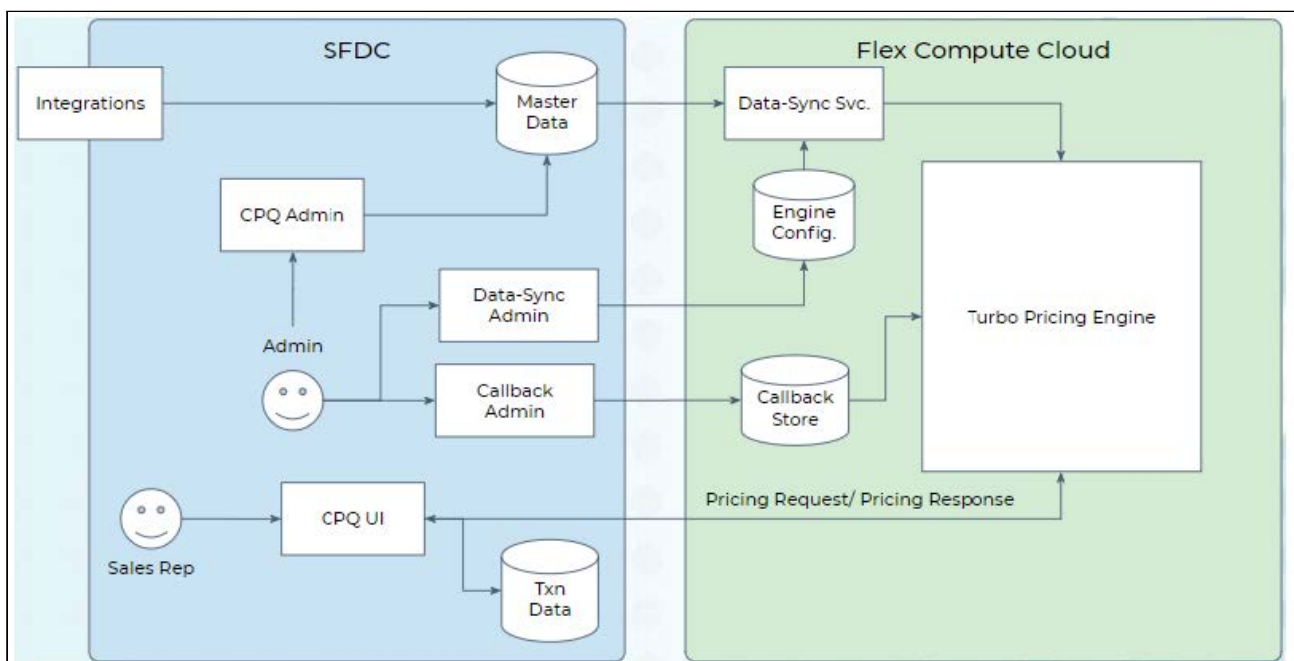
Using Shield Platform Encryption

You can add a layer of security to the sensitive data in CPQ using Salesforce Shield Platform Encryption. CPQ supports Salesforce Shield Platform Encryption with the following limitations:

- Encryption for number fields is not supported
- Encryption for account and contact is not supported

For more information, see How Shield Platform Encryption Works.

Enabling CPQ for Partner Community Users

This section describes how to enable CPQ for partner community users.

Prerequisites

- The *Partner Community User* profile must be available in your org at **Setup** > **Manage Users** > **Profiles**.
- The *Partner Community* user license must be available for *Partner Community User*.
- The **Enable Profile User Interface** setting must be enabled at **Setup** > **Manage Users** > **User Management Settings**.

To Enable CPQ for Partner Community users

1. Configure a profile.
2. Configure field-level security.
3. Configure permission set.
4. Share settings.
5. Configure partner community user.
6. Configure partner community site.
7. Configure a custom button on the quote.

Configuring a Profile

1. Go to **Setup** > **Manage Users** > **Profile**. The Profiles page is displayed.
2. Click the Partner Community User link. The Partner Community User page is displayed.
3. Click **Clone**. The Clone Profile page is displayed.
4. In the **Profile Name** field, enter a name for the profile. For example, Partner Community User Cloned.
5. Click **Save**. The Partner Community User Cloned profile is created. You must complete the following configuration.

Assigned Apps

1. Click **Assigned Apps**. The Assigned Apps page is displayed.

2. Click **Edit**, select the **Visible** checkbox for the following apps:
   - Conga Approvals Management
     (Apttus_Approval__ApttusApprovalsManagement)
   - Conga Approvals Setup (Apttus_Approval__ApttusApprovalsSetup)
   - Conga Batch Updates (Apttus_Config2__ApttusBatchUpdates)
   - Conga Billing Setup (Apttus_Config2__ApttusBillingSetup)
   - Apttus Contract Management (Apttus__ApttusContractManagement)
   - Apttus Contract Management (Lightning)
     (Apttus__ApttusContractManagement_Lightning)
   - Conga Configurable Wizard (Apttus__ApttusConfigurableWizard)
   - Conga CPQ Admin (Apttus_CPQAdmin__ApttusCPQAdmin)
   - Conga Incentive Setup (Apttus_Config2__ApttusIncentiveSetup)
   - Conga Lightning Component Admin (Apttus_Base2__Configurator)
   - Conga Order Management (Apttus_Config2__ApttusOrderManagement)
   - Conga Pricing Setup (Apttus_Config2__ApttusPricingSetup)
   - Conga Product Setup (Apttus_Config2__ApttusProductSetup)
   - Apttus Proposal Management (Apttus_Proposal__ApttusProposalManagement)
   - Apttus Proposal Management
     (Lightning) (Apttus_Proposal__ApttusProposalManagement_Lightning)
   - Conga X-Author™ For Excel (Apttus_XApps__XAuthorExcel)
   - TurboEngines Admin (Apttus_Base2__TurboEngineAdmin)
3. Click **Save**.

Object Settings

1. Click **Object Settings**. The Object Settings page is displayed.
2. Click the required object name link. The <object> page is displayed.
3. Click **Edit**. Configure the following settings:

Object Name	Tab Settings	Page Layouts	Object Permissions
Accounts	Default On	Varies by Record Type	Read, Create, Edit
Contacts	Default On	Varies by Record Type	Read, Create, Edit
DocGen Packages	Tab Hidden	Varies by Record Type	Read
Documents	Tab Hidden	--	Read
Opportunities	Default On	Varies by Record Type	Read, Create, Edit

Object Name	Tab Settings	Page Layouts	Object Permissions
Products	Default On	Product Layout (No Price)	Read
Proposals	Default On	Varies by Record Type	Read, Create, Edit

4. Click **Save**.

Apex Class Access

1. Click **Apex Class Access**. The Apex Class Access page is displayed.
2. Click **Edit**.
3. From the **Available Apex Classes** list, select CPQ-related classes.
4. Click **Add** to move them to the **Enabled Apex Classes** list.
5. Click **Save**.

Visualforce Page Access

1. Click **Visualforce Page Access**. The Visualforce Page Access page is displayed.
2. Click **Edit**.
3. From the **Available Visualforce Pages** list, select CPQ-related pages.
4. Click **Add** to move them to the **Enabled Visualforce Pages** list.
5. Click **Save**.

Custom Permissions

1. Click **Custom Permissions**. The Custom Permissions page is displayed.
2. Click **Edit**.
3. From the **Available Custom Permissions** list, select the following permissions:
    - Apttus.XA_CheckinAsFinal
    - Apttus.XA_CheckInWithoutRedlines
    - Apttus.XA_CheckInWithRedlines
    - Apttus.XA_Contracts
    - Apttus.XA_Template
    - Apttus.XA_UnlockReadOnlyClause
    - Apttus.XA_UnprotectDocument
4. Click **Add** to move them to the **Enabled Custom Permissions** list.

5. Click **Save**.

Custom Metadata Types

1. Click **Custom Metadata Types**. The Custom Metadata Types page is displayed.
2. Click **Edit**.
3. From the **Available Custom Metadata Types** list, select CPQ-related metadata types.
4. Click **Add** to move them to the **Enabled Custom Metadata Types** list.
5. Click **Save**.

Custom Setting Definitions

1. Click **Custom Setting Definitions**. The Custom Setting Definitions page is displayed.
2. Click **Edit**.
3. From the **Available Custom Setting Definitions** list, select the following and other CPQ-related definitions:
   - Apttus*.*
   - echosign_dev1.*
4. Click **Add** to move them to the **Enabled Custom Setting Definitions** list.
5. Click **Save**.

Configuring Field-Level Security

1. Go to **Setup** > **Create** > **Objects**. The Custom Objects page is displayed.
2. Click the Quote/Proposal object link. The Quote/Proposal (Managed) page is displayed.
3. In the **Custom Fields & Relationships** related list, click the Price List field link. The Price List (Managed) page is displayed.
4. Click **Set Field-Level Security**. The Set Field-Level Security page is displayed.
5. Select the **Visible** checkbox for the required profile and click **Save**. For example, Partner Community User Cloned.
6. Repeat step 3 to 5 for the following fields and other required fields for the Quote/Proposal object:
   - Approval Stage
   - Approval Status
   - Opportunity
   - Account
   - Proposal Name

7. Repeat step 2 to 6 for the CPQ-related custom objects.

> ⓘ If you add or remove any field or object, you must configure field-level security for those fields and objects.

Configuring a Permission Set

1. Go to **Setup** > **Manage Users** > **Permission Sets**. The Permission Sets page is displayed.
2. Click **New**. The Create page is displayed.
3. Enter the following details:
   a. **Label**: Enter a label for the permission set. For example, CPQ Sales User – Partner.
   b. **API Name**: The field is automatically populated.
   c. **Description**: Enter a description for the permission set.
   d. **License**: Select the Partner Community license for this permission set.
4. Click **Save**. The CPQ Sales User – Partner permission set is created. You must complete the following configuration.

Assigned Apps

1. Click **Assigned Apps**. The Assigned Apps page is displayed.
2. Click **Edit** and select the **Visible** checkbox for the following apps:
   - Apttus_Proposal.Apttus Proposal Management (Apttus_Proposal__ApttusProposalManagement)
   - Apttus_Proposal.Apttus Proposal Management (Lightning) (Apttus_Proposal__ApttusProposalManagement_Lightning)
3. Click **Save**.

Object Settings

1. Click **Object Settings**. The Object Settings page is displayed.
2. Click the required object name link. The <object> page is displayed.
3. Click **Edit**. Configure the following settings:

Object Name	Object Permissions	Tab Settings
Accounts	Read	--

Object Name	Object Permissions	Tab Settings
Adjustment Line Items	Read, Create, Edit, Delete	--
Admin	Read	--
Agreement Action Conditions	Read	--
Agreement Clauses	Read, Create	--
Agreement Documents	Read, Create	--
Agreement Explorer	Read, Create, Edit, Delete	--
Agreement Line Items	Read, Create, Edit, Delete	--
Agreement Locks	Read	--
Agreement Price Rules	Read, Create, Edit, Delete	--
Agreement Price Tiers	Read, Create, Edit, Delete	--
Agreement Protection	Read	--
Agreement Rule Conditions	Read	--
Agreement Rules	Read	--
Agreements	Read, Create	--
Agreements	Read	--
Agreement Term Exceptions	Read, Create, Edit, Delete	--
App Assignments	Read	--
App Files	Read	--
Application Features	Read	--
Applied Expression Infos	Read, Create, Edit, Delete	--

Object Name	Object Permissions	Tab Settings
Applied Rule Action Infos	Read, Create, Edit, Delete	--
Applied Rule Infos	Read, Create, Edit, Delete	--
Apps	Read	--
Asset Attribute Value History	Read, Create, Edit, Delete	--
Asset Attribute Values	Read, Create, Edit, Delete	--
Asset Line Item History	Read, Create, Edit, Delete	--
Asset Line Items	Read, Create, Edit, Delete	--
Asset Transaction History	Read, Create, Edit, Delete	--
Asset Usage Price Tier History	Read, Create, Edit, Delete	--
Asset Usage Price Tiers	Read, Create, Edit, Delete	--
Async Merge Calls	Read, Create, Edit, Delete	--
Attribute Group Translations	Read, Create, Edit, Delete	--
Attribute Value Matrices	Read	--
Attribute Value Matrix Entries	Read, Create	--
Bundle Components	Read	--
Categories	Read	--
Category Hierarchies	Read	--
CategoryTranslations	Read	--
Charge Group Members	Read	--

Object Name	Object Permissions	Tab Settings
Charge Groups	Read	--
Charge Types	Read	--
Collaboration Requests	Read, Create, Edit, Delete	--
Constraint Rule Action Expressions	Read	--
Constraint Rule Actions	Read	--
Constraint Rule Conditions	Read	--
Constraint Rules	Read	--
Content Events	Read, Create, Edit, Delete	--
Cost Adjustments	Read	--
Cost Items	Read	--
Cost Line Items	Read	--
Cost Models	Read	--
Cycle Time Field Data	Read, Create, Edit	--
Cycle Time Fields	Read, Create, Edit	--
Cycle Time Group Data	Read, Create, Edit	--
Cycle Time Groups	Read, Create, Edit	--
Doc Assembly Components	Read	--
Doc Assembly Rules	Read	--
Doc Assembly Rulesets	Read	--

Object Name	Object Permissions	Tab Settings
Document Agreement Clauses	Read	--
Document Collate Infos	Read	--
Document Version Details	Read, Create, Edit	--
Document Versions	Read, Create, Edit	--
External Order Adjustment Items	Read, Create, Edit, Delete	--
External Order Summary	Read, Create, Edit, Delete	--
External Order Summary Items	Read, Create, Edit, Delete	--
Favorite Configurations	Read, Create, Edit, Delete	--
Features	Read	--
Feature Sets	Read	--
Field Expressions	Read	--
Footnote	Read	--
Formula Fields	Read, Edit	--
Formula Fields (Comply)	Read	--
Frequency/UOM Conversion Rates	Read, Create	--
Guided Questions	Read	--
Guided Search Rule Entries	Read	--
Guided Search Rule Filters	Read	--

Object Name	Object Permissions	Tab Settings
Guided Selling Rules	Read	--
Help Docs	Read	--
Line Item Rollups	Read, Create, Edit, Delete	--
Line Items	Read, Create, Edit, Delete	--
Merge Event Details	Read, Create, Edit, Delete	--
Merge Events	Read, Create, Edit, Delete	--
Opportunities	Read, Create, Edit	--
Payment Terms	Read	--
Permission Set Relationships	Read	--
Price Breakups	Read	--
Price Dimensions	Read	--
Price Escalators	Read	--
Price List Categories	Read	--
Price List Items	Read, Create, Edit	--
Price Lists	Read, Create, Edit	--
Price Matrices	Read	--
Price Matrix Entries	Read	--
Price Rule Entries	Read	--
Price Rules	Read	--
Price Rulesets	Read	--

Object Name	Object Permissions	Tab Settings
Product Attribute Group Members	Read, Create, Edit	--
Product Attribute Groups	Read, Create, Edit	--
Product Attribute Matrix Views	Read, Create, Edit	--
Product Attribute Rule Actions	Read, Create, Edit	--
Product Attribute Rules	Read, Create, Edit	--
Product Attribute Rule Views	Read, Create, Edit, Delete	--
Product Attributes	Read, Create, Edit	--
Product Attribute Values	Read, Create, Edit	--
Product Classifications	Read, Create, Edit	--
Product Configurations	Read, Create, Edit, Delete	--
Product Constraint Entries	Read, Create, Edit, Delete	--
Product Constraints--	Read, Create, Edit, Delete	--
Product Constraint Views	Read, Create, Edit, Delete	--
Product Default Rule Filters	Read, Create, Edit, Delete	--
Product Default Rules	Read	--
Product Default Values	Read, Create, Edit, Delete	--
Product Features	Read, Create, Edit, Delete	--
Product Feature Values	Read, Create, Edit, Delete	--

Object Name	Object Permissions	Tab Settings
Product Filter Views	Read, Create, Edit, Delete	--
Product Footnotes--	Read, Create, Edit	--
Product Group Members	Read, Create, Edit	--
Product Groups	Read, Create, Edit, Delete	--
Product Hierarchy Views	Read, Create, Edit, Delete	--
Product Information	Read, Create, Edit	--
Product Option Components	Read, Create, Edit	--
Product Option Groups	Read, Create, Edit	--
Product Option Prices	Read, Create, Edit	--
Products	Read	--
Product Translations--	Read, Create, Edit, Delete	--
Proposal Adjustment Line Items	Read, Create, Edit, Delete	--
Proposal Document Output Formats	Read	--
Proposal Footnotes	Read, Create, Edit, Delete	--
Proposal Incentive Benefit Data	Read, Create, Edit, Delete	--
Proposal Incentive Limit Data	Read, Create, Edit, Delete	--
Proposal Line Items	Read, Create, Edit, Delete	--
Proposal Location	Read, Create, Edit, Delete	--
Proposal Product Attribute Values	Read, Create, Edit, Delete	--

Object Name	Object Permissions	Tab Settings
Proposal Related Line Items	Read, Create, Edit, Delete	--
Proposal Requests	Read, Create, Edit, Delete	--
Proposal Rollup Data	Read, Create, Edit, Delete	--
Proposals	Read, Create, Edit	Visible
Proposal Summary	Read, Create, Edit, Delete	--
Proposal Summary Groups	Read, Create, Edit, Delete	--
Proposal Tax Breakups	Read, Create, Edit, Delete	--
Proposal Usage Price Tiers	Read, Create, Edit, Delete	--
Published Favorites	Read, Create, Edit, Delete	--
Query Template Filters	Read	--
Query Template Qualifiers	Read	--
Query Templates	Read	--
Related Account Locations	Read	--
Related Agreements	Read, Create, Edit	--
Related Asset Line Items	Read, Create, Edit, Delete	--
Related Incentives	Read, Create, Edit, Delete	--
Related Line Items	Read, Create, Edit, Delete	--
Related Opportunity Line Item Colls	Read, Create, Edit, Delete	--

Object Name	Object Permissions	Tab Settings
Related Opportunity Line Items	Read, Create, Edit, Delete	--
Related Price List Items	Read, Create, Edit, Delete	--
Related Products	Read, Create, Edit, Delete	--
Retention Policies	Read, Create, Edit, Delete	--
Revenue Recognition Policies--	Read, Create, Edit, Delete	--
Revenue Split/Merge Policies	Read, Create, Edit, Delete	--
Revenue Split Policy Entries	Read, Create, Edit, Delete	--
Rollup Data	Read, Create, Edit, Delete	--
Saved Searches	Read, Create, Edit, Delete	--
Search Attribute Values	Read	--
Search Filters (Comply)	Read	--
Search Filters (CPQ)	Read	--
Service Locations--	Read, Create, Edit, Delete	--
Setting Groups	Read, Create, Edit, Delete	--
Smart Search Activity History	Read, Create, Edit, Delete	--
Summary Groups	Read, Create, Edit, Delete	--
Temp Display Columns	Read, Create, Edit, Delete	--
Temp Filters	Read, Create, Edit, Delete	--
Temp Incentive Benefit Data	Read, Create, Edit, Delete	--
Temp Incentive Limit Data	Read, Create, Edit, Delete	--

Object Name	Object Permissions	Tab Settings
Template Clause References	Read, Create, Edit, Delete	--
Template Clause Reference Versions	Read, Create, Edit, Delete	--
Template Datasource Filters	Read, Create, Edit, Delete	--
Template Dynamic Sections	Read, Create, Edit, Delete	--
Template Dynamic Section Versions	Read, Create, Edit, Delete	--
Templates	Read	--
Template Versions	Read, Create, Edit, Delete	--
Temp Objects	Read, Edit	--
Temp Objects (CPQ)	Read, Create, Edit, Delete	--
Temp Relate Data	Read, Create, Edit, Delete	--
Temp Renew Asset Groups	Read, Create, Edit, Delete	--
Temp Renew Asset Line Items	Read, Create, Edit, Delete	--
Temp Renews	Read, Create, Edit, Delete	--
Temp Rollup Data	Read, Create, Edit, Delete	--
Term Exceptions	Read	--
Totaling Groups	Read, Create, Edit, Delete	--
Transfer Prices	Read, Create, Edit, Delete	--
Usage Price Tiers	Read, Create, Edit, Delete	--

Object Name	Object Permissions	Tab Settings
User Views	Read	--

4. Click **Save**.

Apex Class Access

1. Click **Apex Class Access**. The Apex Class Access page is displayed.
2. Click **Edit**.
3. From the **Available Apex Classes** list, select CPQ-related classes.
4. Click **Add** to move them to the **Enabled Apex Classes** list.
5. Click **Save**.

Visualforce Page Access

1. Click **Visualforce Page Access**. The Visualforce Page Access page is displayed.
2. Click **Edit**.
3. From the **Available Visualforce Pages** list, select CPQ-related pages.
4. Click **Add** to move them to the **Enabled Visualforce Pages** list.
5. Click **Save**.

Sharing Settings

1. Go to **Setup** > **Security Controls** > **Sharing Settings**. The Sharing Settings page is displayed.
2. Click **Edit**. The Organization-Wide Sharing Defaults Edit page is displayed.
3. From the **Default Internal Access** drop-down, select a required value.
4. From the **Default External Access** drop-down, select a required value.
5. Select the **Grant Access Using Hierarchies** checkbox for the required object.
6. Click **Save**.

Object	Default Internal Access	Default External Access	Grant Access Using Hierarchies
Product	Public Read/Write	Public Read Only	Checked
Location	Public Read/Write	Public Read/Write	Checked

Object	Default Internal Access	Default External Access	Grant Access Using Hierarchies
Address	Public Read Only	Public Read Only	Checked
Adhoc Approval Process	Public Read/Write	Public Read Only	Checked
Agreement Document Output Format	Public Read/Write	Public Read Only	Checked
Agreement Price Rule	Public Read/Write	Public Read Only	Checked
Agreement Protection	Public Read/Write	Public Read Only	Checked
Agreement Rule	Public Read/Write	Public Read Only	Checked
Agreement Template	Public Read/Write	Public Read Only	Checked
Approval Job	Public Read/Write	Public Read Only	Checked
Approval Matrix	Public Read/Write	Public Read Only	Checked
Approval Process	Public Read/Write	Public Read Only	Checked
Approval Request	Public Read/Write	Public Read Only	Checked
Approval Request History	Public Read/Write	Public Read Only	Checked
Approval Rule	Public Read/Write	Public Read Only	Checked
Approval Rule Dimension	Public Read/Write	Public Read Only	Checked
Asset Line Item	Public Read/Write	Public Read Only	Checked
Asset Line Item History	Public Read/Write	Public Read Only	Checked

Object	Default Internal Access	Default External Access	Grant Access Using Hierarchies
Asset Transaction History	Public Read/Write	Public Read Only	Checked
Attribute Value Matrix	Public Read/Write	Public Read Only	Checked
Bundle Component View	Public Read/Write	Public Read/Write	Checked
Bundle Pricing Mapping	Public Read/Write	Public Read/Write	Checked
Business Category	Public Read Only	Public Read Only	Checked
Business Unit	Public Read/Write	Public Read/Write	Checked
Category	Public Read/Write	Public Read Only	Checked
CategoryTranslation	Public Read/Write	Public Read Only	Checked
Collaboration Request	Public Read/Write	Public Read Only	Checked
Config Settings	Public Read/Write	Public Read Only	Checked
Configuration	Public Read/Write	Public Read/Write	Checked
Constraint Rule	Public Read/Write	Public Read Only	Checked
Custom Data Mapping	Public Read/Write	Public Read/Write	Checked
Customer Success Program	Public Read/Write	Public Read/Write	Checked
Discount Approval Matrix	Public Read/Write	Public Read Only	Checked
DocGen Package	Public Read/Write	Public Read Only	Checked

Object	Default Internal Access	Default External Access	Grant Access Using Hierarchies
Favorite Configuration	Public Read/Write	Public Read Only	Checked
Features	Public Read/Write	Public Read Only	Checked
Feature Set	Public Read/Write	Public Read Only	Checked
Feature Setting	Public Read/Write	Public Read Only	Checked
Field Expression	Public Read/Write	Public Read Only	Checked
Formula Field (Approvals)	Public Read/Write	Public Read Only	Checked
Formula Field (Comply)	Public Read/Write	Public Read Only	Checked
Formula Field (CPQ)	Public Read/Write	Public Read Only	Checked
Group Mapping	Public Read/Write	Public Read/Write	Checked
Guided Question	Public Read/Write	Public Read Only	Checked
Guided Selling Rule	Public Read/Write	Public Read Only	Checked
Legal Entity	Public Read/Write	Public Read Only	Checked
Line Item Rollup	Public Read/Write	Public Read Only	Checked
Payment Term	Public Read/Write	Public Read Only	Checked
Price Dimension	Public Read/Write	Public Read Only	Checked
Price List	Public Read/Write	Public Read Only	Checked
Price Ruleset	Public Read/Write	Public Read Only	Checked

Object	Default Internal Access	Default External Access	Grant Access Using Hierarchies
Product Attribute Group	Public Read/Write	Public Read Only	Checked
Product Attribute Matrix View	Public Read/Write	Public Read Only	Checked
Product Attribute Rule	Public Read/Write	Public Read Only	Checked
Product Attribute Rule View	Public Read/Write	Public Read Only	Checked
Product Configuration	Public Read/Write	Public Read/Write	Checked
Product Constraint	Public Read/Write	Public Read Only	Checked
Product Constraint View	Public Read/Write	Public Read Only	Checked
Product Option Component	Public Read/Write	Public Read Only	Checked
Search Filter (Approvals)	Public Read/Write	Public Read Only	Checked
Search Filter (Comply)	Public Read/Write	Public Read Only	Checked
Search Filter (CPQ)	Public Read/Write	Public Read Only	Checked
Template	Public Read/Write	Public Read Only	Checked
Template Locale	Public Read/Write	Public Read Only	Checked
Temp Object (Comply)	Public Read/Write	Public Read/Write	Checked
Temp Object (CPQ)	Public Read/Write	Public Read/Write	Checked
Translation	Public Read/Write	Public Read Only	Checked

Configuring a Partner Community User

1. Click the **All Tabs** icon (+). The All Tabs page is displayed.
2. Search and click Accounts. The Accounts page is displayed.
3. Click **New**. The New Account page is displayed.
4. In the **Account Name** filed, enter a name for the account and click **Save**. For example, AutoQuotePartnerAccount.
5. On the **Manage External Account** button, click **Enable As Partner**. A confirmation pop-up is displayed.
6. Click **OK**. The AutoQuotePartnerAccount partner account is created. You must complete the following configuration.

To create a user

1. Go to the Contacts related list on the account (for example, AutoQuotePartnerAccount) and click **New Contact**.
2. From the **Salutation** drop-down, select a salutation for the contact. For example, Mr.
3. In the **First Name** field, enter the first name of the contact. For example, AutoQuote.
4. In the **Last Name** field, enter the last name of the contact. For example, PartnerContact.
5. In the **Email** field, enter the email ID of the contact.
6. Click **Save**.
7. On the **Manage External User** button, click **Enable Partner User**. The New User page is displayed where the First Name, Last Name, and Alias fields are auto-populated.
8. From the **User License** drop-down, select Partner Community.
9. From the **Profile** drop-down, select a profile. For example, Partner Community User Cloned.
10. In the **Email** field, enter the email ID.
11. In the **Username** field, enter partner@conga.com.xxxxxxx (here xxxxxxx is your sandbox name).
12. Select the **Active** checkbox.
13. Select the **Salesforce CRM Content User** checkbox.
14. Click **Save**. A confirmation pop-up is displayed.
15. Click **OK**.

To assign permission set to the user

1. Go to the Permission Set Assignments related list on the user (for example, AutoQuote PartnerContact) and click **Edit Assignments**.
2. From the **Available Permission Sets** list, select the required permission set. For example, CPQ Sales User – Partner.
3. Click **Add** to move it to the **Enabled Permission Sets** list.
4. Click **Save**.

Configuring a Partner Community Site

1. Go to **Setup** > **Customize** > **Digital Experiences** > **All Sites**. The Digital Experiences page is displayed.
2. Click **New**. The Choose the Experience You Love page is displayed.
3. On the **All** tab, select a required theme. The Customer Service page is displayed.
4. Click **Get Started**. The Enter a Name page is displayed.
5. In the **Name** field, enter a name for your community.
6. In the **URL** field, enter a site prefix.
7. Click **Create**. It may take time to load the new community page.
8. Click **Administration**. The Administration page is displayed.
9. Click **Settings**. The Setting section is displayed.
10. Click **Activate**. A confirmation pop-up is displayed.
11. Click **OK**. You must complete the following configuration for the site.

Members

1. On the Administration page, click **Members** on the left panel. The Members section is displayed.
2. Under Select Profiles, in the **Search** drop-down, select All.
3. From the **Available Profiles** list, select the required profile. For example, Partner Community User Cloned.
4. Click **Add** to move it to the **Selected Profiles** list.
5. Click **Save**.

Menu Items

1. Click **Administration** on the top-left and click **Builder**.

2. Click the ⊞ icon on the toolbar that is displaying the HOME tab. The Navigation Menu is displayed on the right side.

3. Click **Edit Default Navigation**. The Edit Default Navigation pop-up is displayed.

4. Click **Add Menu Item**. The Menu Items section is displayed on the pop-up.

5. Enter the following details:

    a. In the **Name** field, enter Quotes.

    b. From the **Type** drop-down, select Salesforce Object.

    c. From the **Object Type** drop-down, select Quote/Proposal.

    d. From the **Default List View** drop-down, select Default.

    e. Select the **Publicly available** checkbox.

6. Click **Add Menu Item**. The Menu Items section is displayed on the pop-up.

7. Enter the following details:

    a. In the **Name** field, enter Agreements.

    b. From the **Type** drop-down, select Salesforce Object.

    c. From the **Object Type** drop-down, select Agreement.

    d. From the **Default List View** drop-down, select Default.

    e. Select the **Publicly available** checkbox.

8. Click **Save Menu**. The menus you added are displayed on the toolbar that is displaying the HOME tab.

9. Click **Publish** from top right. The Publish your site? pop-up is displayed.

10. Click **Publish** and then click **Got It**.

Configuring a Custom Button on the Quote Object for Partners

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**, enter the following details and click Next:

    a. In the **Field Label** field, enter Configure Products Partners.

    b. In the **Field Name** field, enter Configure_Products_Partners.

    c. 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_Config2__Cart businessObjectId="&Id&"&flow=PurchaseFlow", IMAGE("/
```

```
resource/Apttus_QPConfig__Button_Configure", "Configure Products Partners"),"_s
elf"), NULL)
```

6. In **Step 4. Establish field-level security**, select Field Level Security for the required profile and click **Next**. For example, Partner Community User Cloned.
7. In **Step 5. Add to page layouts**, choose the page layouts on which you want to display this field.
8. Click **Save**.
9. Provide access of this field to the required profile. For example, Partner Community User Cloned.

Validating the Partner Community User

1. Go to the contact that you created (for example, AutoQuote PartnerContact).
2. Click **Login to Experience as User**.
   - If you have multiple communities, CPQ prompts you to select a community.
   - If you have only one community, CPQ navigates you to the community home page.
3. On the Quotes tab, click **New**.
4. Select a record type as Proposal and click **Next**.
5. Verify if all required fields are visible. If not, you must verify Field-Level Security configuration.
6. Select the quote you created and assign a price list.
7. Click **Configure Products Partners**. The cart configuration page is displayed.
8. Select the required products and finalize the cart.

> ⓘ If CPQ did not launch the cart, verify sharing settings or field-level access again. You must enable subscriber logs to get more detail.

Setting up Salesforce Org with Conga Cloud Instance for CPQ

To set up Salesforce CPQ with the Conga platform, you must configure both the Salesforce org and the Conga platform.

An administrator can be any of the following persona: Customer Administrator, Partner Administrator, or any other administrator assigned the responsibility of setting up Salesforce org with the Conga platform. In this topic, this persona is referred to as the Customer Administrator.

The Customer Administrator is responsible for completing the entire setup process with assistance from the Conga CloudOps team.

Order	Task	Owner
1	Tenant Onboarding	This task is owned by the customer Admin and Cloud Ops team. • Customer Admin: Reaches out to the Conga service team to log a service ticket for tenant onboarding. • CloudOps: Initiates tenant onboarding process based on the service ticket.
2	Configuring Salesforce org	Customer Admin
3	Configuring Conga Platform	Customer Admin
4	Configuring Revenue Admin Settings	Customer Admin

- Tenant Onboarding
- Configuring the Salesforce Org
- Configuring the Conga Platform
- Configuring Revenue Admin Settings
- Creating a Quote from the Proposal

Tenant Onboarding

This topic outlines all the necessary steps for tenant onboarding. Both the Conga CloudOps team and customer admin are involved in this process.

Prerequisites

You have provided the following information to your Conga support representative:

- Your 18-digit Salesforce org ID.
- The administrator's first name, last name, and email ID.
- Your Salesforce org domain name. To view the domain details of your Salesforce org, go to Setup > enter `my domain` in the Quick Find box and click **My Domain**.

To onboard a tenant

Generally, your Conga support representative will log a tenant onboarding request with the following details:

1. Salesforce org and admin details:
   Add the Salesforce org ID and the Admin User's first name, last name, and email ID.
2. Redirect URLs:
   Request to add the following Salesforce URLs to the list of redirect URLs. Ensure the `< CustomerOrgDomainName >` name is replaced with the actual org domain name.

```
"https://<CustomerOrgDomainName>.my.salesforce.com/",
"https://<CustomerOrgDomainName>--conga-ccpq.vf.force.com/",
"https://<CustomerOrgDomainName>.lightning.force.com/",
"https://<CustomerOrgDomainName>--conga-ccpq.vf.force.com/apex/RLPCart"
```

On successful onboarding, the admin user receives a confirmation email containing Conga Org details along with Conga IDP login credentials with a one-time password.

You will also receive the following details from your Conga support representative, which will help you to set up your Salesforce org with the Conga Platform.

1. SPAClientId

   ⓘ If you receive a JSON payload, you will find the clientId within it. This is your SPAClientID. Use this SPAClientId when configuring the admin setting.

2. Environment ClientID and Client Secret. The ClientId and Client Secret are specific to the environment. Use these details when configuring a new Auth. Provider.

To change the password

1. Click the Login URL (Go to URL) in the welcome email.
2. Enter the username and password (Password for first-time login) in the welcome email.
3. Click **Sign-in** and follow the on-screen instructions to change the password.

Once the password is changed, you must log in again with the new password and capture your Conga organization ID, organization display name, and user ID, which you will later use to formulate an org ID when configuring the admin setting.

To capture Conga organization display name and organization ID

1. Click on the Login URL (Go to URL) in the welcome email.
2. Enter the Username and Password (new password).
3. Click **Sign-in**. The Launch your Conga Experience page is displayed.
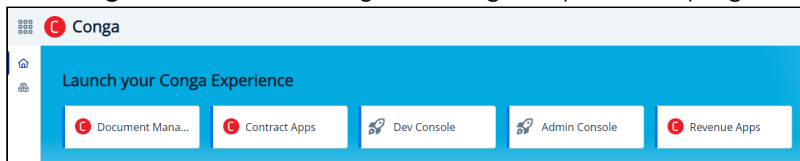
4. Click **Admin Console**. The Platform Administration page is displayed.
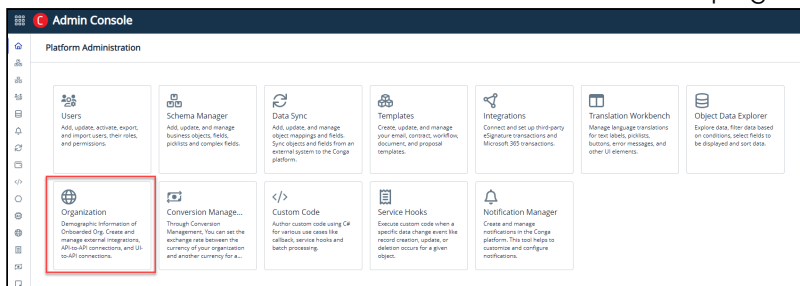
5. Click **Organization**. The Organization page is displayed. By default, the Conga Org tab is selected.
6. Save the following details:
   a. Organization Display Name
   b. Organization ID

To capture user ID

1. Click on the Login URL (Go to URL) in the welcome email.
2. Enter the Username and Password (new password) in the welcome email.
3. Click **Sign-in**. The Launch your Conga Experience Page is displayed.
4. Click **Admin Console**. The Platform Administration page is displayed.
5. Click **Users**. A Users page is displayed.
6. Click the desired username. A User Details page is displayed.
7. Save the value from the id field.

The onboarding process is complete. You must have the following to continue with the post-onboarding settings.

- SPAClientID
- Environment ClientID and Client Secret
- Conga Organization Display Name
- Conga Organization ID
- User Id

Save this information for future use.

Next Step: Configurations on Salesforce Org

Configuring the Salesforce Org

After completing tenant onboarding, use the configurations in this section to access new CPQ APIs on the Conga Platform using the Salesforce UI and synchronize data automatically from Salesforce to the Conga Platform. Review the following topics for more information.

> (i) Start with the Installing Conga CPQ (RLS) Package topic and follow the steps for the successful org setup.

- Installing Conga CPQ (RLS) Package
- Creating a New Admin Setting
- Creating a new Auth. Provider
- Creating Remote Site Settings
- Creating a New Static Resource
- Creating the Configure Products Button for RLS Flow
- Enabling OAuth Settings
- Creating a Connected App

Installing Conga CPQ (RLS) Package

Installing the Conga CPQ (RLS) package is required to integrate and synchronize your CPQ data with the Conga Platform.

Prerequisites

- Conga Community portal credentials
- Installation key

Reach out to the Conga support team for the installation key.

To Install Conga CPQ (RLS) package

Install the latest version of the Conga CPQ (RLS) package. Refer to the latest CPQ on Conga Platform release notes for the packages.

1. Login to the Conga Community portal.
2. Go to **Get Started** > **Install Center**.
3. In the My Packages navigation link, click **CPQ**. A list of CPQ-related packages is displayed.
4. On the CPQ (RLS) tile, click 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.
   To install any specific version, click **Details** on the package tile, and select the required version > click **Install in Production** to install the packages in your production environment. Click **Install in Sandbox** to install the packages in your sandbox. A Salesforce login screen is displayed.
5. Enter login credentials and click **Log In**. Use the onboarded Salesforce org login credentials.
6. Enter the installation key provided by Conga.
7. Select the profile for which you want to install the package. Conga recommends that you select **Install for All Users**.
   - **Install for Admins Only:** This is the default setting. Apply this access level to disable all object permissions.
   - **Install for All Users:** 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.

- **Install for Specific Profiles:** Apply this access level to assign permissions to specific users to read, create, edit, delete, view all, and modify all, for the pricing-related objects in the CPQ package.

8. Click **Install**.

After installing this package, ensure that the onboarded Salesforce org has the following Conga packages with the version numbers specified below. If not, you must upgrade them.

- Conga Quote Configuration Integration: **15.399** or later.
- Conga Quote Management: **13.262** or later.
- Conga Configuration & Pricing: **15.2062** or later.

To view the installed packages on your Salesforce org.

1. Login to your Salesforce org.
2. Go to **Setup >** in the Quick Find box, enter `Installed Packages` and select it. A list of installed packages is displayed.
3. Verify the version number against the packages.

Next Step: Creating a New Admin Setting.

Creating a New Admin Setting

Admin Settings are system properties that have different values, and you can use them for different purposes. You must configure admin settings to meet the business objectives of your implementation along with Custom Settings.

Prerequisites

Field	Description
SPAClientId	A single-page application ID. Enter the SPAClientId that you received from the Conga CloudOps team after onboarding your tenant.
OrganizationId	Enter Conga Organization ID that you captured after tenant onboarding. To view the organization ID, see the "To capture Conga Organization Display Name and Organization ID" section in Tenant Onboarding.
OrganizationFId	Your Conga organization's friendly ID. Friendly ID is formatted as <Conga org display name-org Id>. For more information, see the "To capture Conga Organization Display Name and Organization ID" section in Tenant Onboarding.

Field	Description
UserId	Enter the user ID that is created on the Conga Platform. To view the user ID on the Conga Platform, log in to Conga Platform as administrator, then go to Admin Console > Users > Click on the desired username.
InstanceURL	Conga organization instance URL. Use any of the following URLs per your Conga instance: • For production instance: https://prod-rls10.congacloud.com • For Preview instance: https://preview-rls09.congacloud.com • For staging instance: https://rls-stg.congacloud.io • For qa instance: https://rls-qa.congacloud.io • For dev instance: https://rls-dev.congacloud.io
UIInstanceURL	Conga Organization UI instance URL. Use any of the following URLs per your Conga instance: • For production instance: https://prod-rls10.congacloud.com/ecpqui • For preview instance: https://preview-rls09.congacloud.com/ecpqui • For staging instance: https://rls-stg.congacloud.io/ecpqui • For qa instance: https://rls-qa.congacloud.io/ecpqui • For dev instance: https://rls-dev.congacloud.io/ecpqui
UpsertQuotesEndpointURL	The upsert quotes URL. • For production instance: https://prod-rls10.congacloud.com/api/quote/v1/quotes/upsert-quote • For preview instance: https://preview-rls09.congacloud.com/api/quote/v1/quotes/upsert-quote • For staging instance: https://rls-stg.congacloud.io/api/quote/v1/quotes/upsert-quote • For qa instance: https://rls-qa.congacloud.io/api/quote/v1/quotes/upsert-quote • For dev instance: https://rls-dev.congacloud.io/api/quote/v1/quotes/upsert-quote
LoginURL	Your Conga Platform login URL. • For Production instance: https://login-rls.congacloud.com/ • For preview instance: https://login-rlspreview.congacloud.com/ • For staging instance: https://login-rlsstg.congacloud.io • For qa instance: https://login-rlsqa.congacloud.io • For dev instance: https://login-rlsdev.congacloud.io

Field	Description
Enter the following endpoints only when you use X-Author for Excel, Conga Sign, and Composer applications along with CPQ. These endpoints are specific to the connected environment. Reach out to the Conga CloudOps team for the endpoints.	
XAEndpoint	For production: https://xauthor-prod-rls10.congacloud.com For preview environment: For staging environment: https://xauthor-stg.congacloud.io
SignEndpoint	https://coreapps-rlspreview.congacloud.com
ComposerEndpoint	https://coreapps-rlspreview.congacloud.com

To configure the admin entry

1. Click the **App Launcher** icon (⦂⦂⦂), enter Admin, and select Admin. A list of admin settings is displayed.
2. Click **New**.
3. Enter the following details:

Field	Value
Name	APTS_RLPInstanceSettings
Value	Leave this field blank.

Field	Value
Code	Sample value:

> ⓘ The values in the below code are specific to the connected environment. In the below example, the values are based on the preview environment.

```
{
"SPAClientId": "technicalpublicationteam-spa",
"OrganizationId": "c7552ce4-a380-45b0-9c8a-022d2eafa7df",
"OrganizationFId": "auert00000004wfoai-c7552ce4-
a380-45b0-9c8a-022d2eafa7df",
"UserId": "769d5e44-ef45-cb0d-dd3c-1f046feded1c",
"InstanceURL": "https://preview-rls09.congacloud.com",
"UIInstanceURL": "https://preview-rls09.congacloud.com/ecpqui",
"UpsertQuotesEndpointURL": "https://preview-rls09.congacloud.com/api/
quote/v1/quotes/upsert-quote",
"LoginURL": "https://login-rlspreview.congacloud.com",
"XAEndpoint": "https://xauthor-stg.congacloud.io",
"SignEndpoint": "https://coreapps-rlspreview.congacloud.com",
"ComposerEndpoint": "https://coreapps-rlspreview.congacloud.com"
}
```

4. Click **Save**. Your new admin setting is saved and added.

Next Step: Create a new Auth. Provider

Creating a new Auth. Provider

Authentication providers allow you to configure external identity providers to authenticate users in your Salesforce org. This feature allows users to log in to Salesforce using their credentials from another trusted system, streamlining the login experience and improving security. This provider is particularly useful when users already have accounts with external identity providers, and you want to integrate with their existing authentication mechanisms.

Prerequisites

Field	Description
Consumer Key	Consumer key and consumer secret you received from the CloudOps team after onboarding your tenant (refer to the tenant onboarding topic).
Consumer Secret	
Authorization Endpoint URL	Enter the authorization endpoint URL. For the preview environment: https://login-rlspreview.congacloud.com/api/v1/auth
Token Endpoint URL	Enter the token URL. Use any of the following URLs as per your Conga instance to access the OAuth2 token service. • For the Preview Environment: https://login-rlspreview.congacloud.com/api/v1/auth/connect/token • For the Production Environment: https://login-rls.congacloud.com/api/v1/auth/connect/token

To configure a Salesforce authentication provider

1. Go to **Setup**. In the Quick Find box, Enter `Auth. Providers` and click **Auth. Providers** in the suggestions. The Authentication Providers page is displayed.
2. Click **New** and select Open ID Connect from the Provider Type drop-down menu.
3. Enter the name, which must be the same as the admin setting name created in the previous topic (Creating a New Admin Setting). For example, APTS_RLPInstanceSettings.
4. The URL Suffix name is generated automatically based on the name of the auth. provider.
5. Enter the connected environment's consumer key and consumer secret, authorize endpoint URL, and token endpoint URL, as mentioned in the prerequisites. Leave the remaining fields as they are.
6. Click **Save**.

Next Step: Configuring Remote Site Settings

Creating Remote Site Settings

Remote Site Settings are a security feature that controls access to external web resources from your Salesforce organization. These resources can include services, APIs, or websites that your Salesforce org needs to communicate with. By default, Salesforce blocks calls to external URLs from within its platform due to security considerations. However, you can use

Remote Site Settings to explicitly allow these calls. Remote site settings help you maintain a secure environment while allowing your Salesforce organization to interact seamlessly with external web resources as needed for your business processes.

In this case, you must create the following remote site settings.

- RLPLoginURL
- RLPInstanceURL

Prerequisites

- Remote site URL for RLP login
- Remote site URL for RLP instance

Reach out to the Conga CloudOps team for the details.

To configure remote site settings

1. Go to **Setup**, enter `Remote Site Settings` in the Quick Find box, and select it.
2. Click **New Remote Site**.
3. Enter the following details:
   a. **Remote Site Name**: Enter a remote site name. For Example, `RLPLoginURL` .
   b. **Remote Site URL**: Enter the remote site URL. The remote site URL will be specific to the associated RLP environment. For example:
      For a preview Environment: https://login-rlspreview.congacloud.com
      For a production Environment: https://login-rls.congacloud.com
   c. **Disable Protocol Security**: Leave this checkbox unselected.
   d. **Description**: Enter a description.
   e. **Active**: Select the checkbox.
4. Click **Save & New**.
5. Enter the following details:
   a. **Remote Site Name**: Enter a remote site name. For example, `RLPInstanceURL` .
   b. **Remote Site URL**: Enter the remote site URL. The remote site URL will be specific to the associated RLP environment. For example:
      For a preview environment: https://preview-rls09.congacloud.com.
      For a production environment: https://prod-rls10.congacloud.com
   c. **Disable Protocol Security**: Leave this checkbox unselected.
   d. **Description**: Enter a description.
   e. **Active**: Select the checkbox.
6. Click **Save.**

Next Step: Creating a New Static Resource

Creating a New Static Resource

Static resources in Salesforce enable you to upload and reference various types of content, such as archives (.zip and .jar files), images, style sheets, JavaScript, and other files, specifically for use within your Salesforce organization.

Prerequisites

- The QuoteFieldNameMap.json file is downloaded and saved to your system.

⚠️ DO NOT change the file name or file type while saving this mapping file.

ⓘ **Best Practices**

- **Create a New Custom Static Resource:** Ensure you create a new custom static resource in the target organization. The existing static resources from managed packages are outdated and do not work, which causes errors.
- **Update Mapping File with Custom Fields:** Add any custom fields to the mapping file and update the static resource to sync these fields with RLP.
- **Verify Fields in RLP:** Ensure that the fields you add to the mapping file already exist in RLP. If not, you encounter a "metadata for field xxx not found" error.

To Create a Static Resource in a Salesforce Org

1. Go to Setup > App Setup > Develop > Static Resources.
2. Click **New** and enter `QuoteFieldNameMap` in the Name field.
3. Click **Choose file** and upload the mapping file.
4. In the Cache Control dropdown, select one of the following values.
   a. **Private:** Select this to restrict API access to authenticated users only.
   b. **Public:** Select this to share static resource data cached on the Salesforce server with other users in your organization.
5. Click **Save**.

Next Step: Creating the Configure Products Button for RLS Flow

Creating the Configure Products Button for RLS Flow

Create the Configure Products button to initiate the quote-to-cash process in the RLS flow.

To create a Configure Products button

1. Go to Setup > Object Manager and search for the Quote/Proposal object.
2. Click **Quote/Proposal**. A quote details page is displayed.
3. In the left menu, click **Fields and Relationships** and then click **New**. The New Custom Field page is displayed.
4. In the *Step 1. Choose the field type* page choose **Formula** as Data Type and click **Next**.
5. In the S*tep 2. Choose the output type* page, enter the flow name as required. For example, `Configure Products(RLS Flow)` in the Field Label. For Formula Return Type, choose **Text**.
6. In the *Step 3. Enter the formula* page, copy the following formula and paste it under the Simple Formula tab. Click **Check Syntax** to validate the formula.

   ```
   IF ( LEN( Apttus_QPConfig__PriceListId__c ) > 0 , HYPERLINK("/apex/
   Conga_CCPQ__RLPCart?boid="&Id& "&flow=system&idp=Salesforce", IMAGE("/resource/
   Apttus_QPConfig__Button_Configure", "Configure Products"),"_self"), NULL)
   ```

   > ⓘ • Enter the name of your flow setting in the"flow" parameter in the above formula. In our example, it is "system".
   > • Update the "idp" parameter in the formula based on the org type. Use "Salesforce" for production orgs and "SalesforceSandbox" for sandbox orgs.

7. Click **Next.**
8. In the *Step 4. Establish field-level security* page, Select the profiles to which you want to grant edit access to this field via field-level security and click **Next.**
9. Choose the page layouts on which you want to display this field and click **Save**.

Next Step: Enabling OAuth Settings

Enabling OAuth Settings

You must set up the OAuth and OpenID flows for your connected app.

If you turn on this setting, then the whole org is affected.

To enable OAuth settings

1. Navigate to Setup > Settings > Identity > Oauth and OpenID Connect Settings. A list of flows is displayed.
2. Turn on the Allow OAuth Username-Password Flows setting, if it is not enabled.
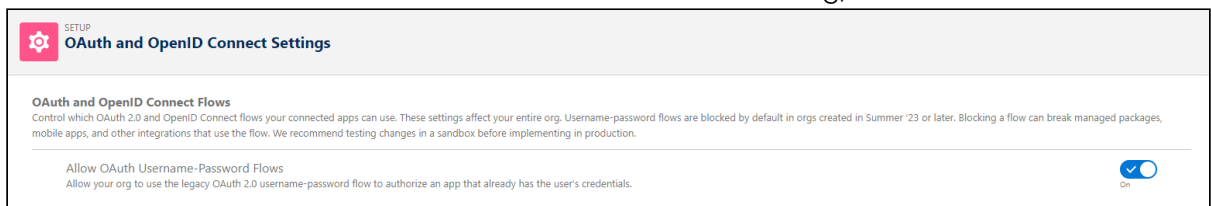
Next Step: Creating a Connected App

Creating a Connected App

You must configure a connected app in the onboarded Salesforce org to provide authentication and authorization for the Conga Platform services.

Prerequisites

Callback URL of your Salesforce org: The callback URL is your domain URL (Setup > My Domain).

To view your current domain URL:

1. Go to **Setup.**
2. Enter `my domain` in the Quick Find box and select **My Domain**.
3. Copy the URL from **My Current Domain URL**. This URL is specific to the instance you are connected to. For example, <Domain Name>.my.salesforce.com.

To create a connected app

1. Log in to the Salesforce portal as an administrator.
2. Go to **Setup**. In the Quick Find box, Enter `App Manager` and click **App Manager** in the suggestions . The Lightning Experience App Manager page appears.
3. Click **New Connected App**.

4. Fill in the following details in the **Basic Information** section.

Field	Description
Connected App Name	Enter the connect app's name, which displays in the App Manager and on its App Launcher tile. For example, RLSInstance.
API Name	The API name is generated automatically based on the name of the connected app.
Contact Email	Enter the email address of the administrator managing the connected app.

5. Fill in the following details in the **API (Enable OAuth Settings)** section.

Fields	Description
Enable OAuth Settings	Select this to define the OAuth settings. When you enable this field, additional settings are displayed under the **API (Enable OAuth Settings)** section.
Enable for Device Flow	Select this to enable the connected app for an external application.
Callback URL	The callback URL is your current domain URL. Refer to the prerequisites section to know your domain URL. When you select the Enable for Device Flow field, the callback URL is generated by default. Add your current domain URL to the callback URL. You can also add other URLs in separate lines.
Use Digital Signatures	Leave this option unchecked.
Selected OAuth Scope	Select **Full access (full)** and move to Selected OAuth Scopes by clicking the **Add** arrow.
Require Secret for Web Server Flow	Select this to require the connected app to provide a consumer secret for authorization.

Fields	Description
Require Secret for Refresh Token Flow	Select this option to include the connected app's client secret in the refresh token flow's authorization request. If you don't select this option and a connected app sends the client secret in the authorization request, Salesforce still validates it.
Enable Client Credentials Flow	Select this option to let this connected app use the OAuth client credentials flow.
Enable Authorization Code and Credentials Flow	Select this option to let this connected app use the authorization code and credentials flow. This flow is supported only for Experience Cloud users.

6. Leave all other fields blank. Click **Save**.

> ⚠ Changes can take up to 10 minutes to take effect.

After you create a connected app, the system generates a consumer key and consumer secret to enhance the security of your connected app. The consumer key serves as a unique identifier for the customer within the Salesforce system, while the consumer secret validates the ownership and authorization associated with the consumer key.

To capture consumer key and consumer secret

1. Log in to the Salesforce portal as an administrator.
2. Go to Setup. In the Quick Find box, Enter `App Manager` and click **App Manager** in the suggestions.
3. Find the connected app, click the drop-down (▼) icon, and then select **View**.
4. Next to Consumer Key and Secret, click **Manage Consumer Details**.
   You're prompted to verify your identity using one of your registered methods.
5. In the Consumer Details section, click **Copy** next to Consumer Key and Consumer Secret and save them for future use.

Next Step: Configuring the Conga Platform

Configuring the Conga Platform

After configuring your Salesforce org, you must configure Conga Platform.

To launch Conga Platform

1. Click the Login URL (Go to URL) in the welcome email.
2. Enter the username and password.
3. Click **Sign-in**. A Launch your Conga Experience Page is displayed.
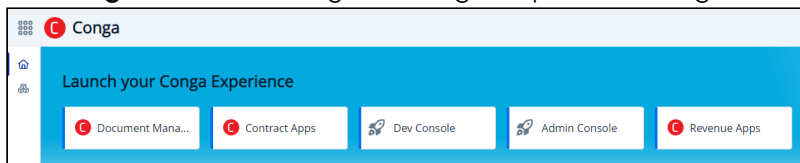
4. Click **Admin Console**. A Platform Administration page is displayed.

> ⓘ Start with the Managing External Integrations and Conga API Connections topic and follow the steps for the successful setup.

- Managing External Integrations
- Adding a New User
- Configuring DataSync Infrastructure
- Running the DataSync

Managing External Integrations

The External Integration tab allows administrators to view and manage users authorized through external organizations. You can add one or more external integrations; however, you can make only one as a default IDP.

To add a new external integration

1. Log in to the Conga Platform as an admin user. A Conga Platform welcome page is displayed.
2. Click the **Admin Console** tile. A Platform Administration page is displayed.
3. Click **Organization**.

4. Open the External Integrations tab and click **Add New**, raising the Add New External Integration screen.
5. Choose **Identity Provider** (Salesforce or Salesforce Sandbox) from the drop-down list and click **Authorize**, to open the selected IDP's login screen.
6. Log in with your onboarded Salesforce org credentials.
   If you get any access-related prompts during this process, please allow full access. When you log in, the selected identity provider's external identifier auto-populates the External ID field.
7. Enter the type of this integration in the **Type** text field. It is a free-form text field. You can add types such as Dev, QA, UAT, Prod, etc.
8. Add the description.
9. Enable the **Default IDP** toggle to make it a Default External Integration IDP.
10. Click **Save**.

To configure an API to API connection

You must establish an API-to-API connection between your Salesforce org and Conga Platform. This connection ensures seamless communication and data exchange between your Salesforce org and Conga Platform.

1. Log in to the Conga Platform as an admin user. A Conga Platform welcome page is displayed.
2. Click the **Admin Console** tile. A Platform Administration page is displayed.
3. Click **Organization**.
4. Open the **Conga API Connections** tab and click **Add API-to-API Connection**.
5. Enter values in the following fields:

Field	Description
External Integration	Select the respective **External Integration** for which you want to configure the connection.
Client Name	Enter the client's name.
Description	Enter the appropriate description.

6. Click **Save & Generate Client Secret** to generate the Client ID.
7. Click **Copy** next to **Client ID and Client Secret.**
8. Save them for future use.

Next Step: Adding a New User

Adding a New User

You can manage Conga Platform users using the User Administration user interface. The Users tab allows you to:

- Create, update, and manage users.
- Import user information.
- Assign roles to users.
- Set default options for users. For example, locale, time zone, date and time format, and other preferences.

To create a new user

1. Log in to the Conga Platform as an admin user.
2. Click the App Launcher (⚏) icon from the top-left corner, then go to **Admin Console** > **Users**.
   A list of users is displayed.
3. Click **Add** to add a new external user. The User Details screen appears.
4. Enter values in the following fields:

Field Name	Description
User Type	Select an external user from the dropdown list.
ExternalId	Enter the user's external ID. This can be any unique alphanumeric text used to identify the user. This ID is used when migrating users from one environment to another. You can enter up to 2000 characters in this field. ⓘ • For the Salesforce identity provider user, the user's unique Salesforce ID (18 digits) must be added as an external ID. For more information on how to get the 18-digit user ID, see Getting Salesforce User ID. • ExtetnalId is optional for Conga IDP users.

Field Name	Description
User Name	Enter the user's unique user name. The Conga IDP user uses this username to log onto the Conga Platform. You can enter up to 255 characters in this field.
First Name	Enter the user's first name. You can enter up to 255 characters in this field.
Last Name	Enter the user's last name. You can enter up to 255 characters in this field.
Email	Enter the user's email address. It must be a valid email address in the form: test@abc.com, test@abc.in, etc. You can enter up to 255 characters in this field.
Role	Enter the keyword to search and select the appropriate role to be assigned to a new user. A role is assigned to users performing similar tasks and consists of a set of permissions. For more information, see Managing Roles and Permission Groups.
Permission Groups	Select permission group(s) from the list. Permission groups are assigned to users for additional access that is not part of the selected role. Permission groups determine a given user's access to objects and records, and permissions to perform actions in the Conga Revenue Lifecycle Platform.
Timezone	Enter the keyword to search and select the time zone from the list. ⓘ If you do not provide a time zone, the application uses the organization's time zone and assigns it to the user.
Locale	Enter the keyword to search and select the locale from the list. ⓘ If you do not provide a locale, the application uses the organization's locale and assigns it to the user.
Currency	Enter the keyword to search and select the currency from the list. ⓘ If you do not provide a currency, the application uses the organization's currency and assigns it to the user.

Field Name	Description
Alias	Enter the user's alias name. You can enter up to 255 characters in this field.
Language	Enter the user's language. For example, `en-IN`. You can enter up to 255 characters in this field.
Address	Enter the user's full address. You can enter up to 64000 characters in this field.
Phone Number	Enter the user's phone number. You can enter up to 255 characters in this field.

5. Click **Save**.

> ⓘ The Conga IDP user receives a Welcome email that includes the username, the first-time login password, and organization information. The user can generate a new password using the default first-time login password.

Newly created users are in an active state by default. Based on the Timezone, Locale, and Currency selections, the Positive Currency Format, Negative Currency Format, Short Date Format, Log Date Format, Time Format, Decimal Symbol, Digit Group, Digit Grouping Symbol, and Negative Number Format information will be added to the new user. All of these fields can be changed while editing the user.

Activating or Deactivating Users

An administrator can activate or deactivate existing users from the user details page. Users can be granted role-based activation and deactivation access. A deactivated user cannot log in to the Conga Platform Administration portal.

To activate or deactivate a user

1. Log in to the Conga Platform as an admin user.

2. Click the App Launcher (⚏) icon from the top-left corner and go to **Admin Console** >
   Users.

   A list of users is displayed.

3. Click the More (⋮) icon next to the user name and select **Activate** or **Deactivate**.

Next Step: Configuring DataSync Infrastructure

Configuring DataSync Infrastructure

After completing the external integration (Salesforce or Salesforce Sandbox) and adding admin users, the next step is to synchronize data from the source system to the Conga Platform. Before synchronizing, you must connect the source and target systems. This section describes the required configurations.

To configure data sync infrastructure

1. Log in to the Conga Platform as an admin user. A Conga Platform welcome page is displayed.
2. Click the **Admin Console** tile. A platform administration page is displayed.
3. Click **DataSync**.
4. Click the More (⋮) icon on the Data Sync UI, and then click **Settings**. The Settings page is displayed. By default, the Salesforce Settings tab is selected.
5. In the Salesforce Settings tab, enter the values in the following fields:

Field	Description
Username	Enter your Salesforce account username.
Password	Enter your Salesforce account password. When you enter the password, dots replace the characters for security reasons. To see or hide the characters, click the eye icon.
Token URL	Enter the token URL. You may use either a sandbox or production instance URL to access the OAuth2 token service. • Sandbox URL: https://test.salesforce.com/services/oauth2/token • Production URL: https://login.salesforce.com/services/oauth2/token

Field	Description
Secret	Enter the security token in this field. When you enter the security token, dots replace the characters for security reasons. To see or hide the characters, click the eye icon.
	To get the secret, reset the security token associated with the Salesforce account. After the security token is reset, the new security token is sent to the email address specified in the Salesforce personal settings. For step-by-step instructions, see Reset Your Security Token.
	If you are unable to see the option to reset your security token, see Troubleshooting.
Client ID	Enter the consumer key generated after creating the connected app.
Client Secret	Enter the consumer secret generated after creating the connected app. When you enter the client secret, dots replace the characters for security reasons. To see or hide the characters, click the eye icon.
	For more information on creating a connected app and getting the client ID and secret, see Create a Connected App.
Domain URL	Enter the domain URL for the Salesforce portal. We recommend using the classic URL format that ends with ".salesforce.com."
	ⓘ The system does not currently support URLs in Lightning Mode format, which end with ".lightning.force.com".
	For example, use https://<domainname>.my.salesforce.com/ URL instead of https://<domainname>.develop.lightning.force.com/ URL.
Salesforce API Version	It is advisable to mention the latest API version. For more details on getting the API version information, see Find Salesforce Edition and API version.

6. Click **Save**.
7. Click the Platform Settings tab.
8. Enter the values in the following fields:

Field	Description
Client ID Client Secret	Enter the client ID and client secret generated after creating the API-to-API connection. When you enter the client secret, dots replace the characters for security reasons. To see or hide the characters, click the eye icon. For more information on creating the API-to-API connection and getting the client ID and secret, see "To Configure API to API Connection" in the Conga API Connections topic.
Token URL	Enter the Conga Platform token URL. **For example:** Preview environment: https://login-rlspreview.congacloud.com/api/v1/auth/connect/token Staging environment: https://login-rlsstg.congacloud.io/api/v1/auth/connect/token Production environment: https://login-rls.congacloud.com/api/v1/auth/connect/token

9. Click **Save**.
10. Click the **Custom Settings** tab. A list of Flows and Settings are displayed.
11. Click all CPQ-related flows and settings.

Next Step: Running DataSync

Running the DataSync

After you onboard your tenant, all the CPQ and CRM OOB objects in Salesforce are automatically mapped to the Conga Platform. Once the configurations on both Salesforce and Conga Platform are completed, sync the master data objects like Products, Categories, Pricelists, Accounts, Opportunities etc to the RLP object database using the Data Sync tool.

To run data sync manually

1. Log in to the Conga Platform as an admin user. A Conga Platform welcome page is displayed.
2. Click the **Admin Console** tile. A platform Administration page is displayed.
3. Click **DataSync**. The Mapped Objects for Data Sync page appears.

> ⓘ Conga recommends enabling one master object at a time and then performing a data sync.

4. Click **Sync Now**. The Sync Now window appears.

Field	Description
Sync Type	Select the synchronization method that suits your needs: > ⓘ If you are running data sync for the first time for the selected objects, select **Full.** • Full: Resyncs all records, including previously synced ones. • Incremental: Syncs only new records or those not on the Conga Platform yet.
Sync Attribute Data	Enable this toggle to sync attribute data (which may include fields, metadata, or additional information) associated with the object.

> ⓘ The system displays a validation alert if the entered Salesforce and Platform Settings information is invalid.

5. Click **OK**. The data sync process begins, and you are redirected to the Run History tab.
6. Click **Run History** to view sync details and any associated errors.

Use Case: Running DataSync for the Product Object

1. Log in to the Conga Platform as an admin user. A Conga Platform welcome page is displayed.
2. Click the **Admin Console** tile. A platform administration page is displayed.
3. Click **DataSync**. The Mapped Objects for Data Sync page appears.
4. On the Data Sync user interface, click the More (⋮) icon > click **Disable All**. A confirmation pop-up is displayed. Click **Yes**.
5. Turn on the **Is Enabled?** option next to the Product object.
6. Click **Sync Now**. The Sync Now window appears.

Field	Description
Sync Type	Select the synchronization method that suits your needs: ⓘ If you are running data sync for the first time on the product object, select **Full**. Otherwise, select **Incremental.** • Full: Resyncs all records, including previously synced ones. • Incremental: Syncs only new records or those not on the Conga Platform yet.
Sync Attribute Data	Enable this toggle to sync attribute data (which may include fields, metadata, or additional information) associated with the object.

ⓘ The system displays a validation alert if the entered Salesforce and Platform Settings information is invalid.

7. Click **OK**. The data sync process begins and you are redirected to the Run History tab, where you can view sync details and any associated errors.
8. You can verify the sync status by clicking on the pipeline ID in the run history tab. If the data sync fails for any reason, click the **Errors** button to download the error report.
9. Follow the same procedure for other required master objects.

A successful data sync for all master objects indicates your Salesforce organization is connected to the Conga Platform.

Next Step: Configuring Revenue Admin Settings

Configuring Revenue Admin Settings

After running the data sync on all CPQ and CRM master objects, perform the following configurations on the CPQ on Conga Platform to load appropriate pages for various sections, such as Catalog, Configuration, Cart, and Asset.

The following are the minimum required configurations to create quotes using Conga RLS. You can also make additional configurations to the following settings based on your business requirements.

Configuring Revenue Admin Application Settings

Configuring General Settings

1. Log in to the Conga Platform as an admin user.
2. Click the App Launcher (⬚) icon from the top-left corner and go to Revenue Apps. A Proposals page is displayed.
3. In the left navigation pane, click the Revenue Admin Icon.
4. Under ADMIN SETUP, go to Application Settings > Global Settings > General.
5. Enter the following details.

Setting	Description
System Global	
Keep Abandoned Carts	Turn this setting on to indicate whether abandoned carts must be kept.
Bypass Shopping Cart	Turn this setting on to bypass the shopping cart. CPQ displays the **Update Price** and **Finalize** buttons and hides the **Go To Pricing** button on the configuration page.
Field Precision	
Percentage Field Precision	Enter a number to specify decimal places for percentage precision. For example, number 5.
Currency Field Precision	Enter a number to specify decimal places for currency precision. For example, number 5.
Quantity Field Precision	Enter a number to specify decimal places for quantity precision. For example, number 5.
Term Field Precision	Enter a number to specify decimal places for selling term precision. For example, number 5.

Setting	Description
Finalization and Approvals	
Auto Finalize On Cart Approval	Turn this setting on. This indicates whether the cart is auto-finalized. CPQ automatically finalizes the cart when approved and ready for finalization.
Auto Sync On Cart Approval	Turn this setting on. This indicates whether the cart is auto-synchronized. CPQ automatically synchronizes the cart when approved and ready for finalization.

6. Click **Save**.

Configuring Catalog Settings

1. Log in to the Conga Platform as an admin user.
2. Click the App Launcher (⚏) icon from the top-left corner and go to Revenue Apps. A Proposals page is displayed.
3. In the left navigation pane, click the Revenue Admin icon.
4. Under ADMIN SETUP, go to Application Settings > Global Settings > Catalog.
5. Enter the following details and click **Save.**

Setting	Description
Display Settings	
Enable Location	Turn this setting ON. This enables users to select a location for a cart. Line items will have the current cart location assigned to them.
Disable Add Another	Turn this setting OFF. Turning this setting on prevents the user from adding multiple instances of the same product.
Hide Configure Action	Turn this setting OFF. Turning this setting on hides the Configure action for a product on the catalog page.
Listed Products Column 2	Select **Product Description** from the dropdown.
Listed Products Column 3	Select **Product Family** from the dropdown.

Setting	Description
Hide Single Top Category	Turn this setting ON. This hides the top category. This is useful only when you have a single root category and want a cleaner look for its child categories, without the root category.
Enable Category Filter	Turn this setting ON. This enables Narrow Your Search category filtering during product search. It filters the categories based on search results and displays them in the Narrow Your Search area. For example, if the search result has products from only two categories, the Narrow Your Search section displays only those two categories.
Hide Product Image	Turn this setting OFF. Turning this setting on hides the product image from the catalog page.

Configuring Config Settings

1. Log in to the Conga Platform as an admin user.
2. Click the App Launcher (⋮⋮⋮) icon from the top-left corner and go to Revenue Apps. A Proposals page is displayed.
3. In the left navigation pane, click the Revenue Admin Icon.
4. Under ADMIN SETUP, go to Application Settings > Global Settings > Config.
5. Enter the following details and click **Save.**

Setting	Description
Display Settings	
Hide Price Column	Turn this setting on to disable the price column from the options page. By default, this setting is turned off.
Listed Options Column 2	Enter the API name of the product field to be displayed in the Listed Options section.
Listed Options Column 3	Enter the API name of the product field to be displayed in the Listed Option section.

Setting	Description
Show Option Quantity	Turn this setting on to display the quantity in the body of the options pages.
Disable Add To Cart for Option Products	Turn this setting on to hide the Add To Cart and Configure buttons for an option product on the catalog page. This is helpful to prevent your sales representative from selling an option product without its bundle product.
Show Tab View	Turn this setting on to display option groups in a tab view instead of sections on the options page.
Bundle Description Field	Enter the field names from the product object to be displayed as bundle description.
Enable Option Page Search	Turn this setting on to allow option search on the bundle configuration page.
Collapse All Leaf Option Groups	Turn this setting ON. This option defines whether the child option groups inside a bundle are expanded or collapsed by default.
Miscellaneous	
Enable Field Expressions	Turn this setting on to enable field expressions.

Configuring Cart Settings

1. Log in to the Conga Platform as an admin user.
2. Click the App Launcher (⚏) icon from the top-left corner and go to **Revenue Apps**. A Proposals page is displayed.
3. In the left navigation pane, click the Revenue Admin Icon.
4. Under ADMIN SETUP, go to Application Settings > Global Settings > Cart.
5. Enter the following details and click **Save.**

Settings	Description
Display Settings	

Settings	Description
Enable Auto Sequencing For Options	Turn this setting on to enable auto-sequencing for options. If you do not turn this setting on, the sequencing of option line items on the cart will be based on the user selection sequence rather than the bundle options sequence.
Show Recommended Products Cart View	Turn this setting on to display the recommended products component in the Cart Detail View page.
Product Display Max Length	Enter the maximum number of characters to be displayed for the product name on the cart page. If the product name exceeds the maximum length, CPQ displays an ellipsis or wraps the product name. The default value is 21.
Show Attributes In Cart	Turn this setting on to display the specified attributes applicable to the line items in the cart. Attributes appear in the product dialog.
Expand Bundles In Cart	Turn this setting on to expand all attributes and options of a bundle product in the cart.
Hide Copy Action	Turn this setting on to hide the copy action in the cart view.
Group By Fields	Enter the API names of those line item fields that are of the type: Lookup, Formula Fields, and Text. These line item fields are displayed under the Group By drop-down on the cart page.
Hide Grand Total	Turn this setting on to hide the grand total line on the cart page.
Hide Subtotals In Cart	Turn this setting on to hide subtotals in the cart view.
View Cart Total Custom Fields	Enter the list of custom fields from the summary group object to be displayed in the custom view cart page. Each field API name must be separated by a comma or a new line.
Disable Cart Activity History	Turn this setting on to disable the generation of activity history in product configuration. If you turn this setting on, CPQ does not add entries to the Activity History-related list when you create, finalize, or reconfigure a product configuration.

Settings	Description
Disable Cart Async Notification	Turn this setting on to disable the email notifications you receive after completion of any async operation such as cart finalization and pricing in every type of quote or agreement.
Enable Contextual Totals	Turn this setting on to display the subtotals and totals section at the bottom of the cart page.
Grid Render Threshold	Enter the number of rows to load when the cart grid renders. This setting improves the grid scroll performance, with a default value of 20. You must adjust the value based on the number of lines displayed on the cart.
Hide Cart Views	Turn this setting on to hide cart views on the cart.
Hide Cart Views Creation	Turn this setting on to hide cart view creation.
Column Render Threshold	Enter the number of columns to load when the cart grid renders. This setting improves to improve the grid horizontal scroll performance, with a default value of 10. Must be adjusted to a higher value based on the number of columns displayed in a cart.
Enable Paginated Grid	Turn this setting on to enable pagination for the cart grid.
Field Set Settings	
Line Item Fields to Revalidate Invalid Price List Items	Select the line item fields to be used to revalidate invalid price list items. This field set controls a list of display fields in the revalidation pop-up for price list item changes.
Line Item Fields to Revalidate Product Structures	Select the line item fields to be used to revalidate product structures. This field set controls the list of display fields in the revalidation pop-up for product structure changes.
Advanced Settings	
Enable Optional Items	Turn this setting on to enable optional items in the cart.
Max Allowed Lines For Mass Actions	Enter the maximum number of line items (supported by the product) to be allowed for performing mass actions on the cart page.

Settings	Description
Run Validation Callback On Add	Turn this setting on to invoke the Validation Callback when adding a product.
Save On All Actions	Turn this setting on to save the configuration when you perform any action on the cart page.
Mini Cart Settings	
Hide Selected Products Column1	Turn this setting on to hide the column that displays the product name.
Selected Products Column2	Enter the API name of the line item field to be displayed in the selected products section. Example value: ProductId__r.ProductCode
Selected Products Column3	Enter the API name of the line item field to be displayed in the selected products section. Example value: ProductId__r.ProductCode
Selected Products Column4	Enter the API name of the line item field to be displayed in the selected products section. Example value: ProductId__r.ProductCode

Next Step: Flow Settings

Configuring Flow Settings

When you launch the flow settings, the Flow Settings page is displayed. All the existing flows are listed on this page with their details. There are various functionalities available for you to manage the flows.

Navigating the flow settings

1. Log in to the Conga Platform as a revenue admin user.

2. Click the App Launcher (⊞) icon from the top-left corner and go to Revenue Apps. A Products page is displayed.
3. Go to ADMIN SETUP > Application Settings > Flow Settings. A list of flows is displayed.

> ⓘ Start with the Cart Page Settings topic and follow the steps.

- Cart Page Settings
- Catalog Page Settings

Cart Page Settings

You can configure system flows to make specific actions available on the cart page.

> ⓘ The following are the minimum required settings (display actions, column settings, etc.) that you must configure to make them available on the cart page.

To configure the flow settings

1. Log in to the Conga Platform as a revenue admin user.
2. Click the App Launcher (⊞) icon from the top-left corner and go to Revenue Apps. A Products page is displayed.
3. Go to ADMIN SETUP > Application Settings > Flow Settings. A list of flows is displayed.
4. Click **System**. A Flow Settings page is displayed. By default, a Cart Page is selected.
5. In the Action Settings tab, enter the following minimum settings required to load the cart page.

> ⓘ
> - Click **Add New** or the add icon (➕) to add an action (custom button).
> - Click the delete icon (🗑) to delete an action. The delete icon is visible only for custom actions added by you.
> - Drag the display actions using the icon ⠿ next to a display type to rearrange the cart action sequence for the cart page.

Action	Action Label Name	Display As	Action Area	Is Enabled	Always Display
GenerateDocument	GenerateDocument	Action	Center	Yes	Yes

Action	Action Label Name	Display As	Action Area	Is Enabled	Always Display
EditPriceAgreement	EditPriceAgreement	Action	Center	Yes	Yes
Finalize	Finalize	Action	Center	Yes	Yes
Abandon	Abandon	Action	Center	Yes	Yes
Save	Save	Action	Center	Yes	Yes
QuickSave	QuickSave	Action	Center	Yes	Yes

6. Click **Save**.
7. Click the Column Settings tab. The following column settings are displayed.
   - Cart Line Item
   - Total Columns
   - Price Ramp
   - Usage Price Tier
   - Tiered Price
   - Related Price
   - Adjustment Line Item
   - Mass Update
   - Agreement Price Rule
8. In the Cart Line Item settings, enter the following and click **Save**.

Field Name	Style	Style Class	Is Editable
BasePrice	`text-align:right`	`aptCurrency`	Yes
Quantity	`text-align:right`	`aptQuantity`	Yes
SellingTerm	`text-align:right; width:60px;`	`aptQuantity`	Yes
NetPrice	`text-align:right`	`aptCurrency`	Yes
StartDate	`text-align:right`	`Leave blank`	Yes

Field Name	Style	Style Class	Is Editable
EndDate	`text-align:right`	`Leave blank`	Yes
ExtendedPrice	`text-align:right`	`aptCurrency`	Yes

9. In the Total Columns settings, enter the following and click **Save**.

Field Name	Style Class	Is Editable
AdjustmentType	`aptAdjustmentType`	Yes
AdjustmentAmount	`aptAdjustmentAmount`	Yes

10. In the Price Ramp, enter the following and click **Save**.

Field Name	Is Editable
Quantity	Yes
Selling Term	Yes
SellingFrequency	Yes
ExtendedPrice	Yes
AdjustmentTyoe	Yes
AdjustmentAmount	Yes
StartDate	Yes
EndDate	Yes

11. In the Usage Price Tier, enter the following and click **Save**.

Field Name	Is Editable
TierStartValue	Yes
TierEndValue	Yes
UsageRate	Yes

Field Name	Is Editable
FlatPrice	Yes
IsIncluded	Yes

12. In the Tiered Price, enter the following and click **Save**.

Field Name	Is Editable
Sequence	Yes
TierStartValue	Yes
TierEndValue	Yes

13. In the Related Price, enter the following and click **Save**.

Field Name	Is Editable
Related ChargeType	Yes

14. In the Adjustment Line Item settings, enter the following and click **Save**.

Field Name	Is Editable
Type	Yes
AdjustmentAppliesTo	Yes
AdjustmentType	Yes
AdjustmentAmount	Yes
BenefitQuantity	Yes
RunningTotalAdjustedPrice	Yes

15. In the Mass Update settings, enter the following and click **Save**.

Field Name	Is Editable
Quantity	Yes

Field Name	Is Editable
AdjustmentType	Yes
AdjustmentAmount	Yes
StartDate	Yes
EndDate	Yes

16. In the Agreement Price Rule settings, enter the following and click **Save**.

Field Name	Is Editable
Sequence	Yes

Next Step: Catalog Page Settings

Catalog Page Settings

You can configure action settings for flows to be available on the catalog page.

To configure catalog page settings

1. Log in to the Conga Platform as a revenue admin user.
2. Click the App Launcher (⁝⁝⁝) icon in the top-left corner and go to Revenue Apps. A Products page is displayed.
3. Go to ADMIN SETUP > Application Settings > Flow Settings. A list of flows is displayed.
4. Click **System**. A Flow Settings page is displayed.
5. Click **Catalog Page**.
6. Enter the following in the Action Settings page. These are the minimum settings required to load the catalog page.

Action	Action Label Name	Display As	Action Area	Is Enabled	Always Display
InstalledProducts	InstalledProducts	Action	Right	Yes	Yes
CompareProducts	CompareProducts	Action	Center	Yes	Yes

Action	Action Label Name	Display As	Action Area	Is Enabled	Always Display
GoToPricing	GoToPricing	Action	Center	Yes	Yes
UpdatePrice	UpdatePrice	Action	Center	Yes	Yes
Finalize	Finalize	Action	Center	Yes	Yes
CloseCart	CloseCart	Action	Center	Yes	Yes
Abandon	Abandon	Action	Center	Yes	Yes
Save	Save	Action	Center	Yes	Yes
QuickSave	QuickSave	Action	Right	Yes	Yes

7. Click **Save**.

Now, you can test the quote creation process in the Revenue lifecycle solution (RLS) flow.

Next Step: Creating a Quote from the Proposal

Creating a Quote from the Proposal

After configuring the required settings on the Conga Platform, you can test and verify the quote creation flow using the RLS flow. To achieve this, you must log in to the onboarded Salesforce org to create a new quote or proposal.

Prerequisites

- You have associated the products with the price list you are using for a new quote. For more information, see Associating Price Lists with a Product.
- All required products associated with the price list you are using for new quotes are activated and published. For more information, see Configuring the Products.
- The account you are using for a new quote has been synced with the Conga Platform.
- Document generation is configured for a quote in the Salesforce org. For more information, see Configuring Document Generation for Quote.

You can use the Conga CPQ on the Salesforce interface to enter the necessary information and prepare a quote/proposal request or draft. To complete the quote/proposal, you must

enter details relevant to the proposal. Some of the fields, identified with either a red vertical line or asterricks, are required to save the proposal record. If the required details are not entered or incorrect details are entered, the system displays validation rule errors at the top of the page, highlighted in red.

To create a quote or proposal

1. Log in to Salesforce.
2. Click the App Launcher (⣿) icon from the top-left corner and enter `Conga   CPQ` , and select **Conga CPQ**.
3. Go to the Proposals tab and click **New**.
4. Enter the proposal and opportunity names in the **Propoosal Name** and **Opportunity** fields.
5. Enter an account and price list information in the **Account** and **Price List** fields.
   The price list determines the categories and products you see in the configuration. The account is auto-populated if the proposal is created from an opportunity.
6. Click **Save**. Navigate to the new quote you just created and proceed with product selection.
7. Click the **Configure Products** button you created for RLS flow, raising the catalog page.
8. Select the products from the catalog page. By default, you do not see products on the catalog page if you have not configured the catalog page with the price list selected for your proposal. Hence, place the cursor in the Find Products search box and click **Enter** on your keyboard. A list of products associated with the selected price list is displayed.
9. Add required products to the cart. Some of the products may require configuration. Click the Configure button to complete the product configuration.
10. Click the Mini-cart (🛒) icon to view the summary of added products.
11. In the mini-cart, click **View Cart** to launch the cart in grid view.
12. Click **Finalize** to create a new proposal and open the proposal and quote details.
13. In the Actions section, click **Send Proposal** to open the doc gen page where you can view the generated proposal and such other options as adding supported documents.
14. Click **Email** to display a Send Email pop-up window.
15. Enter **To** and select **Email folder**, **Email Template**. The Subject text box is automatically generated.
16. Click the **Mark Proposal as "Presented"** checkbox and click **Merge Fields**.
17. Click **Send**. The acceptor will receive an email with the proposal details to verify.
    After the quote/proposal has been presented to the trading partner either manually or via email, the **Accept** button is enabled. Selecting Accept signifies that the trading partner has accepted the quote.

18. In the Actions section, click **Accept**. The Approval Stage changes to Accepted.
19. Do one of the following:
    - To create an agreement, click **Create Agreement**.
    - To create an agreement with line items, click **Create Agreement with Line Items**.
    - To synchronize the quote with an opportunity, click **Synchronize with Opportunity**.
        - Synchronizing Quote/Proposal With Opportunities: Synchronization ensures that both the quote/proposal and opportunity accurately reflect the current state of the quote. A best practice for sales pipeline reports and forecasts in Salesforce is to display the amount of revenue and the products or services that are forecasted to be sold.

The contracting process begins with the creation of an agreement.

Managing Products

Products are the primary component of your CPQ system. Using CPQ, you can create individual products and product bundles. Once you've created products, you can associate attributes and options to them. Options are the additional accessory products to sell together with a product. Attributes allow you to associate product specifications such as color, size, or model.

You must associate the products with Categories to classify products into logical groups. You can define hierarchies in a category to further divide the products into the smaller groups. On the Catalog page, CPQ displays the categories and hierarchy containing the products you associate with them.

Using feature sets, you can also configure 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.

- About Products
- Managing Categories and Hierarchies
- Managing Bundles
- Managing Product Groups
- Creating and Maintaining Attributes
- Configuring Product Visibility
- Configuring Product Comparison
- Configuring Numeric Expressions

About 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 about creating products.

- Navigating the Product Page
- Creating Products
- Cloning Products

- [Configuring Product Families](#)

Navigating the Product Page

When you launch the CPQ Admin console, Manage Product page is displayed. All the existing products are listed on this page along with their details. You can configure the columns displayed on the Manage Product Page. There are various functionalities available for you to manage the products.

Product Search

Search for the product to narrow down the product list using the Search Products box. Enter a keyword related to your product details like Product Name, Product Code, Configuration type, description. Search Product box returns all the products that contain that keyword in the product details.

There are two types of search available on the products homepage. Saved Search and Product search query.

> ⓘ Either of the search filters is applicable at a time.

Saved Search provides you with various options to search, sort, and filter products. You can either define search criteria to search for a range of products or use the product search option. The search query uses options provided in the Field, Operator, and value and displays a search result that matches the criteria. Saved Search functionality lets you save a set of filters you define that narrows down the list of products according to your requirement.

Perform the following steps to view and search configured products in the Saved Search:

1. Click **View All** to view all products that are configured in CPQ on the dashboard. If you have queried a filter or a product, then a select **View All** to disable the existing search filter.
2. Click **New Search**. A window that consists of various fields, operators, and values are displayed.
   a. **Field**: All product Fields are available in the drop-down list such as Text, Checkbox, Date, Date/Time, Number, Percent, Picklist, Multi-Select Picklist, Currency.
   b. **Operator**: Operators that are available for selection helps create the filter expression. Operators are displayed based on the field types. For example, a

picklist value will display equal to and not equal operators and a text field will display equal to, not equal to, starts with, contain, does not contain operators. If it is a multi-value picklist, the application displays include and does not include operators.

   c. **Value**: Enter a value.

   d. Click one of the following options to execute the search:

- Click **Save As** to save the search filter and use the filter criteria.
- Click **Apply** to enable the search filter.
- Click **Cancel** to cancel the search criteria.

3. To search a product in the Product Search, enter the name of the product that you want to search in the search field. Products that match the search are displayed.
4. Click field headers in the product's dashboard to sort dashboard based on the field.

You can simply click the saved search record you created and the system filters the search results. You can see the saved searches dropdown list in alphabetical order for better accessibility of your list of saved searches. If you have configured a customized view, click the menu icon (⋮) next to the New Search button. The following options are displayed.

- **Edit View**: Displays a new search window.
- **Save View As**: Displays options to Save the edit view.
- **Remove View**: Removes the filter.

You can also narrow down the list using the column search (inline search). Enter a keyword related to the column in space under the column name. Unlike the Search Product field, column search matches the keyword in that particular column. Use the keywords True and False for the checkbox type columns. You can enter keywords in multiple columns at the same time, the result displayed contains the products with details matching to all keywords you entered in different columns.

Refresh the page to clear all the filters.

Sorting and Adjusting the Columns

You can sort the product by clicking the column names. The products are sorted alphabetically, every time you click the column name the sorting switched between ascending and descending order. You can adjust the column width by dragging the columns. Hover near the column boundaries till two-way arrow appear. Click when the arrow appears and drag left or right to adjust the column width. At the bottom of the page, you can control the number of products to be displayed on each page. You can select the number of products from a pre-defined list. You cannot configure this list. By default, you see 50 records on each page.

Product Details

Product details hold the information about the product. You can view them on the Manage Product page or the product details page. When you click the product name displayed in blue color, the **DETAILS** tab is displayed. All the primary details of the products are displayed on this page. Other tabs on the secondary toolbar hold additional information like options information, price list associated with the product, rules associated with the products. You can edit the details on this page.

You can use the Mini Display functionality to view the details without leaving the Manage Product page. Click the information in any column other than the product name for the respective product. A panel on the right-hand side displays the details of the product. You cannot edit any details on the mini display. Click the edit icon to go to the **DETAILS** tab, to edit the details.

You can see a list of all the existing products displayed in a panel on the left. When you hover the cursor over the product name, product details are displayed in a pop-up. You can configure the pop-up to display other fields. For more information, refer to Configuring the Tooltips.

Creating Products

A Product in CPQ 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. You can associate a product with a

category or a price list. If you created an option product, you must first associate it with an option group.

- 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 through an option group.

To create a product

1. On **App Menu**, click **Apttus CPQ Admin** and then click the **CPQ Admin** tab. The new admin console is launched.
2. On the **Products** menu, click **Manage Products**. Click **New Product**. The product details page is displayed. **Details** tab opens by default.
3. Enter the following product information, as required.

Option	Description
Product Name	Enter a name for the product.
Product Code	The Product Code can be an internal code or a product number to identify the product within your organization.
Product Family	Enter a Product Family to categorize the products. 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. For more information refer to Configuring Product Families.
Uom	Select a unit of measurement.
Product Description	Enter a description of the product. This description is displayed when the user clicks the product name on the catalog page.

Option	Description
Disable Bundle Expansion	Select this checkbox to disable the expansion of the bundle or a sub-bundle on the Cart page. This does not allow the Sales Rep to expand the bundle to view the options and secondary line items. If you only disable the expansion of a sub-bundle, the Sales Rep can still expand the parent bundle. You must create this field in the Product object. The checkbox is only considered if the **Configuration Type** is *Bundle* and **Has Options** is selected. Otherwise, CPQ does not disable expansion. ⓘ You must create this field on the Product object. Refer to Configuring Bundle Expansion on Cart Page.
Configuration Type	Select one of the following types of product: · **Standalone** · **Bundle** · **Option** This field is mandatory.
Must Configure	Indicates that the 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 checkbox 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 Options	Indicates a product as a bundle that has options (bundles/options). You can associated options and option groups to the bundle on the Structure tab. Options and Options groups are not displayed on the Configuration page, is this disabled.
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.
Active	Indicates the product is active. The Product is not displayed on the Catalog page if this setting is not selected.

Option	Description
Product Type	Select a type of product. For example, Equipment, Service, and so on.
Has Search Attributes	Indicates that the product has search attributes and is useful for searching related products based on product attribute values.
Renewal Lead Time	Enter a value when the renewal quote must be created for this product after an order is activated and an asset is generated. By default, the value is 0, which indicates that the renewal quote is created immediately after the order is activated, with the same number of Line Items in the order. If you specify 30, the renewal quote is created 30 days before the Asset End Date.

The fields on the details page are customizable. For more information refer to the Navigating the CPQ Admin User Interface topic.

4. Click **Save**.

A product is created and saved.

Associating a Product with Option Group

1. On **App Menu**, click **Apttus CPQ Admin** and then click the **CPQ Admin** tab. The new admin console is displayed.
2. On the **Products** menu, click **Manage Products**. Select the **Product** created in the previous section. The product details page is displayed. The **Details** tab opens by default.
3. Click the **Associations** tab.
4. Drag and drop the required Category from the Category Hierarchy or
Click **New Category** to create a new category and click the check box next to the required option group you want to associate the new product with.
5. Enter the following product information, as required.

Field	Description
Name	Enter a name for the Option Group.
Label	Enter the label for the Option Group.

Field	Description
Default Search Category	When selected, the **Default Search Category** displays the product by default under the category or sub-category on the catalog page. If not, the user has to first select the category from the category panel or the browse section to see the product.
Expanded By Default	Indicates that the hierarchy is expanded at all times on the catalog page
Include In Totals View	Indicates that the mode is displayed in the totals view on the shopping cart
Hide All Search Filters	Indicates that all the search filters in **Narrow Your Search** and **Refine Your Search** on the Catalog page are hidden.
Search Filter Fields	Selected filter fields are displayed in the **Refine Your Search** section. Select fields from the dropdown menu, and click anywhere on the screen once you have selected all the desired fields.
Long Description	The Long Description is displayed for categories with bundle products on the Options page.
Upload Image	You can add an image like a symbol or an icon for the sub-category. Click **Browse** to select an image from your local machine.

6. Click **Save**.

Creating and Associating Attributes with the New Product

You can create attribute groups and associate them with your product based on your requirements. For more information, see Creating and Maintaining Attributes.

Creating and Associating Product Rules with the New Product

You can create product rules for your product based on your requirements. For more information, see About Manage Rules Page.

Adding Collateral to the New Product

You can add multimedia such as icons, symbols, or images to your product from your local system. This enables sales representatives to identify the correct image while adding and configuring the products on the catalog page.

To add collateral to the new product

1. On **App Menu**, click **Apttus CPQ Admin** and then click the **CPQ Admin** tab. The new admin console is displayed.
2. On the **Products** menu, click **Manage Products**. Select the **Product** created in the previous section. The product details page is displayed. The **Details** tab opens by default.
3. Click the **COLLATERAL** tab. A new pop-up window is displayed where you can add collaterals by uploading a multimedia file, using a multimedia URL address, or using embed code. By default, the UPLOAD FILE tab is selected.
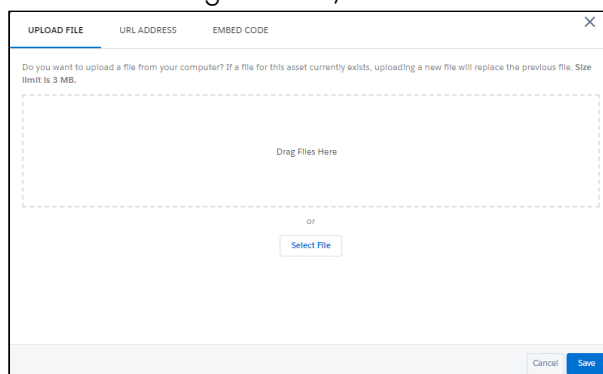
4. To upload a multimedia file.
   a. Using UPLOAD FILE:
      i. Drag and drop the multimedia file or
      ii. Click **Select File** and choose the required file from your local system.
   b. Using URL ADDRESS:
      i. Click the URL ADDRESS tab.
      ii. In the URL field, enter the multimedia file URL address.
   c. Using EMBED CODE

i. Click the EMBED CODE tab.

ii. In the Embed Code field, refer to the sample embed code and enter the multimedia embed code.

5. Click **Save**.

Cloning Products

You can also create new products by cloning existing products. You can clone the product with partial or entire configuration details. This helps you to save time if you want to create a large number of products with a similar configuration.

To clone a Product

Follow the steps below.

1. On **App Menu**, click **Conga CPQ Admin** and then click the **CPQ Admin** tab. The new admin console is launched.
2. On the **Products** menu, click **Manage Products**.
3. From the menu icon (⫶) next to the **New Products** button, click **New from Clone**. A pop-up is displayed.
4. Fill in the required details.

Option	Description
New Product Name	Enter a name for the new product.
Product To Clone	Type the name of the product you want to clone. Or open the details of the product you want to clone and then click **New From Clone**. In this case, the field is populated automatically.
External ID	Select an external ID.
Entire Product	Select this to clone the entire configuration of the product. When you select this all the checkboxes are automatically selected.
Product Details	This indicates that the products details of the products are cloned. This option is selected by default.

Option	Description
Categories	Select this to include the categories associated with the product.
Attributes	Select this to include the attribute groups and attributes associated with the product.
Pricing	Select this to include all the price list and price list item associated with the product.
Options	Select this to include the option groups and options associated with the product.
Product Collateral	Select this to include the media associated with the product.
Product Group	Select this to include the product group associated with the product. If you select this, the new product is added in the product group.
Rules	Select this to include the rules associated with the product. You can select which rule to include.

5. Click **Save.**

A new product is created.

Configuring Product Families

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. On **App Menu**, click **Conga CPQ Admin** and then click the **CPQ Admin** tab. The new admin console is launched.
2. On the Products menu, click **Manage Products**.
3. Search and click the product to open the product details page.
4. On the **Details** tab, from Product Family, select an appropriate value.
5. Click **Save**.

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

Managing Categories and Hierarchies

Categories are high-level logical groupings of products, affecting the way the Sales rep 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 group the total price of the products belonging in a category on the Cart Summary.

There are three 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 through this method only. Most implementations will have an existing hierarchy.
- Offering and Option Group: This category type groups products that can be used as a category and option group. 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).
- 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. You can manage option groups in **Manage Option Groups** menu.

The following topics provide instruction to manage Categories and Category Hierarchy.

- Creating Categories

  Categories are created to group products for the end user's product selection. Categories are also used to group options within bundles.
- Creating Category Hierarchies

  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. 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

The purpose of categories is to group products for the end user's product selection. Categories are also used to group options within bundles. Follow the steps below to create and modify a category.

To create a Category

1. On **App Menu**, click **Conga CPQ Admin** and then click the **CPQ Admin** tab. The new admin console is launched.
2. On the **Catalog** menu, click **Manage Catalog**.
3. Click **New Category**. **Create New Category** pop-up is displayed.

4. Enter a mandatory **Name** and **Label**.
5. Select **Active** to activate the category. By default, the **Active** check box is already selected.
6. Select one of the following from the **Type** picklist.
   - *Offering*
   - *Offering and Option Group*

   Only these two types are allowed for category creation. Option Group type is only available for option group creation on the **Manage Option Group** menu.
7. Click **Save**.

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

You can create category hierarchies based on your business requirement. Select a Parent Category when you create a new category.

To modify an existing Category

1. On **App Menu**, click **Conga CPQ Admin** and then click the **CPQ Admin** tab. The new admin console is launched.
2. On the **Catalog** menu, click **Manage Catalog**.
3. Search and click the category to open the category details page.
4. Modify desired fields as required.

Field Name	Description
Name	Enter a name for the category.
Label	The label name is populated automatically.
Description	Enter a description of the category.
Default Search Category	When selected, **Default Search Category** displays the product by default under the category or sub-category in the catalog page. If not, the user has to first select the category from the category panel or the browse section to see the product.
Expanded By Default	Indicates that the hierarchy is expanded at all times in the catalog page

Field Name	Description
Include In Totals View	Indicates that the mode is displayed in the totals view on the shopping cart
Hide All Search Filters	Indicates that all the search filters in **Narrow Your Search** and **Refine Your Search** on the Catalog page are hidden.
Search Filter Fields	Selected filter fields are displayed in the **Refine Your Search** section. Select fields from the dropdown menu, click anywhere on the screen once selected all the desired fields.
Long Description	The Long Description is displayed for categories that have bundle products on the Options page.
Catalog's Image(s)	You can add an image like a symbol or an icon for the sub-category. Click **Browse** to select an image from your local machine.

5. Click **Save**.

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

Creating Category Hierarchies

Category hierarchy defines the structure of the product catalog. You can build and maintain a category hierarchy on the Category page. 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**.

Prerequisite

You must have an existing Offering category.

To build a Hierarchy

1. On **App Menu**, click **Conga CPQ Admin** and then click the **CPQ Admin** tab. The new admin console is launched.
2. On the **Catalog** menu, click **Manage Catalog**.
3. Search and click the category to open the category details page.

4. The Category details page is displayed, which allows you to perform the following actions for the selected category.
    - Add or Edit a Sub-category
    - Remove a category
    - Associate Products
    - Reorder Categories
5. 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.

In the following section:

- Adding Sub-Categories
- Removing Sub-Categories

Adding Sub-Categories

Sub categories (also known as nodes) categorize products within a category. You can add multiple sub-categories/nodes to categorize products within a category.

Before You Begin

You must have an existing top-level category.

To create a new sub-category within a Category

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

1. On **App Menu**, click **Conga CPQ Admin** and then click the **CPQ Admin** tab. The new admin console is launched.
2. On the **Catalog** menu, click **Manage Catalog**.
3. Search and click the category you desire to be the parent category. The category details page is displayed.
4. Click **ASSOCIATED CATEGORY**. Click **New Category**. Or click **New Category** on the category details page. **Create New Category** pop-up is displayed.

5. On the **Create New Category**, check that the desired parent category is selected. Search or use arrow icons at the bottom to find the desired category.

6. Fill in the information as explained in the table below.

Field Name	Description
Name	Enter a name for the category.
Label	The label name is populated automatically.
Default Search Category	When selected, **Default Search Category** displays the product by default under the category or sub-category in the catalog page. If not, the user has to first select the category from the category panel or the browse section to see the product.
Expanded By Default	Indicates that the hierarchy is expanded at all times in the catalog page
Hide All Search Filters	Indicates that all the search filters on the catalog page in Narrow Your Search and Refine Your Search sections are hidden.
Include In Totals View	Indicates that the mode is displayed in the totals view on the shopping cart
Description	Enter a description of the category.
Long Description	The Long Description is displayed for categories that have bundle products on the Options page.
Upload Image	You can add a small image like a symbol or icon for the sub-category. Click **Browse** to select an image from your local machine.

7. Click **Save.**

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

Removing Sub-Categories

You can remove a sub-category and its nested categories from the Associated Category page.

To remove a sub-category

1. On **App Menu**, click **Conga CPQ Admin** and then click the **CPQ Admin** tab. The new admin console is launched.
2. On the **Catalog** menu, click **Manage Catalog**.
3. Search and click the category you desire to be the parent category. The category details page is displayed.
4. Click **ASSOCIATED CATEGORY**. All the associated sub-categories are listed on this page.
5. Click the remove icon(✕) under the **Action** column for the sub-category you want to remove.
6. Click **Save**.

The sub-category is removed from the hierarchy.

Associating Products to a Category

Following the steps below you can associate products to a category at a category-level. You can also associate a product to a category from the product-level.

> ⓘ You cannot associate products to a category with sub-categories associated to it.

To associate a Product to a Category

1. On **App Menu**, click **Conga CPQ Admin** and then click the **CPQ Admin** tab. The new admin console is launched.
2. On the **Catalog** menu, click **Manage Catalog**.
3. Search and click the category to open the category details page. **DETAILS** tab opens by default.
4. Click **ASSOCIATED PRODUCTS**. If the category has sub-categories, **ASSOCIATED PRODUCTS** tab is not visible. In this Case, click **ASSOCIATED CATEGORY** and then select a sub-category. If you nested sub-categories, navigate to the last sub-category.
5. Drag and drop all the products you want to add in the category from the **All Product** list. You can search for the products using the search bar.
6. Click **Save**.

Your desired product is associated with a category.

Associating Categories to a Product

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

This can be achieved at a category level or a product level. At the product level you can associate categories to a product and at the category level, you can associate products to a category. On the Products page and Category page, you can update and modify the categories and products respectively.

To associate Category to a Product

1. On **App Menu**, click **Conga CPQ Admin** and then click the **CPQ Admin** tab. The new admin console is launched.
2. On the **Products** menu, click **Manage Products**.
3. Search and click the product to open the product details page.
4. Click **ASSOCIATIONS**. Click the category hierarchy icon ( ). **Category Hierarchy** pane is displayed.
5. Drag and drop the desired Categories from **Category Hierarchy** pane. You can search for the categories using the search bar.
6. Click **Save**.

The product is now associated with a category.

> ⓘ For performance implications, it is recommended that you limit the number of products in a single category to 1000.

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 field.
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&productOrderByClause=Ca
tegory_Rank__c NULLS LAST, Name" ,IMAGE("/resource/Apttus_QPConfig__Button_Configure",
 "Configure Products"),"_self"), NULL)
```

On the Catalog page, Products are displayed based on the sort order defined in the custom field.

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 in 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 with a Price List for it to show up in the catalog.

1. In Salesforce Classic, go to the **Price Lists** tab.
2. Select a price list in which you want to reorder categories.
3. Under the related list for Categories, click **Edit** for the category that is related to your target root (top-level) category.
4. Update the **Sequence** fields with the desired number. Set the sequence number relative to other categories to drive the order (Object: Price List Category, Field:

Sequence).

5. Click **Save**.

> ⓘ You must edit the layout to include the field **Sequence** on the related list for Category and on the Price List Category Details page.

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. On **App Menu**, click **Conga CPQ Admin** and then click the **CPQ Admin** tab. The new admin console is launched.
2. On the **Catalog** menu, click **Manage Catalog**.
3. Search and click the category you desire to be the parent category. The category details page is displayed.
4. Click **ASSOCIATED CATEGORY**. All the associated sub-categories are listed on this page.
5. Drag and drop the sub-categories list in the section in the middle in the desired order.
6. Click **Save**.

The system updates the order and displays the hierarchy.

Running a Category Maintenance Job

When you create a category and associate products to the category or change the category definition, you must run a Category Maintenance job. If you do not run category maintenance, the Sales rep may see incorrect hierarchy or totals on the cart page.

After making changes to the existing category definition you can run the Category Maintenance batch job only for changes rather than running for the entire definition. For example, if your category has 10 products and you added 5 more, you can run the Category Maintenance batch job only for the 5 additional products.

For more information, see Running Maintenance Jobs.

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

Managing Bundles

Bundled products represent a combination of standalone products that offer added values. An option group is used to combine product options and then associate that group to a bundled product.

In this section:

- Creating Option Groups
- Associating Options to an Option Group
- Associating Options to a Bundle
- Configuring Multi-Level Bundles
- Cloning a Bundle within the same Quote
- Configuring Bundle Expansion on Cart Page
- Running a Bundle Maintenance Batch Job
- Distinguishing Cloned Options on the Bundle Configuration Page

Creating Option Groups

Option groups are a type of category that is used to group option products together to be associated with bundles. Options group can also be used as categories.

There are two types of option groups:

- *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 options to a bundle product.
- *Offering and Option Group:* 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).

Option groups are further categorized into two types Shared Option Group and Standalone Option Group. You can create these option groups on the **STRUCTURE** tab on the products details page in CPQ Admin. All the option Group you create in the **Manage Options Group** menu are by default shared.

Shared Option Groups can be associated with multiple bundles. Whereas, Standalone Option Groups are bundle specific. Though, you can associate standalone options groups with other bundles. However, only the option group structure that you saved for the first time while creating the standalone option groups is associated with another bundle. The changes you make to the option group structure after saving the first time are not reflected whenever you associated the same option group with other bundles.

When you click **New Shared Option Groups** or **New Standalone Option Groups** a pop-up opens in which you create a new option group. Follow the steps in the *To create an Option Group* below to after you click **New Shared Option Groups** or **New Standalone Option Groups**. When you create an option group in the structure page of a bundle, that option group is automatically associated with the bundle. Shared option groups are also associated with the bundles. Any change structural changes you make are reflected on all the bundles the shared option group is associated with. For example, if you change the options in the option group, the changes are reflected in all the bundles the shared option group is associated with.

To create an Option Group

1. On **App Menu**, click **Conga CPQ Admin** and then click the **CPQ Admin** tab. The new admin console is launched.
2. On the **Products** menu, click **Manage Option Groups**.
3. Click **New Option Group**. **Create New Option Group** pop-up is displayed.
4. Enter a mandatory **Name** and **Label**.
5. Select **Active** to activate the category. By default, the **Active** check box is already selected.
6. Select one of the following from the **Type** picklist.
    - *Option Group*
    - *Offering and Option Group*
7. Click **Save**.

Based on your **Type** selection; an *Option Group* or *Offering* and Option *Group* type options group is created and added.

You can create option group hierarchies based on your business requirement. Select a Parent Option Group when you create a new option group.

To modify an existing Option Group

1. On **App Menu**, click **Conga CPQ Admin** and then click the **CPQ Admin** tab. The new admin console is launched.
2. On the **Products** menu, click **Manage Option Groups**.
3. Search and click the option group to open the option group details page.
4. Modify desired fields as required.

Field Name	Description
Name	Enter a name for the options group.
Label	The label name is populated automatically.
Description	Enter a description of the options group.
Default Search Category	When selected, **Default Search Category** displays the product by default under the category or sub-category in the catalog page. If not, the user has to first select the category from the category panel or the browse section to see the product.
Expanded By Default	Indicates that the hierarchy is expanded at all times on the catalog page
Include In Totals View	Indicates that the mode is displayed in the totals view on the shopping cart
Hide All Search Filters	Indicates that all the search filters on the catalog page in Narrow Your Search and Refine Your Search sections are hidden.
Search Filter Fields	Selected filter fields are displayed in the **Refine Your Search** section. Select fields from the dropdown menu, click anywhere on the screen once selected all the desired fields.

Field Name	Description
Long Description	The Long Description is displayed for categories that have bundle products on the Options page.
Catalog's Image(s)	You can add an image like a symbol or an icon for the sub-category. Click **Browse** to select an image from your local machine.

5. Click **Save**.

The option group is modified and saved. After creating or modifying the options, you must run the Category Maintenance batch job.

ⓘ In the Salesforce Classic user interface select the **Cascade Group Changes** field on the Product Option Group details page to enable a warning message when you modify any shared option group.

Associating Options to an Option Group

You need to associate options with options groups to use them in a bundle.

Prerequisite

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

To Associate Options to Option Groups

1. On **App Menu**, click **Conga CPQ Admin** and then click the **CPQ Admin** tab. The new admin console is launched.
2. On the **Products** menu, click **Manage Option Groups**.
3. Search and click the options group to open the option group details page.
4. Click **ASSOCIATED PRODUCTS**.
5. Drag and drop all the options you want to add in the options group from the **All Product** list. You can search for the options using the search bar.
6. Click **Save**.

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

To Associate Options to Option Groups within a Bundle

1. On the **Manage Products** page, open the products details page of the bundle.
2. Click **Structure**.
3. Click the shortcut icon (‹) under **New Shared Option Group** button. **Option Hierarchy** panel is displayed.
4. On the **Option Hierarchy** panel select a category from the dropdown menu.
5. Drag and drop desired option group and options from the **Option Hierarchy** pane.
6. Click **Save**.

> ⓘ When you modify an option group that is associated with multiple bundles, all the associated bundles are affected.
> Select the **Cascade Group Changes** field on the Product Option Group details page to enable a warning message when you modify any shared option group.

Associating Options to a Bundle

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.

Prerequisite

Has Options in the **Details** tab must be selected. Otherwise, the **Structure** tab is not displayed. For more information refer to Creating Products.

To associate option groups to a bundle

1. On **App Menu**, click **Conga CPQ Admin** and then click the **CPQ Admin** tab. The new admin console is launched.
2. On the **Products** menu, click **Manage Products**.
3. Search and click the bundle you want to associate options with.
4. Click **Structure**. The page displays all the options and option group associated with the bundle.

5. Click the shortcut icon (‹) under **New Shared Option Group** button. **Option Hierarchy** panel is displayed.
6. On the **Option Hierarchy** panel, select Options Groups from the dropdown menu.
7. Search the option group using the search bar and then drag and drop the options group in the middle.
8. Click **Save**.

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 option group within a bundle

1. On the **Manage Products** page, open the products details page of the bundle the option group is associated to.
2. Click **Structure** and select the option group you want to modify.
3. Fill in the required details as explained in the table below:

> ⓘ Option Groups can be of *Shared* or *Standalone* type. When you make any structural changes to the *Shared* option group such as removing an option, such changes are reflected in all the bundles where you have associated this option group. The changes you make to the *Standalone* type option groups are limited within the bundle.

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.

Field Name	Description
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.

To modify existing options inside a bundle

1. On the **Manage Products** page, open the products details page of the bundle the option is associated to.
2. Click **Structure** and select the option group which is associated with the options you want to modify. Options are displayed in the middle section.
3. Fill in the required details for an option as explained in the table below:

Field Name	Description
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	Enter a specific number of options to the bundle during configuration or you can also use Numeric Expressions to populate this field. If you define a numeric expression, a lock icon is displayed on the Configuration page next to the option quantity.
	
	ⓘ The default quantity of an option product defined here at the bundle structure takes precedence now over the default quantity of the option product defined at the price list item. The precedence for calculating the default quantity is as follows:
	   a. Transactional input provided by the sales representative on the Configuration page
	   b. Default quantity defined at the bundle structure level
	   c. Default quantity defined at the price list item level

Field Name	Description
Default	Select the check box to mark this option as Default. ⓘ CPQ overrides default option if the **Max Options** field is set to *1* and there is a rule defined to auto-include another option from the same option group.
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 checkbox 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. ⓘ Cloning is only allowed for option, bundle or sub-bundle with **Max Quantity** more than 1.
Required	Select the check box to mark this option as mandatory.
Auto Update Quantity	Select this check box to enable an option quantity to automatically update, when you have the option quantity being derived from a numeric expression. Selecting this checkbox, CPQ recalculates the new quantity of an option when something else has changed that would re-evaluate the expression itself. When you disable the checkbox, the quantity is updated only has the first time of option selection.

Field Name	Description
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.
Inclusion Criteria	Specify the minimum product options that a user should select on the catalog page.

4. You can define the fields mentioned in the table above for options individually or define for all the options at once using the edit icon (✏) next to **Action**.
5. Click **Save**.

Whenever you modify a bundle structure by adding or removing options or option groups, a message to execute the bundle maintenance batch job is displayed. If you are using Salesforce Classic, you must execute the bundle maintenance job. The bundle maintenance batch job is automatically executed in CPQ Admin. However, you must execute bundle maintenance batch job after mass changes. For more information, refer to Running a Bundle Maintenance Batch Job.

To lock an Option value

1. Define a numeric expression in the **Default Quantity** field. You can use the expression builder to create the expression.
2. When the Sales rep configures a bundle, on the Configuration page, a lock icon appears next to the option's quantity. The Sales rep 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.

Excluding Optional Product in Min/Max Criteria

You can exclude options, bundles, and sub-bundles that are marked optional from the min/max criteria validation of the option group. When marked as options using the **Is Optional** checkbox, options or bundles as optional are not considered in the **Min Options**, **Max Options**, **Min Total Quantity**, and **Max Total Quantity** criteria if this setting is enabled. This feature is useful for excluding the price and quantity of optional products from the total. To exclude optional products from min/max criteria, you need to enable the **Exclude Options Products** custom setting.

To enable optional product exclusion

1. In CPQ Admin, go to **Application Management** > **Config Page** > **Display Settings**.
2. Select **Exclude Optional Products**.
3. Click **Save**.
4. Execute the Custom Settings Maintenance batch job.

For more information on marking products are optional, refer to Configuring Products from the Catalog.

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

1. Go to **Setup** > **App Setup** > **Develop** > **Custom Settings**.
2. Click **Manage** for *Config System Properties*.
3. Click **Edit** for *System Properties*.
4. 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 a 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.

5. 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.

> ⓘ **Hyperlinks for Options Groups in the Summary Section**
> - A summary section displays Option Group and child Option Groups with hyperlinks that takes you to the corresponding section wherein the entire configuration is shown.
> - You must enable a custom setting Enable Hyperlink navigation in Bundle Summary to activate the hyperlinks. This setting is enabled by default.

> • In New Admin UI, this setting is available under Display settings of Bundle Page.

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 Managing Pricing.

Configuring Bundle Expansion on Cart Page

The Sales Rep can expand the bundles to view the options, sub-bundles, and secondary line items on the Cart page using the expand icon. CPQ provides a feature to prevent the Sales Rep from expanding the bundles. This is helpful if you do not want to expose the detailed prices and discounts applied at the option level of a bundle or sub-bundle to the Sales Rep.

You must create the **Disable Bundle Expansion** toggle button on the Product object. You can use this button to control Sales Rep's ability to expand the individual bundle or sub-bundle on the Cart page. When you enable this field for a bundle, the Sales Rep is not allowed to expand that bundle or sub-bundle on the Cart page as the expand icon is disabled. However, the Sales Rep can expand other bundles where this field is not enabled.

You can enable the **Disable Bundle Expansion** button on the Creating Product page in CPQ Admin UI. For more information on how to enable **Disable Bundle Expansion**, refer to Creating Products. You also need to add the field in **View Cart Custom Fields** in Config System Properties.

Perform the following tasks to define Bundle Expansion functionality.

To create the field **Disable Bundle Expansion**

1. Go to **Setup** > **App Setup** > **Customize** > **Products** > **Fields**.
2. In **Product Custom Fields & Relationships** section and click **New**.
3. Select **Checkbox** data type and click **Next**.
4. Enter the details:

Option	Description
Field Label	Enter *Disable Bundle Expansion* as the value of the field. You can change the label if required.
Default Value	Select **Unchecked**.
Field Name	Enter *APTS_DisableExpandInCart* as the value of the field. ⓘ You must only use the *APTS_DisableExpandInCart* API name in the **Field Name**. The feature is only enabled if you enter the exact name.

5. Click **Next**.
6. Select the profiles to which you want to give access to the field and click **Next**.
7. Ensure the field is added to the available layout.
8. Click **Save**.

To add field values in View Cart Custom Fields

1. Go to **Setup** > **Develop** > **Custom Settings** > **Config System Properties** > **Manage**.
2. Find **System Properties** and click **Edit**.
3. In the **View Cart Custom Fields** field, enter the following values separated by commas:
   - *Apttus_Config2__ProductId__r.APTS_DisableExpandInCart__c*
   - *Apttus_Config2__OptionId__r.APTS_DisableExpandInCart__c*
4. Click **Save**.
5. Execute Custom Setting Maintenance. Refer to Running Maintenance Jobs.

To execute Query **Field Metadata**

1. Go to All Tab (✚) > **CPQ Console** > **Maintenance Console**.
2. In the **Query Field Metadata** section, click **Refresh Field Metadata**.

To add **Disable Bundle Expansion** in Create Products page

You must add the **Disable Bundle Expansion** button to Create Product page in CPQ Admin. The button is not displayed otherwise.

Follow the instructions in Configuring the Product Creation Screen Fields to add the button.

Running a Bundle Maintenance Batch Job

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 maintenance of view record for individual bundles. It is necessary to run 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.

For more information, see Running a Bundle Maintenance Batch Job.

Distinguishing Cloned Options on the Bundle Configuration Page

On the bundle configuration page, under Itemized Options, CPQ provides a way to distinguish the cloned options from the other option products based on line item fields.

1. Go to **All tabs** > **Config Setting** > **Bundle Page Settings**.

2. Under Option List Settings, in the **Option Field In Summary** field, enter one the following options:

Value	Description
Blank	CPQ displays the name of options.
API Name	CPQ displays the name of options.
LineItem.<API name of the field>	CPQ displays option products with line item fields. For example, if you enter LineItem.Name, CPQ displays the line item ID on the bundle configuration page, under Itemized Options.
Option.<API name of the field>	CPQ displays the name of options.

3. Click **Save**.

Managing Product Groups

You can create a logical grouping of one or more product records using Product Group. This construct allows you to create combinations of products with similar characteristics/ qualities for use in a Price Rule. Product Group (as opposed to Product Family) can consist of products 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.

In this section:

- Creating Product Groups
- Associating Products to a Product Group

Creating Product Groups

To create a product group

1. On **App Menu**, click **Conga CPQ Admin** and then click the **CPQ Admin** tab. The new admin console is launched.

2. On the **Products** menu, click **Manage Product Groups**. Details first Product Group is displayed by default.
3. Click **New Product Group.**
4. In **Group Name**, enter a mandatory group name.
5. In **Description**, enter a brief description to describe the purpose of the group.
6. 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. You can view the details on the product group in the **Details** tab.

Associating Products to a Product Group

To Associate Products to Product Group

You must have an existing product group.

1. On **App Menu**, click **Conga CPQ Admin** and then click the **CPQ Admin** tab. The new admin console is launched.
2. On the **Products** menu, click **Manage Product Groups**. Details of the first Product Group is displayed by default.
3. Search and click the product group to open the product group details page.
4. Click **Associated Products**.
5. Drag and drop all the products you want to add in the Product Group from the **All Product** list. You can search for the products using the search bar.
6. Click **Save**.

All the products added to the product group are displayed.

To Associate Products to Product Group Within a Product

You must have an existing product group.

1. On **App Menu**, click **Conga CPQ Admin** and then click the **CPQ Admin** tab. The new admin console is launched.
2. On the **Products** menu, click **Manage Product**.
3. Search and click the product to open the product details page.

4. Click **Associations**. Click the Product Group icon(🔶). **Product Group** pane is displayed.
5. Drag and drop the desired product groups from the **Product Group** pane. You can search for the product groups using the search bar.
6. Click **Save**.

The product is now associated to the product group.

You can view all the products that you associated with the product group listed in the center pane under the **Associated Products** tab on the Manage Product Groups page. Select a product from the list on the left-hand side pane and the list of associated products is displayed in the center. You can also search for a product from the list of associated products using the search box at the top of the center pane.

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 with 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 on this page explain the following functionalities:

- Creating Product Attribute Fields
- Creating Product Attribute Groups
- Associating Attribute Groups to a Product
- Setting the Default Price from Attributes Using a Formula Field
- Cascading Shared Attribute Values
- Classifying Attributes
- Enabling Optimized Attribute Retrieval

Creating Product Attribute Fields

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 through attribute groups.

To Create a Product Attribute Field

1. Go to **Setup** > **App Setup** > **Create** > **Objects**.
2. On the Custom Objects page, scroll down and select the **Product Attribute Value** object.
3. Scroll down to the **Custom Fields & Relationships** related list, and click **New**. You will be directed to the standard Salesforce field creation wizard.
4. Use the wizard to 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 Product Attribute Groups

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.

Prerequisite

Select the **Has Attributes** checkbox. Clearing the **Has Attributes** checkbox disables the Configuration page.

To Create a Product Attribute Group

1. On **App Menu**, click **Conga CPQ Admin** and then click the **CPQ Admin** tab. The new admin console is launched.
2. On the **Products** menu, click **Manage Products**.
3. Search and click the product you want to associate attributes with.
4. Click **ATTRIBUTES**. The page displays all the attribute and attribute groups associated with the product.
5. Click **New Attribute Group** button. A pop-up is displayed.
6. Fill in the required details.

Field Name	Description
Group Name	Type a name for the Attribute Group
Two Column Attribute Display	Indicates that the configured attributes are displayed in two column format.
Three Column Attribute Display	Indicates that the configured attributes are displayed in three column format.
Description	Describe the attribute group.

7. Click **Save**.

Attribute Group are created and automatically associated with the product.

Associating Attribute Groups to a Product

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

Prerequisite

1. Select the **Has Attributes** check box. Clearing the **Has Attributes** check box disables the attributes page.
2. You must have an existing product attribute group with product attribute field associated to it.

To Associate an Attribute Group to a Product

1. On **App Menu**, click **Conga CPQ Admin** and then click the **CPQ Admin** tab. The new admin console is launched.
2. On the **Products** menu, click **Manage Products**.
3. Search and click the product you want to associate attributes with.
4. Click **ATTRIBUTES**. The page displays all the attribute and attribute groups associated with the product.
5. Click the shortcut icon (‹) under **New Attribute Group** button. **Attribute Groups** panel is displayed.
6. On the Attribute Groups panel, select *Attributes Group* from the dropdown menu.

7. Search the Attribute Group using the search bar and then drag and drop the desired Attribute Group in the middle.
8. Click **Save**.

An attribute group is associated with a product.

To Associate Attribute to Attribute Group

1. On **App Menu**, click **Conga CPQ Admin** and then click the **CPQ Admin** tab. The new admin console is launched.
2. On the **Products** menu, click **Manage Products**.
3. Search and click the product you want to associate attributes with.
4. Click **ATTRIBUTES**. The page displays all the attribute and attribute groups associated with the product.
5. Select an attribute group you want to associate attributes with. Details of that attribute group are displayed in the middle.
6. Click the shortcut icon (‹) under **New Attribute Group** button. Attribute Groups panel is displayed.
7. On the Attribute Groups panel, select *Attributes* from the dropdown menu.
8. Search the Attribute using the search bar and then drag and drop the desired Attributes in the Product Attributes section.
9. Fill in the required details.

Field Name	Description
Attribute	This is fields populated automatically and is not editable.
Expression	Define an expression to defined criteria.
Is Read Only	Indicates that the value of the attribute is not editable. The user cannot edit the value.
Is Hidden	Indicates that the attribute is not displayed on the configuration page. However, the attribute is associated with the attribute group.

Field Name	Description
Is Primary	Indicates that the attribute is critical and the attribute will be displayed on the Installed Products page as a critical attribute. ⓘ CPQ displays the critical or primary attributes associated with the main bundle product on the Installed Products page. However, CPQ does not display the critical or primary attributes associated with sub-bundles on the Installed Products page.

10. Click **Save**.

Setting the Default Price from Attributes 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.

Cascading Shared Attribute Values

You can enable cascading of attribute values from bundles to the associated options when you have defined shared attributes. Initially, the bundle attributes will be cascaded to options in the bundle, but subsequent changes in the bundle attribute will not be cascaded to option attributes. You can enable the cascading of attribute values at the global level or individual attribute level.

You can define the **Cascade Shared Attributes Updates** custom setting to enable cascading of attribute values globally. At the attribute level, you can enable cascading for a particular attribute after associating them to a Product Attribute Group. You can also enable cascading selectively for different actions of Asset-Based Ordering. When you define cascading at the attribute level, the **Cascade Shared Attributes Updates** custom setting is ignored for that attribute.

To enable cascading globally

1. Go to **Setup > Develop > Custom Settings > Config System Properties > Manage**.
2. Find **System Properties** and click **Edit**.
3. Enable **Cascade Shared Attributes Updates**.
4. Click **Save**.

To enable cascading at the attribute level

> ⓘ This functionality is only available in Salesforce Classic.

1. Go to All Tabs > **Product Attribute Groups** tab.
2. Click on **Attribute ID** from the Attributes related list. Product Attribute Details page is opened.
3. Click **Edit**.
4. Update the following fields:

Field	Description
Cascade Shared Attribute Updates	Select one of the following values: • *None:* Cascading is executed based on the global setting • *Yes*: Cascading is executed regardless of the global setting for the line item status selected in **Cascade Shared Attribute Status** • *No*: Cascading is not executed regardless of the global setting for the line item status selected in **Cascade Shared Attribute Status** > ⓘ If you select the values *Yes* or *No*, CPQ overrides the value of the global setting for that attribute. The global setting is only considered when **Cascade Shared Attribute Updates** is *None*.

Field	Description
Cascade Shared Attribute Status	Select one or more line item status from the following and add them to the right-hand side box labeled as **Chosen**. CPQ only cascades attribute values for the line item with line status that matches a value selected in this field. · *All* · *Renewed* · *Incremented* · *Incremented and Merged* · *Amended* · *Changed Configuration* · *Upgraded* · *Split* · *New* · *Existing* If you leave the **Chosen** box blank, based on the value of **Cascade Shared Attribute Updates**, cascading is executed for all line items regardless of line status.

5. Click **Save**.

Cascading is executed based on the combination of the **Cascade Shared Attribute Updates** and **Cascade Shared Attribute Status** fields. The following table explains the various combination of these fields:

Cascade shared attribute updates	Cascade shared attribute status	Description
Yes	Renewed, Upgraded	Cascading is only executed for renewed and upgraded line items.
Yes	Blank	Cascading is executed for all the line items with exception of line items with *Cancelled* and *Merged* line statuses.
No	Blank	Cascading is not executed for any line item.

Cascade shared attribute updates	Cascade shared attribute status	Description
Blank		Regardless of the values selected in **Cascade Shared Attribute Status**, cascading is executed based on the **Cascade Shared Attributes Updates** custom setting.
No	Renewed, Upgraded	Cascading is executed for all the line items with exception of line items with *Renewed* and *Upgraded* line statuses.

Classifying Attributes

CPQ classifies attributes as configuration attributes and price driving attributes when you define them. Attributes are marked as price driving when you define attribute-based pricing for any product. Otherwise, the attributes are considered as configuration attributes. Classifying the attributes enables CPQ to identify the attributes used in pricing computation and fetch only the required attributes while calculating prices. The classification of attributes also reduces the possibility of encountering governor limits while calculating asset pricing by enabling CPQ to only fetch the attribute relevant for calculation.

When you execute the Criteria Maintenance Batch job, the field **Attribute Type** in Product Attribute Member Type object of these attributes is automatically set to *Pricing*. You must manually update the **Attribute Type** field for attributes that are being used in in-flight proposals.

To manually update attribute type as pricing

1. Go to All Tabs > **Product Attribute** tab.
2. Click on **Attribute ID** from the Attribute Types related list. Click **Edit**
   Or click **New Product Attribute Type Member** to create a new entry. Product Attribute Type Member detail page is opened.
3. Update the following fields:

Fields	Description
Attribute	Search and select an attribute. If you are editing, the attribute is automatically selected.
Product	Select the product associated with the attribute for pricing
Attribute Type	Select *Pricing* to indicate that the attribute drives the price.

4. Click **Save**.

Enabling Optimized Attribute Retrieval

You can enable optimized retrieval of attributes on the cart page. When you enable optimized retrieval, CPQ only retrieves the attributes in the cart that are associated with the product added to the Cart. By default, all the attributes from Product Attribute Value, Product Attribute Value Ext, Product Attribute Value Ext 2, Product Attribute Value Ext 3, and custom Product Attribute Value Ext objects are loaded on the Cart page.

To enable optimized attribute retrieval

You must define the Admin Setting *APTS_LoadProductRelatedAttributesOnly* by following the steps below:

1. Go to **All Tabs** > **Admin**
2. Click **New**, to create a new record.
3. Fill the following details

Field	Value
Name	*APTS_LoadProductRelatedAttributesOnly*
Value	*true* or *false*
Code	Leave the field blank

4. Click **Save**.

Configuring 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 to ensure that the products having Red color are visible on the Catalog page.	Refine Your Search	Select **Red** check box in the color section in Refine your Search dialog box.
	Search Filters (CPQ)	Select the appropriate price list in the Inclusion Criteria; select *Color* from Field, *Equal to* from Operator, and *Red* from Value.
	Visibility Rules through Custom Classes. Refer to Product Filter Callback Class	Modify the custom callback class code to list the catalog products as ```\nglobal String\ngetProductFilterExpr(Cus\ntomClass.ActionParams\nparams) {\n return 'Color__c\n= \\'Red\\'';\n```

Scenario	Functionality	Configuration Required

You want to ensure that the Bundle products having Red color and relevant Option products having Green color are visible on the Catalog page.	Search Filters (CPQ) and Visibility Rules through Custom Classes. Refer to Product Filter Callback Class	Select the appropriate *Price List* in the Inclusion Criteria; select *Configuration Type* from Field, *Equal to* from Operator, and *Bundle* as Value.
		
		Modify the custom callback class code to list the catalog products as
		
		```\nglobal String\ngetProductFilterExpr(Cus\ntomClass.ActionParams\nparams) {\n return 'Color__c\n= \\'Red\\'';\n```
		
		Modify the custom callback class code to list the option products as
		
		```\nglobal String\ngetOptionFilterExpr(Cust\nomClass.ActionParams\nparams) {\n return\n'ComponentProductId__r.C\nolor__c =\\'Green\\'';\n```

Product Visibility can be controlled in the following functionality.

- About Refine Your Search
- About Search Filters (CPQ)

About Refine 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.

Use Case:

For example, if you select Asia and Europe from Regions drop-down menu, then products available in both these regions are shortlisted.

> ⓘ 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:

Configuring 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.
Configuring 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.
Configuring Refine Your 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.
Selecting Search Field Values 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.

Configuring 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 __c to it.
Description & Help Text	Standard Salesforce description and help text fields.

5. Click **Next**.
6. Select which profiles will have access to the field and click **Next**.
7. 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.
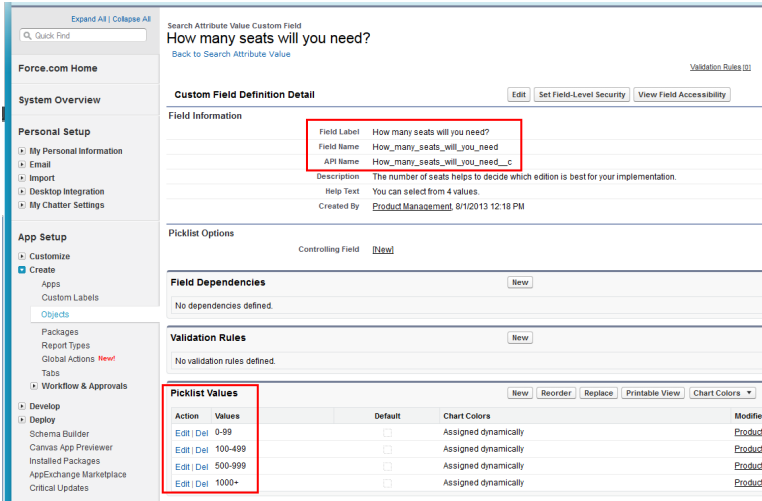
You must now complete the same task again, creating the same field label, values, and field name, but for Products. Go to App **Setup** > **Customize** > **Products** > **Fields** and then begin from Step 2.

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.
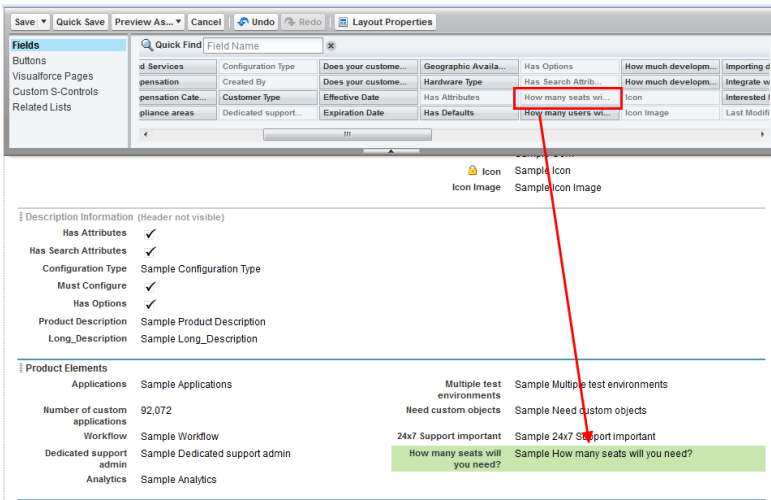
Configuring Search Filter Fields

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.

Configuring Refine Your 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 ▶.
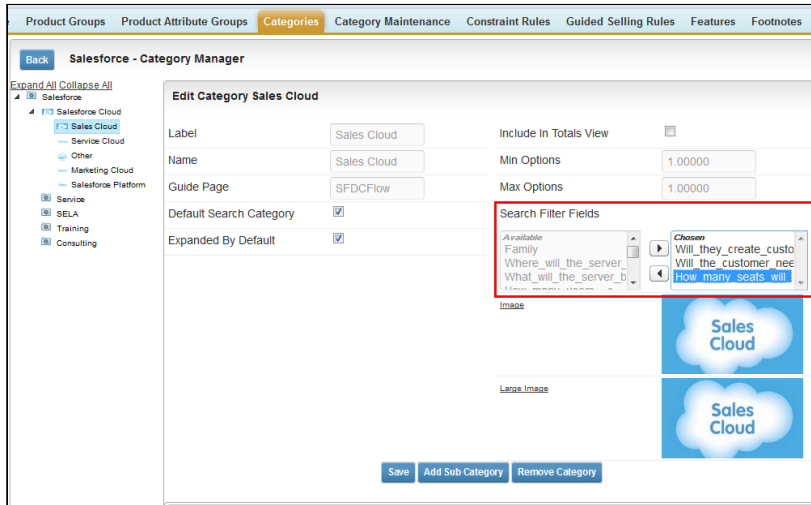
4. Click **Save**.

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.

Selecting Search Field Values for a Product

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 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.

Configuring 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.

About 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.

For example, your company has 2 types of accounts in a quote, *Personal* and *Business*, and the products are of types *Personal*, *Business* or *Both*. You can define filters to display only products that are valid for the type of account on the quote. If the account type is *Business*, only the products that are of type *Business* or *Both* are displayed. In this scenario, you need to create 2 Search Filters.

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.

You can also apply the visibility rules for the categories displayed on the Catalog page. This functionality 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 the Sales rep to search for the products and add them to the Cart.

> ⓘ Filters defined in Search Filter (CPQ) are not applied on the Service Catalog, when the Sales rep uses **Add/Remove** and **Add/Remove at all levels** buttons available on the Service Cart page to navigate to the Service Catalog.

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

In this section,

- Configuring Search Filter (CPQ)
- Creating Visibility Rules
- Use Case: Configuring Category Visibility

Configuring Search Filter (CPQ)

You must follow the steps below in Salesforce Classic.

To Configure Business Object Field

1. Go to **Setup** > **App Setup** > **Create** > **Objects** and select **Search Filter (CPQ)**.
2. Select **Business Object** under the **Custom Fields & Relationships** section.
3. From **Values** section, Remove *SearchAttributeValue__c* by clicking **Del**. This value must be deleted if it has not already been deleted.
4. Click **New** to add the following values in the picklist. Enter the API names of the objects you want to be able to use as a filter in separate lines. For example,
   a. To define visibility for products add the following values.
      - *Apttus_Config2__SearchAttributeValue__c*
      - *Asset*
      - *Apttus_Config2__AssetLineItem__c*
      - *Product2*
   b. To define visibility for categories add the following value.
      - *Apttus_Config2__ClassificationHierarchy__c*
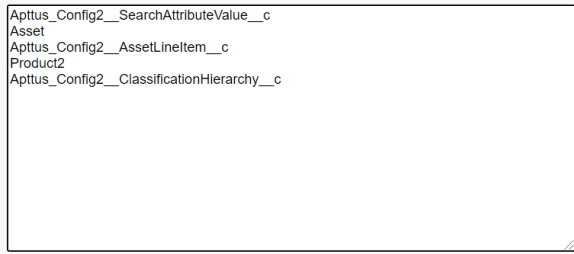
Add Picklist Values
Business Object

Add one or more picklist values below. Each value should be on its own line and it is used for both a value's label and API name.

If a value matches an inactive value's API name, that value is reactivated with its previous label.

If a value matches an inactive value's label but not the API name, a new value is created.

```
Apttus_Config2__SearchAttributeValue__c
Asset
Apttus_Config2__AssetLineItem__c
Product2
Apttus_Config2__ClassificationHierarchy__c
```

5. Click **Save**.

To Configure Filter Type Field

1. Go to **Setup** > **App Setup** > **Create** > **Objects** and select **Search Filter (CPQ)**.
2. Select **Filter Type** under the **Custom Fields & Relationships** section.
3. In the **Values** section, click **New** and enter the names you want to use to group the **Business Object** picklist values.
   a. For Products
      - *Asset*
      - *Product*
   b. For Category
      - *Category*

4. Click **Save**.
5. From **Field Dependencies** section, click **New**.
6. Select **Filter Type** from **Controlling Field** drop-down and select **Business Object** from **Dependent Field** drop-down.
7. Click **Continue**.
8. Select which business object fields should be associated with which filter. To select the Business Object value, double-click the value.

Filter Type:	Product	Asset
Business Object:	*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*
	Apttus_Config2__RollupData__c	*Apttus_Config2__RollupData__c*
	Apttus_Config2__OrderLineItem__c	*Apttus_Config2__OrderLineItem__c*
	Apttus_Config2__ExternalOrderSummaryItem__c	*Apttus_Config2__ExternalOrderSummaryItem__c*
	Apttus_Config2__Incentive__c	*Apttus_Config2__Incentive__c*
	Apttus_Config2__Milestone2__c	*Apttus_Config2__Milestone2__c*
	Apttus_Config2__MilestoneTemplate__c	*Apttus_Config2__MilestoneTemplate__c*
	Apttus_Config2__SearchAttributeValue__c	*Apttus_Config2__SearchAttributeValue__c*
	Apttus_Config2__ClassificationHierarchy__c	*Apttus_Config2__ClassificationHierarchy__c*

9. Click **Save**.

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

Creating Visibility Rules

Follow the steps below to create a visibility rule.

1. On **App Menu**, click **Conga CPQ Admin** and then click the **CPQ Admin** tab. The new admin console is launched.
2. On the **Catalog** menu, click **Manage Visibility Rules**.
3. Click **New Visibility Rule** button. **DETAILS** page is displayed.
4. Fill in the information as explained in the table below.

Field Name	Description
Name	Enter a name for the visibility rule.

Field Name	Description
Filter Type	Select the type of filter you want to define. You can filter the products or categories. This field determines the values in the **Business Object** dropdown. ⓘ You must use Salesforce Classic UI to define Product Attributes Value or Product Attributes Value Extension objects in Product Visibility Rules. For more information about Product Attributes Value Extension objects, refer to About Product Attribute Value Extensions.
Business Object	Select which object you to apply the filter on. This field controls the values available in the **Filter Criteria** field.
Value Object	Select which object you want to use to control the fields available for **Inclusion Criteria**.
Sequence	Enter a number to define a sequence in which the visibility rules are evaluated.
Description	Describe the visibility rule.
Active	Select the toggle button to activate the visibility rule.

Inclusion Criteria	This is fields is visible after you select **Value Object**. 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. You can use Advanced Options to define a more complex formula.
	
	Click expression icon(⬀). Click **Add New Criteria**. Fill in the following information to define an inclusion rule.
	
	   a. Row Num
	   b. Field
	   c. Operator
	   d. Value
	Click **Save**.

Field Name	Description

Filter Criteria	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.
	
	Click expression icon(⬈). Click **Add New Criteria**. Fill in the following information to define an inclusion rule.
	
	    a. Row Num
	    b. Field
	    c. Operator
	    d. Map To
	    e. Value
	Click **Save**.

5. Click **Save**.

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

Use Case: Configuring Category Visibility

To display the category whose name starts with Hardware

1. On **App Menu**, click **Conga CPQ Admin** and then click the **CPQ Admin** tab. The new admin console is launched.
2. On the **Catalog** menu, click **Manage Visibility Rules**.
3. Click **New Visibility Rule** button. **DETAILS** page is displayed.
4. Fill in the information as explained below.

Field Name	Description
Name	Enter a name for the visibility rule.
Filter Type	Select *Category*.
Business Object	Select *Category Hierarchy*.

Field Name	Description
Value Object	Select *Product Configuration* or *User* if you want to define inclusion criteria.
Sequence	Enter *1* as the sequence number.
Description	Describe the visibility rule.
Active	Select the toggle button to activate the visibility rule.
Filter Criteria	Click expression icon(⬀). Click **Add New Criteria**. Fill in the following information to define an inclusion rule. i. Enter *1* in the **Row Num**. ii. Select *Name* as a **Field.** iii. Select *starts with* as an **Operator.** iv. Specify desired category name as a **Value.** For example *Hardware*. Click **Save**.

5. Click **Save**.

CPQ displays only the categories that satisfy the criteria in the visibility rule on the Catalog page.

Configuring Product Comparison

You can enable a Sales rep 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.

Set up the Feature Set → Associate a feature set to a product → Define the Features of the product
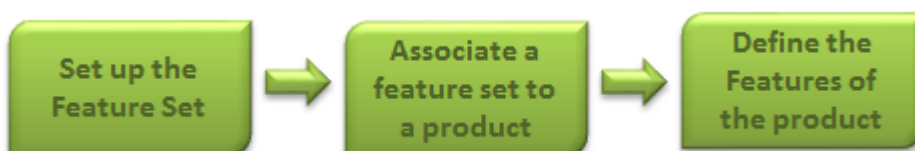
If you do not want to use the Compare Products Feature, you can hide the button from the Catalog page.

The following topics describe the instruction to configure the Product Comparison:

- [Setting Up Feature Sets](#)
- [Associating Feature Sets to a Product](#)
- [Defining Feature Values for a Product](#)
- [Hiding Products Comparison](#)

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 Feature Sets 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 the association and click **Add**.

The feature set is associated with a product.

Define values for the features you just created.

Defining 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 Products Comparison

There is a custom setting that enables you to show or hide the *Compare Products* button and the product compare checkbox 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.

Configuring Numeric Expressions

Numeric expression is the extended custom implementation of SFDC formula builder, which is fine tuned by Conga. 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 the criteria of product attribute rule (PAR)

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

Use Case:

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 watt-hour and battery capacity requirements. 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:

Summary of Usage			
Refrigerator Run Time (hrs per day)	8	Television Run Time (hrs per day)	4
Microwave Run Time (hrs per day)	0.50	House Lights (hrs per day)	3.00

Summary of WattHours required	
Total daily WattHours required	= Refrigerator run time * 140 + Television Run Time * 120 + Microware run time * 1000 + House Lights * 200
WattHours required for 3 days	= Total daily WattHours required * 3
Battery capacity (50% discharge)	= WattHours required for 3 days * 0.50

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
- Populating Option Group Min/Max Quantities Using Numeric Expression
- Populating Option Default and Min/Max Quantity Using Numeric Expression
- About Manage Field Expressions Page
- Running the Criteria Maintenance Job for Expression Fields
- Creating Numeric Roll-Up Summary Fields for Objects
- Examples and Related Syntax for Numeric Expressions

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 the text field next to the attribute in the Expression Column. An expression builder pop up is displayed.
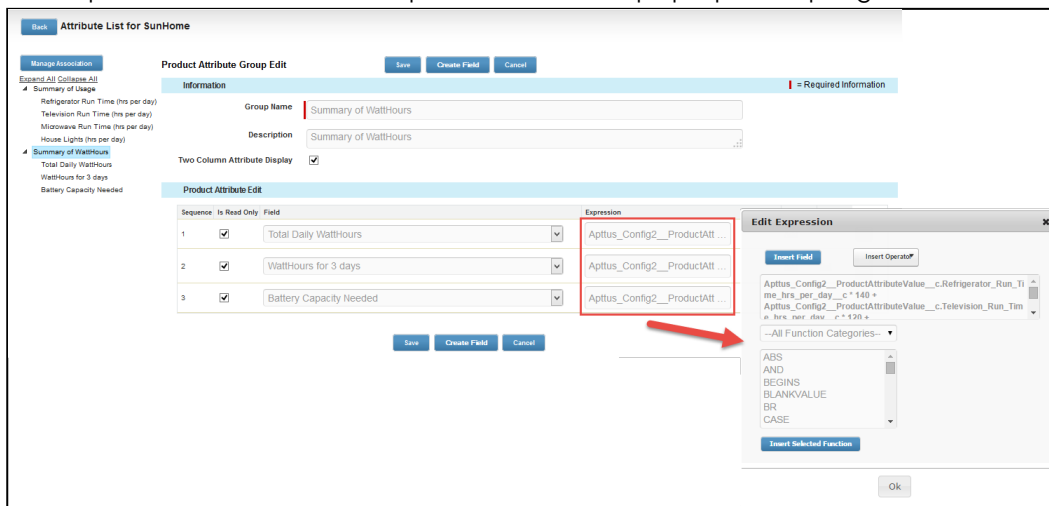
Selecting any function displays the example syntax on the right on how you can use

it.
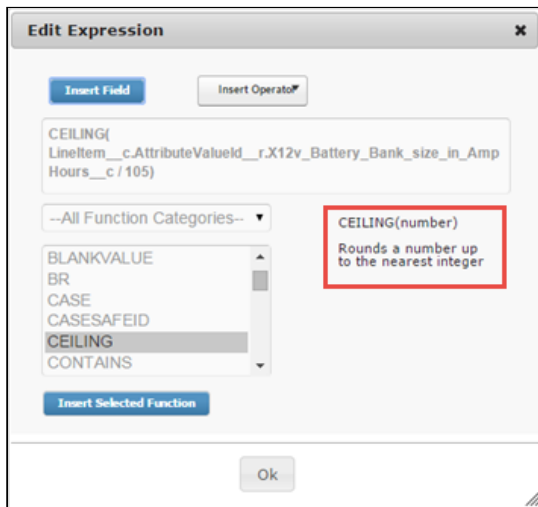
5. Click **Insert Field**. To retrieve the fields for building the expression, click Insert Field and choose Product attributes or Line Item. Line Item allows you to get access to fields at Product and Quote objects as well.
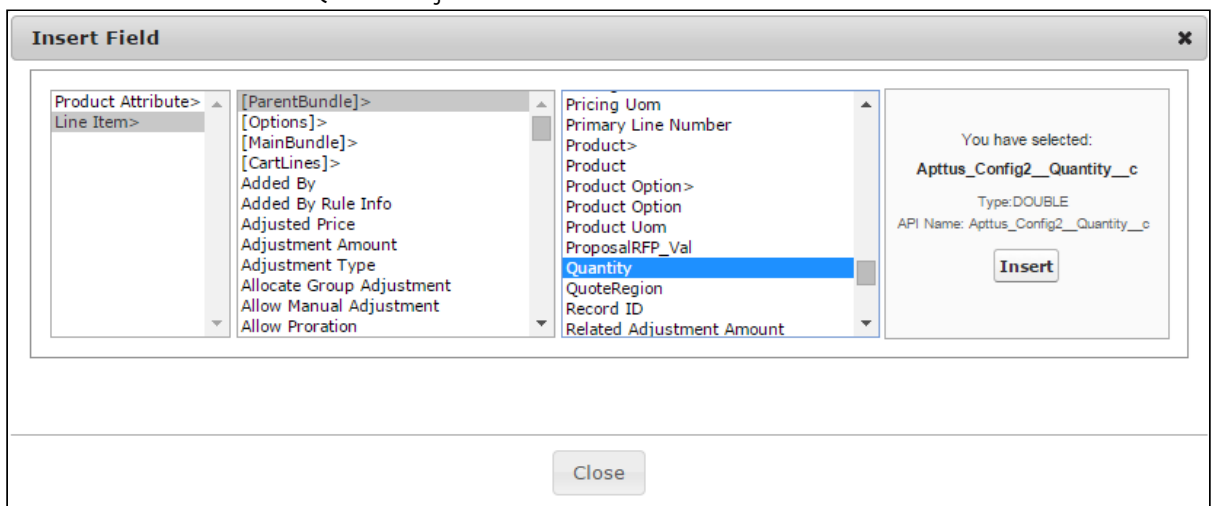
6. In the Insert Field dialog, click **Product Attribute** and select the required attributes to derive the values for attributes of SunHome. The following expressions are built for each attribute:

   - 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 Examples and Related Syntax for Numeric Expressions.

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 to Running the Criteria Maintenance Job for Expression Fields.

Populating Option Group Min/Max Quantities Using Numeric Expression

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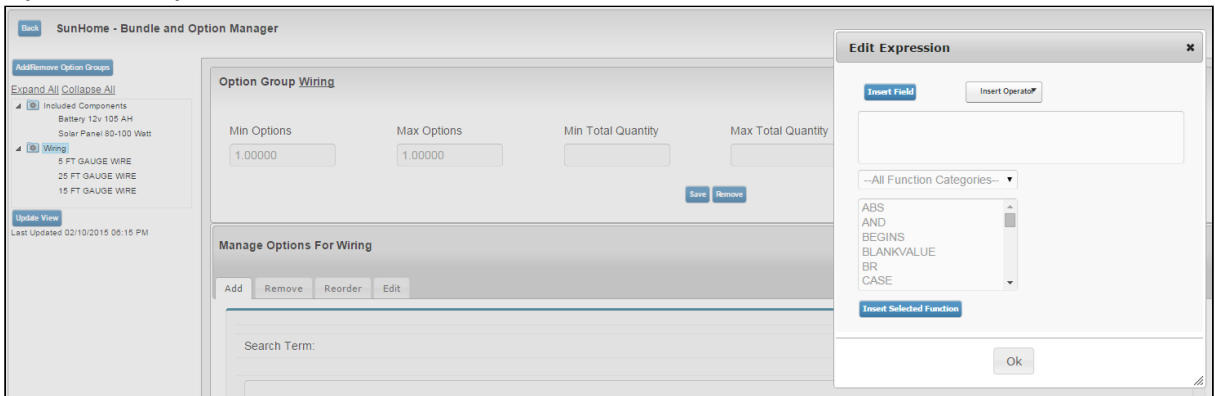
4. Select one or both of the following fields to add an expression.
   - Min Total Quantity
   - Max Total Quantity
   The expression builder opens.

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 Examples and Related Syntax for Numeric Expressions.

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 ☐

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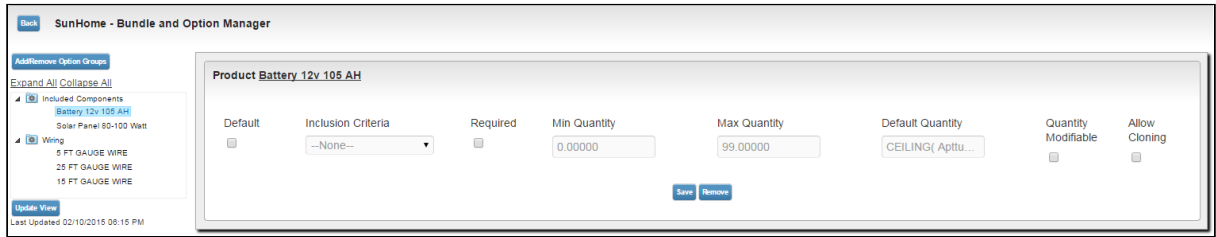
4. To add an expression, click one or more of the following:
   - Min Quantity
   - Max Quantity
   - Default Quantity

   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__Attribut eValueId__r.Battery_Capacity_Needed__c/(12 * 105))
   - Solar Panel:
     CEILING(Apttus_Config2__LineItem__c=>ParentBundle.Apttus_Config2__Attribut eValueId__r.Total_Daily_WattHours__c/(90 * 5))
     Total_daily_WattHours_required__c, 0) / (90 * 5))

   > ✅ For more common syntax examples, see Examples and Related Syntax for Numeric Expressions.

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 to Running the Criteria Maintenance Job for Expression Fields.

About Manage Field Expressions Page

The Manage Field Expressions page in the CPQ Admin user interface holds all the existing field expressions that you created. You can search, open, update, or create field expression from the Manage Field Expressions page. In CPQ Admin, Go to the **Products** tab and from the dropdown click **Manage Field Expressions**. A list of all the existing field expressions is displayed with the fields **Name**, **Active**, **Update Field** and **Is Modifiable**. There are various functionalities available for you to manage field expression.

Field Expression Details

You can search for a field expressions on the Manage Field Expressions page using the **Search Field Expressions** bar. You must search using any keyword that the field expressions name contains. Click the name of the field expression to view the related information on the Details page. The fields like **Name**, **Update Object**, **Update Field**, **Value Expression** and, **Operator** are displayed on the Details page. You can update the field expressions on this page as well.

New Field Expressions

You can create new field expressions using the **New Field Expression** button on the Manage Field Expressions page and Details page of the Field Expressions.

To create new field expressions

Follow the steps below to create a new field expression.

1. Click **New Field Expression**. The Details page is displayed.
2. Fill in the following information in the fields displayed on the Details page. Because you are creating field expressions from the Manage Field Expressions page, only the values relevant to field expressions are displayed in the dropdown of the fields listed below.

Field	Description
Name	Enter the name of the field expression
Active	Toggle the button to the right-hand side to activate the field expression

Field	Description
Is Modifiable	Toggle the button to the right-hand side to enable editing of the field expression
Expression Type	This field is populated by default with the value *Field update*.
Expression Context	Select the value *Record Update* from the dropdown.
Source Object	Select the value *Line Item* from the dropdown.
Scope	Select one of the following values that are displayed in the dropdown: • *Header Level*: to apply field expression on fields at header level • *Product*: to apply field expression on the product line items • *Product Group*: to apply field expression on the product group level.
Filter Expression	Enter the filter expression. Use the expression builder to create a formula.
Update Object	Select one of the following objects that are displayed in the dropdown: • *Line Item* • *Product Attribute*
Update Field	Select the field on which you want to apply the field expression.
Operator	Select an operator from the dropdown.
Value Expression	Enter the value expression. Use the expression builder to create a formula.

3. Click **Save**.

4. Click the menu icon (⋮) next to the **New Field Expression** button. Click **Update Field Expression Criteria Fields** to execute the batch job.
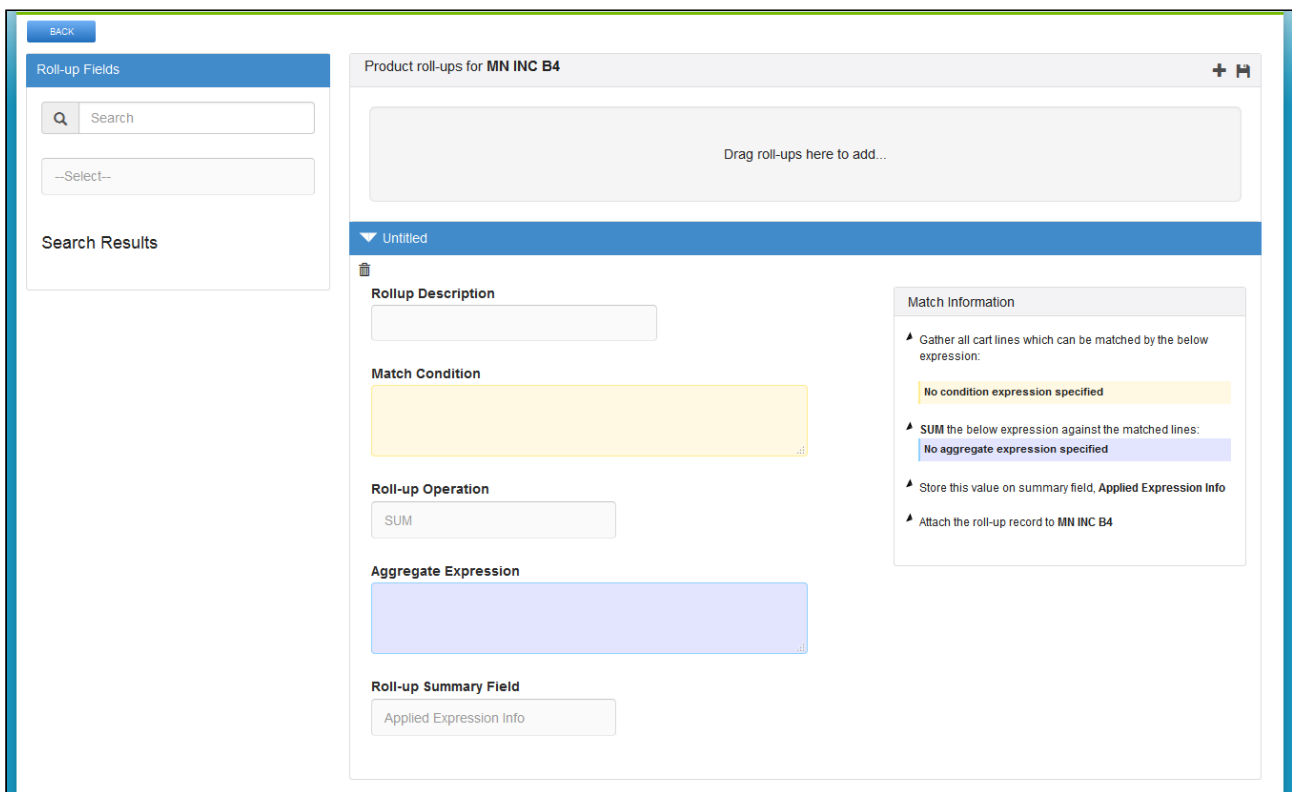
Running the Criteria Maintenance Job for Expression Fields

You must have existing field expressions.

1. On **App Menu**, click **Conga CPQ Admin** and then click the **CPQ Admin** tab. The new admin console is launched.
2. Click the menu icon( ) on the main menu bar at the top.
3. Click **Criteria Maintenance**. The Criteria Maintenance popup is displayed.
4. Click the **Update Expression Fields** button. This might take some time to update all expression criteria fields.
5. Close the Criteria Maintenance popup after the update is complete.

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.

ⓘ You can model your numeric expressions at the configuration level so that can rollup custom fields at the deal level. Rollups on configuration can be set through Apttus_Config2_*LineItem*_c.$Config.$Rollups (but does not return any value in client mode).

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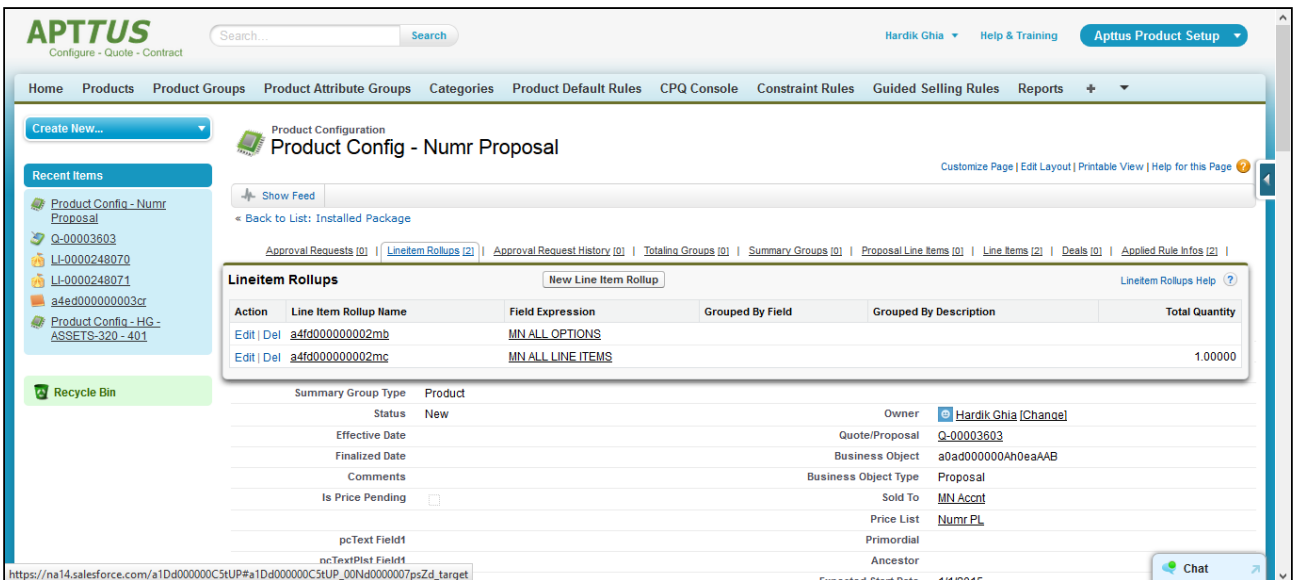
After you create rollup fields for a product, you can re-use the rollups across products or within the same product by searching for the rollup, product, or product group name in the Search field. Drag and drop the requisite roll-up conditions that you want to use to the right greyed out pane.
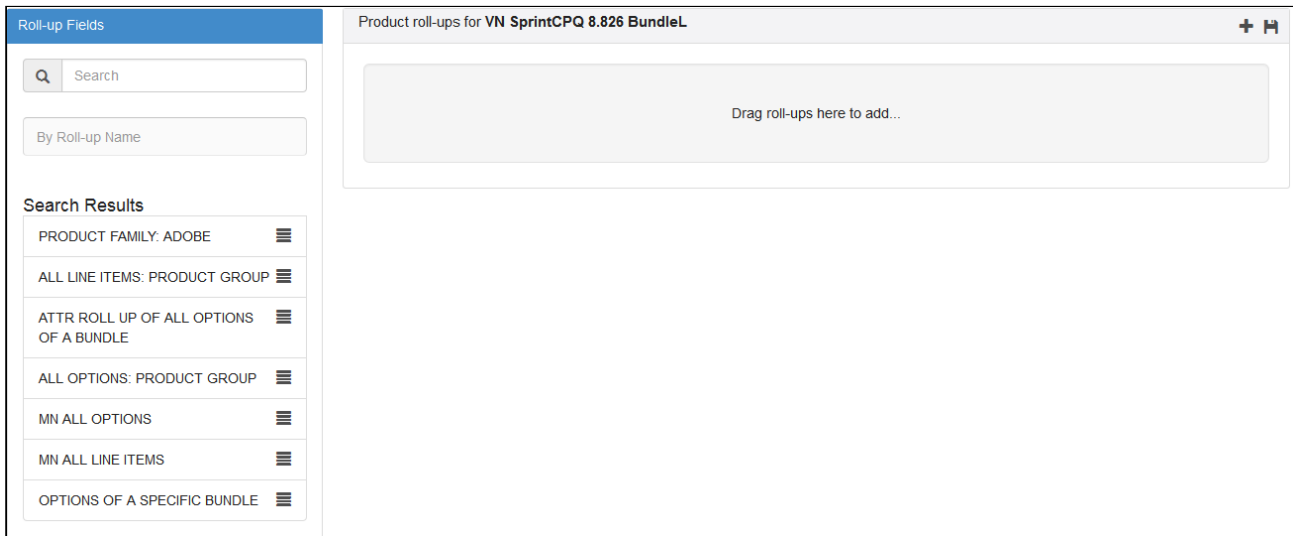
About Manage Rollups Page

The Manage Rollups page in the CPQ Admin user interface holds all the existing rollup type numeric expressions that you created. You can search, open, update, or create rollups from the Manage Rollups page. In CPQ Admin, Go to the **Products** tab and from the dropdown click **Manage Rollups**. A list of all the existing rollups is displayed with the fields **Rollup Name**, **Active**, **Update Field** and **Is Modifiable**. There are various functionalities available for you to manage rollups.

Rollup Details

You can search for a rollup on the Manage Rollups page using the **Search Rollups** bar. You must search using any keyword that the rollup name contains. Click the rollup name to view the related information on the Details page. The fields like **Name**, **Update Object**, **Update Field**, **Value Expression** and, **Operator** are displayed on the Details page. You can update the rollups on this page as well.

New Rollups

You can create new rollups using the **New Rollup** button on the Manage Rollups page and Details page of the rollups.

To create new rollups

Follow the steps below to create new rollups.

1. Click **New Rollup**. The Details page is displayed.

2. Fill in the following information in the fields displayed on the Details page. Because you are creating rollups from the Manage Rollups page, only the values relevant to rollups are displayed in the dropdown of the fields listed below.

Field	Description
Name	Enter the name of the rollup
Active	Toggle the button to the right-hand side to activate the rollup
Is Modifiable	Toggle the button to the right-hand side to enable editing of the rollup
Expression Type	This field is populated by default with the value *Rollup*.
Expression Context	Select the value *Rollup* from the dropdown.
Source Object	Select the value *Line Item* from the dropdown.
Scope	Select one of the following values from the dropdown: • *Header Level*: to display rollups on all products that have rollup fields and are added to the cart • *Product*: to display rollup on specific products that have rollup fields. Add the names of the products. • *Product Group*: to display rollup on the products added to the cart, that are part of a product group and have rollup fields defined. Add the name of the product group.
Filter Expression	Enter the filter expression. Use the expression builder to create a formula.
Update Object	This field is populated by default with the value *Line Item Rollup*
Update Field	Select the field in which you want to display the rollup.
Operator	Select an operator from the dropdown.

Field	Description
Value Expression	Enter the value expression. Use the expression builder to create a formula.

3. Click **Save**.

4. Click the menu icon (⋮) next to the **New Rollup** button. Click **Update Field Expression Criteria Fields** to execute the batch job.

About Rollup Scope

The scope of rollup defines the level at which the CPQ displays the rollups. The following scope is available to you:

Scope	Description
Header Level	The rollups are displayed on all line items with rollups that are added to the cart.
Product	The rollups are displayed on the product defined in this scope.
Product Group	The rollups are displayed on the products that are part of a product group defined in the scope.

The following is an example describes a of product configuration, rollup setup, and the output.

Products Configuration

Bundle 1	Bundle 2	Bundle 3
product scope attribute The attribute has a numerical expression to display the value of the rollup field "Product Rollup"	product scope attribute This attribute has a numerical expression to display the value of the rollup field "Product Rollup"	product scope attribute This attribute has a numerical expression to display the value of the rollup field "Product Rollup"

Bundle 1	Bundle 2	Bundle 3
product group scope attribute This attribute has a numerical expression to display the value of the rollup field "Product Group Rollup"	product group scope attribute This attribute has a numerical expression to display the value of the rollup field "Product Group Rollup"	product group scope attribute This attribute has a numerical expression to display the value of the rollup field "Product Group Rollup"
header level scope attribute This attribute has a numerical expression to display the value of the rollup field "Header Level Rollup"	header level scope attribute This attribute has a numerical expression to display the value of the rollup field "Header Level Rollup"	header level scope attribute This attribute has a numerical expression to display the value of the rollup field "Header Level Rollup"

Product Groups

Product Groups	Products
Product Group A	Bundle 1
	Bundle 2

Rollups defined

Rollup Scope	Description
Product	**Product** = *Bundle 1*
	Expression: the logic to calculate the number of products added to the cart.
Product Group	**Product Group**= *Product Group A,*
	Expression: the logic to calculate the number of products added to the cart.
Header Level	Expression: the logic to calculate the number of products added to the cart.

Output

The following table describes the output of each rollup based on the above-mentioned product configuration. When you add **Bundle 1**, **Bundle 2**, and **Bundle 3** to the cart, the expression result is *3* as there are three line items added to the cart.

Rollup Scope	Output
Product	*3* is displayed only on **Bundle 1**.
Product Group	*3* is displayed on **Bundle 1** and **Bundle 2**.
Header Level	*3* is displayed on **Bundle 1**, **Bundle 2**, and **Bundle 3**.

Examples and Related Syntax for Numeric Expressions

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__ProductId__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_Config2__OptionId__r.Name','Solar Panel 80-100 Watt'][0].Apttus_Config
  2__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_QPCo
nfig__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__Proposal
d__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 upsummary 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 upsummary 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_Config2__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_Config2__ProductId__r.Name','Logitech Keyboard
K120'].Apttus_Config2__Quantity__c,0) +
BLANKVALUE(Apttus_Config2__LineItem__c.
$CartLines['Apttus_Config2__ProductId__r.Name','Logitech Wireless
Keyboard'].Apttus_Config2__Quantity__c,0)
```

Managing Product Rules

You can use Product Rules to regulate the selection of the products by the Sales rep on the Catalog page. When the Catalog exceeds a simple number of products, it can be difficult for the Sales rep to find their desired product based on the specific requirements of the company. Product Rules can guide the Sales rep to follow what is best and compliant for the company.

For example, your company sells a range of products and there are certain regulations you must follow to sell the products, for example, there are products that cannot be sold together, products that cannot be sold in certain regions, or products that must be sold along with another product. You can define rules for such scenarios to help the Sales rep avoid selling a partial product or sell products they are not supposed to.

Product Rules consist of a set of conditions and actions. The rules are invoked when the conditions are satisfied and based on the definitions of the rules, actions are executed on the Catalog page. You can manage them from the Manage Rules tab in CPQ Admin.

You can configure the following types of rules:

- Constraint Rules: Configuration rules used to conditionally include, exclude, recommend, replace, or validate a product or set of products based on other products or set of products added to the cart.
- Attribute-based Configuration: Helps you modify the product configuration on your cart based on the selection of product attributes using Product Attribute Rules and Attribute Value Matix.

About Manage Rules Page

The Manage Rules page in the CPQ Admin user interface holds all the details of the existing constraint rules, product attribute rules, and attribute values matrices that you created. You can see them listed in the left-hand side pane. You can search, open, update, or create different rules from the Manage Rules page. In CPQ Admin, Go to the **Products** tab and from the dropdown click **Manage Rules**. There are various functionalities available for you to manage rules.

Rule Details

You can search for a rule on the Manage Rules page using the **Search** bar. You must search using any keyword that the rule name contains. You can filter out the different types of rules listed on the page. Click different icons that represent different rules to filter them.

- Constraint Rule icon ( )
- Product Attribute Rule icon( )
- Attribute Value Matrix icon( )

Click the rule name to view the related information in the center pane. The fields **Rule Name**, **Sequence, Effective Date**, **Bundle Context**, **Active** and, **Description** are displayed on the center pane. You can update the rules on this page as well.

New Rules

You can create new rules using the **New Constraint Rules**, **New Product Attribute Rule** and **New Attribute Value Matrix** buttons on the Manage Rules page. The below table mentions the list of topics you can refer to create different rules.

Rules	Topic
Constraint Rules	Creating Constraint Rules
Product Attribute Rules	Creating Product Attribute Rules
Attribute Value Matrix	Creating Attribute Value Matrices

ⓘ You can start following the instructions from Step 5 in the abovementioned topics to create the rules.

Managing Constraint Rules

Constraint Rules are configuration rules used to conditionally include, exclude, recommend, replace, or validate a product or set of products based on other products or set of products added to the cart. For example, when Product A is added to the cart, Product B can be automatically added based on an auto-inclusion type rule. Similarly, when Product B is added to the cart, the sales rep can be prevented from adding Product C. Constraint rules are applied to a product during the product configuration process, either on the client-side or the server-side.

The constraint rule consists of three associated Objects:

- Constraint Rule Object: Header level object that links the condition with the action. There are only three significant properties in this object, the active flag, action

association, and condition association. To configure a constraint rule, you must first create a constraint rule header.

- Constraint Rule Condition: Captures the condition that triggers the rule.
- Constraint Rule Action: Captures the rule action that is applied to the cart when the rule condition is satisfied.

CPQ supports five types of constraint rules. Each of the rules has support for their action intents on either server-side, client-side, or both.

Rule Type	Definition	Example	Server-Side Supported Functionalities	Client-Side Supported Functionalities
Inclusion	Used to add a product to the cart.	If the Sales rep has selected a laptop, then automatically include a laptop charger.	• Auto-inclusion • Prompt • Message	• Auto-inclusion • Prompt • Message
Exclusion	Used to exclude or prevent the addition of a product to the cart.	If the Sales rep has selected a region where a certain model of laptop is not available, then disable that laptop model.	• Prompt • Hide • Disable • Message	• Prompt • Hide • Disable • Message

Rule Type	Definition	Example	Server-Side Supported Functionalities	Client-Side Supported Functionalities
Replacement	Used to replace and add a product to the cart. ⓘ CPQ enables replacement rules only on the ABO carts, for swap and split-swap functionalities.	If the Sales rep has selected a product that is deprecated or not available then replace that product with another product that is available in the catalog.	• Prompt • Message	
Recommendation	Used to prompt product suggestions to the user for addition to the cart.	If the Sales rep has selected a laptop and has not selected a mouse, then recommend the user to select a mouse.	• Prompt • Message	• Prompt • Message
Validation	Used to prevent the user from finalizing the cart without resolving validation errors.	If the Sales rep chose a laptop and their limit was to buy only 5 units at a time, then validate the quantity.	• Message	• Message (Except for Product Groups)

Constraint rules are optimized with the client-side caching mechanism to improve performance and user experience. You can choose to process all constraint rules at the client-side or server-side, by defining the setting **Constraint Rule Execution Mode**, in *Config System Properties*.

> ⓘ The constraint rules are processed at the server-side by default. However, always use Client-Side Constraint Rules in a Service CPQ flow.

Note the following limitations when enabling Client-Side Constraint Rules (CSCR).

- Replacement type constraint rules are not supported.
- Product Scope, **Option Groups**, is not supported. If such a constraint rule is defined an error is displayed on the Catalog page.
- When you configure a constraint rule with **Action Type** = **Exclusion** and **Min/Max Match Rule** = **Exclude After One**, if the sales representative adds the product for the first time, CPQ displays a constraint rule message on the cart in the "Client" mode.
- **Match in Assets** in Match Conditions for Constraint Rule Actions is only supported for Inclusion type Client-Side Constraint Rule for Prompt, Auto-include, and Show Messages actions. For Exclusion type Client-Side Constraint Rule for Prompt, and Show Message actions.

> ⓘ Constraint rules are not executed on line items with the status as *Cancelled*. To avoid errors, add a criterion to exclude line items with line status as *Cancelled* in all such constraint rules.

You can also execute a constraint rule across the following bundle structures:

- From parent bundle structure to nested bundle structure
- From nested bundle structure to parent bundle structure
- Across nested bundles

You must follow the best practices while defining Constraint Rules. For more information, see CPQ Best Practices: Governor Limits.

Subsequent sections explain the following topics:

- Creating Constraint Rules
- Creating Constraint Rule Conditions
- Creating Constraint Rule Actions
- Translating Custom Messages for Constraint Rules
- Running the Constraint Rule Maintenance Job
- Configuring Inclusion Rules
- Configuring Exclusion Rules

- Configuring Validation Rules
- Configuring Product Option Group in Product Scope
- Scenarios for Using Constraint Rules
- Disabling Constraint Rule Execution upon Cart Finalization

Creating Constraint Rules

There are two different methods you can use to create Constraint Rules. You can create new constraint rules from the beginning or clone an existing constraint rule. You can set up to 50000 constraint rules with maximum of 10000 products. For more information, see Guidelines for CPQ. Follow the steps below to create constraint rules.

To create constraint rules

1. On **App Menu**, click **Conga CPQ Admin** and then click the **CPQ Admin** tab. The new admin console is launched.
2. On the **Products** menu, click **Manage Products**.
3. Search and click any product for which you want to define a constraint rule.
4. Click **Rules**. The page displays different rules tab.
5. Click Constraint Rule icon (🔗) next to the title, **RULES**, to launch the constraint rules listed for the product.
6. Click **New Constraint Rule** button.
7. Fill in the required details.

Field Name	Description
Rule Name	Enter a name for the constraint rule.
Is Active	Flags whether the Attribute Value Matrix is active.
Bundle Context	Indicated whether the constraint is evaluated within a bundle.
Sequence	Enter the number in which you want to sequence the constraint rule.
Effective Date	Select the date from which the constraint rule is in effect.

Field Name	Description
Expiration Date	Select the date until which the constraint rule is valid.
Short Description	Describe the constraint rule.

8. Click **Save**.

The constraint rule is created.

You can also clone an existing constraint rule.

To clone a constraint rule

1. Open an existing constraint rule.

2. Click the menu icon (⋮) under the **Rule Name** column.
3. Click **Clone Rule**.
4. Enter a name for the new constraint rule.
5. Click **Save**.

The constraint rule is cloned.

> ⓘ If a constraint rule is deleted or deactivated, then, if it was used in an existing quote, the changes made by the rules are reverted and the user can finalize and proceed with their quote without being blocked. CPQ does not show any error while you modify the constraint rules and re-configure the products for the existing quotes.

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. A constraint rule condition is met when the Sale rep adds a product that is part of the product scope you defined in constraint rule condition, to the cart. When the constraint rule condition is satisfied, the constraint rule action is executed on the cart. 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. Product scopes are groups in which the product belongs or a set of fields you can use as conditions. The product scope could be specific to a Product, Product Family, Product Group, Product Option Group, and a Product Field Set. Product scops are explained in detail later in this section.

You can define additional condition criteria like Country Codes or Color. The additional criteria on the condition is based on the line item object. Condition Criteria filter conditions that are applied to the line item created from the selected products that are within the product scope definition.

When criteria is defined, the condition is met when the product scope specification and criteria are both met. For example, based on your company's requirement, you cannot sell Product B if you are selling more than 3 units of Product A. In this case, you can define condition criteria to validate the quantity of Product A.

When using a constraint rule with 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 criteria. Even if you remove a constraint rule that has criteria defined, you must run a criteria maintenance.

> ⓘ   You can create a Constraint Rule Condition or Constraint Rule Action criteria that is based on a field from the Product Attribute Value Extension object.

To Create Constraint Rule Conditions

You must have an existing constraint rule header.

1. On **App Menu**, click **Conga CPQ Admin** and then click the **CPQ Admin** tab. The new admin console is launched.
2. On the **Products** menu, click **Manage Products**.
3. Search and click any product for which you want to define a constraint rule.
4. Click **Rules**. The page displays different rules tab.
5. Click Constraint Rule icon (🔗) next to the title, **RULES**. All the constraint rules associated with that product are displayed.
6. Search and click the constraint rule.
7. Fill in the required details under the **IF(scope)...** section. Enter a number under **Condition** column.
8. Select one of the following options from **Product Scope** drop down. The last column changes based on the product scope you select. Enter the name in the last column accordingly.

Option	Description
Product	After selecting, enter the name or product code of the 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 name 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 Sets in the product scope, an add icon (+) is displayed, add product fields. This implies that when the value of the field specified in the product scope matches the value of the field on the cart, the constraint rule condition is satisfied. You must add at least one Product Field. a. Click the add icon (+). b. Select a product field from the picklist. c. Select an operator. d. Define the value of the product field in **Value**. Each of the Product Field has one picklist value from the product object. ⓘ 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 on Product Field Set based constraint rules. CPQ considers **Match Rule** as **Match Any in Group** by default in such case.

Option	Description
Product Option Group	After selecting this option, enter an **Option Group ID** associated with the bundle in the **Product Option Group** field. This implies that one or more products from the specified product option group must be present in the shopping cart. The number of products that must be present depends on the Match Rule. ⚠ If you define the constraint rule in CPQ Admin, enter the option group name associated with the bundle in the **Product Option Group** field.
Header Level	This functionality is deprecated.

9. Click the setting icon (⚙). A pop-up is displayed. Click **CONDITIONS** tab and fill in the details as explained below.

Option	Description
Condition Name	The name is populated automatically. You can edit the name as you desire.
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 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 Cart Options	Selecting this means the condition is met only when the condition products are among the cart line items of the shopping cart.

Option	Description
Match in Location	Selecting this means that constraint rules are executed within a location selected on the catalog. When the Sales rep selects a location and adds the condition product to the cart, constraint rule actions are executed. Upon changing the location, constraint rules are invoked again if conditions are met for the newly selected location.
Match in Related Lines	Selecting this means the condition is met only when the condition products are among the related line items (bundle or standalone) on the shopping cart.
Match in Service Assets	Selecting this means the condition is met only when the condition product is a selected asset. An asset remains a selected asset while the Sales Rep is still on the Service Catalog. The selected asset considered a related line item if the Sales rep navigates to the Service Configuration, Service Cart after relating a service, or Installed Products page and returns to the Service Catalog. Follow the below practices while defining **Match in Service Assets:** • to exclude or include options from a bundle on the Configuration page, use **Match in Related Lines**. • to exclude services on the Service Catalog, use **Match in Service Assets**. • to exclude and include both services from Service Catalog and option in service bundle from the Configuration page, use **Match in Service Assets** and **Match in Related Lines** both. • Conga recommends that for the Inclusion type constraint rule, you must select **Match in Service Assets** and **Match in Related Lines** both for the functionality to work as expected. ⓘ CPQ creates related line items for service product added through auto-inclusion, which are deleted if you remove the auto-included service product

Option	Description
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 the same as Include Min/Max with Match Min Products as 1. It is also applicable to the 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.
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 Max Products	This field is applicable only when you select Min/Max from the Match Rule picklist. The condition is met, when the max 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.

10. Click **Save**.

11. Click the flare icon ( ) to define condition criteria. A pop-up is displayed.
12. Click **New Filter**.
13. Type a **Field**, an **Operator**, and type a **Value**. You can enter more than one criteria.
14. Click **Save**.

You can add multiple conditions in a constraint rule. Click add icon (+) and follow the steps again.

You can also define condition association to define additional criteria to further filter the product scope condition. Next to **IF(scope)...**, select *Custom* from the drop-down menu. When there are more than one condition for a constraint rule, this field defines when to consider one or more of the conditions using AND/OR association. The sequence number defined in the **Condition** column is used as an identifier of the condition.

Few examples, you defined conditions with sequence numbers as *1*, *2*, *3*, *4*, *5*, and *6*. The following examples describe various condition association

Condition Association	Description
(1 AND (2 OR 3))	Either Conditions 1 and 2 or Conditions 1 and 3 must be satisfied on the cart
1 AND NOT 2	Condition 1 is in the cart while Condition 2 is not
1 AND NOT(2)	Condition 1 is in the cart while Condition 2 is not
1 AND 1	Only Condition 1 is evaluated on the cart. If you have defined multiple conditions, CPQ does not evaluate the remaining conditions
NOT(5 OR 6)	Neither Condition 5 or 6 are in the cart
NOT 5 AND 6	Condition 5 is not in the cart, but Condition 6 is in the Cart.

It is recommended that you use logical operator - AND - to display the resultant record if all the conditions separated by AND is TRUE. For instance, (1 OR 2) AND (3 OR 4) AND 5. If you enter an invalid condition association, an error is displayed when you save the constraint rule.

> ⓘ 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 conditions are created. 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. The recommendation rules support Product, Product Family, and Product Group.
- The replacement rule helps you swap a product with another product by displaying a message with products that can be replaced. This is useful when an organization decides to discontinue a product and prompts you to go for a replacement.

> ⓘ CPQ supports replacement rules on ABO carts only for Swap and Split-Swap.

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.

> ✅ You should be careful not to put too many actions into a single constraint rule. You must follow the best practices while defining Constraint Rules. Refer to *CPQ on Salesforce Best Practices: Governor Limits*. You should test the constraint rules thoroughly before pushing them to a live environment.

For Validation type constraint rules, you can set up action criteria by defining product scope as Action Criteria. Action criteria validate the condition against the criteria you defined and executes the action intent. For example, based on your company's requirement, you cannot sell more than 3 units of Product A at a time. In this case, for condition as Product A, you can define action criteria to validate if the quantity of Product A is more than 3. If the quantity is more than 3, CPQ executes the action intent which could be an error or a warning.

When the criteria conditions are met, based on the action intent the rule action message is displayed either instantly or upon finalization. You can use the line item fields and related fields of the line item object, such as the attribute value field, product configuration fields, and product fields can also be used. When you use a constraint rule with action criteria for

a product with multiple line items, the system only validates the primary line item to evaluate the constraint rule.

If the rule action is error type, the Sale rep cannot finalize the configuration until the criteria condition values have changed to make the validation ineffective.

You must run a criteria maintenance job whenever you add, remove or make any changes to criteria. Even if you remove a constraint rule that has 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. On **App Menu**, click **Conga CPQ Admin** and then click the **CPQ Admin** tab. The new admin console is launched.
2. On the **Products** menu, click **Manage Products**.
3. Search and click any product for which you want to define a constraint rule.
4. Click **Rules**. The page displays different rules tab.
5. Click Constraint Rule icon ( ) next to the title, **RULES**. All the constraint rules associated with that product are displayed.
6. Search and click the constraint rule.
7. Fill in the required details under the **THEN(statements)...** section. Enter a number under **Action** column.
8. From **Action Type**, select one of the following:

Option	Description
Exclusion	Enforces exclusion of certain products from the shopping cart.
Inclusion	Enforces selection of certain products as specified by the product scope specification or criteria specification in the inclusion rule.
Recommendation	Product recommendations are displayed when the condition is met.
Replacement	Product replacements are suggested when the condition is met.

Option	Description
Validation	Displays the error message when the criteria conditions are 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.

9. From **Action Intent**, select one of the following:

Option	Description
Check on Finalization	Applies rule action on finalization of the shopping cart.
Disable Selection	Disables selection of excluded products.

Hide	Hides the selection of excluded products.
	
	 ⓘ If the value **Hide** is not present in the picklist, you can add the value in the picklist by following the steps below.
	     a. Go to **Setup> Create> Objects> Constraint Rule Action> Action Intent**
	     b. Click **New**. Type **Hide** and click **Save**.
	 After adding the value, on the Action Intent details page find the field **Controlling Field** under the section Picklist Options and click **[Change]**. Double-click **Hide** below **Exclusion** (do not select **Hide** for any other Action Types). Click **Save**.

Prompt	Displays a message with choices of products that you can select or remove.
Show Message	Displays a message.

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, auto-inclusion of 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.

10. From **Product Scope**, select one of the following:

Option	Description
Action Criteria	The action criteria are defined using the flare icon (✴).
Product	In the Product field, click to select a product to be included or excluded. You search the product using product name or code.
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.

11. According to the product scope you select, enter values in the last column.

12. Click the setting icon (⚙). A pop-up is displayed. Click **STATEMENTS** tab and fill in the details as explained below.

Option	Description
Statement Name	The name is populated automatically. You can edit the name as you desire.
Match In Primary Lines	Select this option to include or exclude the products that are primary line items (offering) based on your Action Type selection.

Option	Description
Match In Options	Select this option to include or exclude the products that are options based on your Action Type selection. ⓘ When you define an auto-inclusion constraint rule action on option, CPQ overrides default selection in the option group if the option group has **Max Options** set to 1. However, if the option is manually selected by the Sales rep, CPQ does not override the selection.
Match in Cart Options	Select this option to include or exclude the product that are cart line items based on your Action Type selection.
Match in Asset	Select this option to exclude, auto-include, or prompt to include or exclude a product, product family, or product group when a product does not already exist as an installed product.
Repeat Inclusion	Indicates that every time you add a condition product the action product is auto-included. For example, adding Product A to the cart auto-includes Product B then, if you add Product A to the cart again, Product B is auto-included again. By default, action product is only auto-included once. ⓘ Repeat Inclusion is supported only with product scope on server side.

Option	Description
Min/Max Match Rule	Applies to inclusion and exclusion 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. For inclusion type constraint rule: • *Include Any*: Allows selection of up to one product. Rest of the products in the prompt are disabled. • *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 a minimum number of products specified in the Min Products field. For exclusion type constraint rule: • *Exclude All*: All the products are excluded. • *Exclude After One*: Allows selection of only one product. Any other selection will trigger the rule action. ⓘ When you configure a constraint rule with **Action Type** = **Exclusion** and **Min/Max Match Rule** = **Exclude After One**, if the sales representative adds the product for the first time, CPQ displays a constraint rule message on the cart in the "Client" mode. • *Exclude After Max*: Allows selection of a maximum 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
	ⓘ When you define Min/Max for options at Option Group level, then Option Group level configuration is honored over the **Min/Max Match Rule** for prompt type inclusion constraint rule. Hence, if the maximum value that is configured at Option Group level is satisfied then the remaining options are disabled.
Sequence	Enter a number to indicate the order in which constraint rule action messages should be displayed.
Message Type	Select from *Error* and *Warning*. If you select *Error,* you cannot finalize the cart unless you satisfy the constraint rule and if you select *Warning,* you can finalize the cart but the warning message is displayed on the cart until you satisfy the constraint rule.
Message Details	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.

13. Click **Save**.

14. Click the flare icon (⚹) to define action criteria. A pop-up is displayed.
15. Click **New Filter**.
16. Type a **Field**, an **Operator**, and type a **Value**. You can enter more than one criteria.
17. Click **Save**.

You can add multiple actions in a constraint rule. Click add icon (+) and follow the steps again.

Constraint rule actions are created which will trigger when a constraint rule condition is met.

For constraint rules where **Action Type** = **Exclusion** and **Product Scope** = **Product**, CPQ now displays validation error messages from constraint rule actions as hyperlinks on the

Configuration page or Cart page. Clicking on the hyperlink highlights the excluded product set in the product scope.

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.

Running the Constraint Rule Maintenance Job

You must execute Constraint Rule maintenance batch job whenever changes you change constraint rules and when you upgrade to a new version of CPQ. When you run the

Constraint Rule Maintenance job, CPQ inserts new rule records without affecting existing records and the quotes with existing records.

This includes:

- Adding or removing a constraint rule.
- Adding or removing a constraint rule condition and actions.
- Modifying anything in a constraint rule condition and actions.

To run Constraint Rule Maintenance you must have read/write access to ProductConstraintView__c object. You must 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 **Setup > App Setup** > **Create** > **Apps** and click **Edit** next to **Apttus Batch Updates**.
2. Select **Constraint Rule Maintenance** from Available Tabs, click **Add** to add it to Selected Tabs.
3. Click **Save**. The tab should now be available.
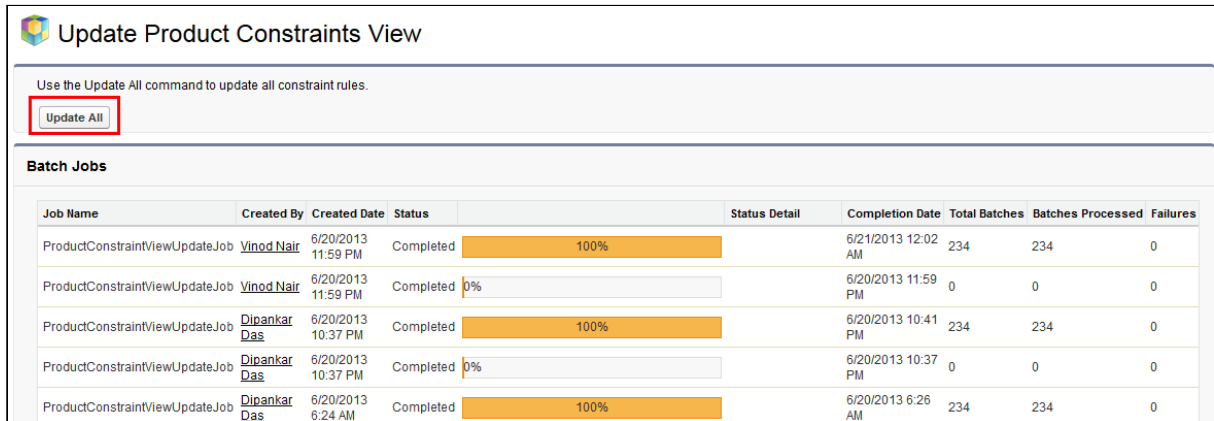
If the tab is still not available, as a workaround you can manually append apex/Apttus_Config2__ProductConstraint View to your URL in the address bar.
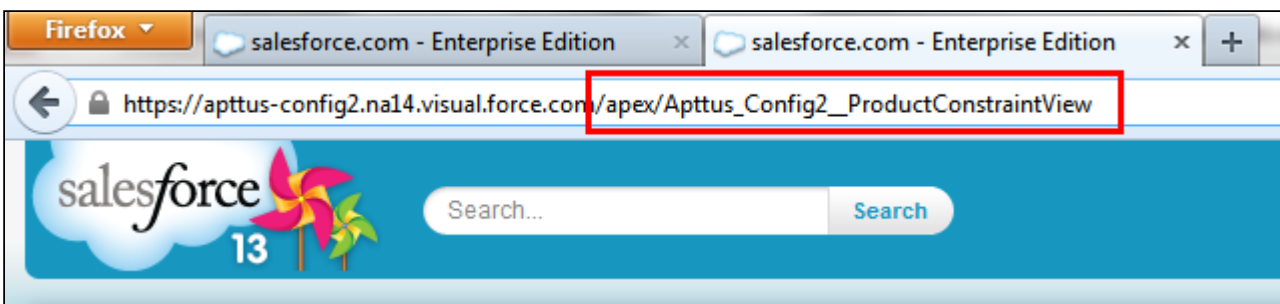
To run a constraint rule maintenance job

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 is executed successfully, even if the percentage indicator remains at 0%.

You can see the status or notes about the Constraint Rule Maintenance job on the Maintenance Console page. Go to **All Tabs** > **CPQ Console** > **Maintenance Console**.

When you have a large number of constraint rules to be processed in the maintenance, CPQ may hit governor limits. You can define a threshold to execute the Constraint Rule Maintenance Job in batches and avoid hitting governor limits. The default percentage value of the threshold, when left blank, is 100%. You can reduce the percentage based on the amount of processing your org performs. The amount of processing is based on the following factor. You can reduce the threshold when you have a number of records or a combination of them:

- Total number of SOQL queries issued
- Total number of records retrieved by SOQL queries
- Total heap size
- Maximum CPU time on the Salesforce servers

To set a threshold for Constraint Rule Maintenance Job

Follow the steps below:

1. Go to **Setup > Develop > Custom Settings > Config System Properties > Manage**.
2. Find **System Properties** and click **Edit**.
3. Enter a number in the **CR Maintenance Governor Limits Threshold** field. The number denotes the percentage of the governor limit.
4. Click **Save**.

Configuring Inclusion Rules

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.

> ⓘ CPQ provides support for maximum products to be added to cart using Inclusion Prompt Rule. Hence, you need to set minimum and maximum both values in order to add products and CPQ disables all remaining option products that go beyond the maximum number post selection. If you select less than the maximum value, you have to manually close the prompt as the prompt closes automatically only when the maximum number is reached.
> To enable checkboxes in the prompt, select **Show Checkbox In Rule Dialog** in Configuring Configuration Engine Settings.

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.

CPQ displays the Auto-Included Product ⟳ icon to distinguish auto-included line items in the cart. The icon is displayed on auto-included standalone, bundle, and option products.

> ⓘ **Note**
>
> Line updates are only applicable on Auto-inclusion rules with the scope as product. Add only 10 line item expressions. Please consult with Conga 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.

Added By and Added By Rule Info

The fields **Added By** and **Added By Rule Info** on the Line Item detail page of a product holds the information about how the product was added to the cart. The **Added By** field holds the value *Constraint Rule when* the product was added through a constraint rule or *User* when the product was added by a Sales Rep. The field **Added By Rule Info** is only populated when the product is added by auto-inclusion or prompt inclusion constraint rule. The information about the constraint rule is displayed in **Added By Rule Info** field.

Configuring Exclusion Rules

Example 1: 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, an 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 the Sales Rep selects Product A, and then selects B. The system shows an error message that Product B is excluded in the addition of Product A.

ⓘ **Hide excluded options at rule or product level**

You can hide the disabled options at an individual product level in addition to the global level wherein it is applied on all products.
You can hide the product options that are excluded by using the Exclusion type constraint rule.
Suppose you have a product, Bundled ABC Software and Hardware with the following options:

1. 24x7 Same Day Onsite Response
2. 24x7 Support for Sev 1
3. Proactive Support Services
4. Remote Account Manager

Then you can select option 1 and disable options 2 to 4. You have the flexibility to hide disabled options at product level. This enables you to use the product configuration for the targeted product.

Example 2: Exclusion rule use case: You have Asset A, Service A. You want to set up an exclusion rule such that when a user selects Asset A on the Installed Product page, on the Service Catalog Service A must be disabled.

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 *Asset A* as the product, and in the **Match Conditions** section, select *Match in Service Assets* 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 *Disable Section*.

5. From **Action Disposition**, select *Error*.

6. From **Product Scope**, select *Product,* and in Product, enter *Service A* 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 Installed page, when the Sales Rep selects Asset A, Service A is disabled on the Service Catalog.

Configuring Validation Rules

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.

Configuring Product Option Group in Product Scope

A constraint rule supports the product option group as a scope.

While creating the constraint rule condition, you can add a Product Option Group picklist value under Product Scope and have a lookup field for the Product Option Group, which is represented by a product ID. These product IDs are different for each bundle and help trigger a constraint rule separately for each bundle. Earlier instead of product IDs, different option groups were created and shared across bundles, which in turn caused malfunctioning of the constraint rules.

Use Case: 24x7 support option must be included when all options are selected from the Additional Services option group (Option Group (Match All) to Product, Auto-include rule) Suppose you are configuring the Conga Snap Performance bundle, when you select the Telephone Support option, Email Support option, and Online Chat Support option, then you can see that the 24x7 Support option is selected by the system automatically.

Similarly, When you select all of the service options from the Additional Services Option Group. you can automatically include a support option for the following constraint rule criteria:

- Option to Option, Option Group (Match All) to Product, Auto-include
- Option to Option, Option Group (Match All) to Product, Recommendation
- Option to Option, Option Group (Match All) to Product, Validation
- Option to Option, Option Group (Match All) to Product, Replacement
- Option to Option, Option Group (Match Any) to Product, Auto-included

Additionally, when you select all of the service options from the Additional Services Option Group. you can automatically exclude a support option for the following constraint rule criteria:

- Option to Option, Option Group (Match All) to Product, Disable Selection
- Option to Option, Option Group (Match Any) to Product, Disable Selection
- Option to Option, Option Group (Match Any) to Product, Recommendation
- Option to Option, Option Group (Match Any) to Product, Replacement
- Option to Option, Option Group (Match Any) to Product, Validation

> ⓘ On the Server side, CPQ supports the Product Option Group as a scope for constraint and attribute rules, wherein the option groups are shared amongst the bundle products through associations.

Scenarios for Using Constraint Rules

State-based Constraint Rules

A constraint rules evaluates a state of the configuration while presenting options. For example, an inclusion rule may have an action to add option A or B to the model, but option A has already been excluded by a different rule that was associated with a different selection Hence, in this case option A is not available for selection.

Use Case:
Bundle 1 - Laptop
Processor 1TB (Option 1)
Processor 2TB (Option 2)

Bundle 2 - CISCO Switch
8 Ports (Option 1)
12 Ports (Option 2)
16 Ports (Option 3)

Cisco Server (standalone)
Office 365 (standalone) Product Group Software
Google suite(standalone) Product Group Software

Rule setup -
Rule 1 - Condition add Cisco Server. Google Suite is excluded. Setup exclusion error rule
Rule 2 - Condition add Cisco Switch, Office 365 is excluded. Setup exclusion warning rule
Rule 3 - Condition add laptop, Product Group Software is included (max 1). Setup inclusion error rule.

Scenario 1 (error) = Rule 1 and 3 are explained below:

- You add Cisco Server to cart, Google Suite is disabled and CPQ displays an error message.
- You configure a laptop, you are prompted to add a product from Product Group Software, Google Suite is disabled on prompt, and Office 365 is available to be added.

Scenario 2 (Warning) = Rule 2 is explained below:
You add Office 365 to cart, then add Cisco Switch to cart, CPQ removes Office 365 and provides a warning message.

Constraint Rules with Prompt Scenarios

On the inclusion prompt, CPQ disables or hides the products that are excluded by an exclusion type constraint rule to keep the validity of the cart intact.

When a product is disabled, CPQ shows an error message on the Cart or Catalog page. However, for the Warning type exclusion rule the product is not disabled in the prompt. When you enable **Hide Disabled Options** setting under Config System Properties, then CPQ hides the product.

Use Case:

Bundle 1 - Laptop

Processor 1TB (Option 1)

Processor 2TB (Option 2)

Bundle 2 - CISCO Switch

8 Ports (Option 1)

12 Ports (Option 2)

16 Ports (Option 3)

Cisco Server (standalone)

Office 365 (standalone) Product Group Software

Google suite(standalone) Product Group Software

Rule setup

Rule 1 - Condition add Cisco Server. Google Suite is excluded. Setup exclusion error rule

Rule 2 - Condition add Cisco Switch, Office 365 is excluded. Setup exclusion warning rule

Rule 3 - Condition add laptop, Product Group Software is included (max 1). Setup inclusion error rule.

Scenario 1 (error)

Jane adds Cisco Server to cart

Google Suite is disabled

System displays Error Message

Jane configures laptop

User is prompted to add product from Product group software

Google Suite is disabled on prompt

Office 365 is available to be added

Scenario 1 (Warning)

Jane adds Office 365 to cart

Jane adds Cisco Switch to cart

System removes Office 365 and provides Warning Message

Jane configures laptop

User is prompted to add product from Product group software

Google Suite and Office 365 is available to be added

Criteria-based Replacement Rules on the Server-side

Replacement rules on Server-side that based on the condition criteria that must be defined using fields from the following objects:

- Line Item
- Product
- Configuration
- Option
- Product Attribute Value

The condition criteria comprise a filter expression with operators and values.

> ⓘ Replacement type constraint rules are not supported on the Client-side.

Use Case for Replacement type constraint rule:

You bought 5 units of Performance Cloud -Enterprise, CPQ prompts to replace the product with Performance Cloud- Ultimate.

Rule has following constraint rule Action and Condition

Condition:

Product scope: Product

Product: Performance Cloud -Enterprise

Condition criteria: Quantity = 5

Action:

Action type: Replacement

Action Intent: Prompt

Product Scope: Product

Product: Premium Support

Disabling Constraint Rule Execution upon Cart Finalization

You can disable the execution of server-side constraint rules to optimize performance upon cart finalization. You must set the Admin Setting *APTS_DisableConstraintRulesOnFinalize* to *true* to use this feature.

> ⓘ This feature is not supported with the Replacement type Constraint Rule.

To disable constraint rule execution

Perform the following steps to define the Admin Setting *APTS_DisableConstraintRulesOnFinalize*:

1. Go to **All Tabs** > **Admin**
2. Click **New**, to create a new record.
3. Fill in the following details

Field	Value
Name	*APTS_DisableConstraintRulesOnFinalize*
Value	*true*
Code	Leave the field blank

4. Click **Save**.

Managing 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* in Configuring Custom Settings.

With the New UI, CPQ allows you to create and configure an 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 CPQ, 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.

The 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. You must add these attributes in the Attribute Value Matrix Entry object as well, in case you want to use Attribute Value Martix.

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.

Edit | Del | Replace	Laptop Color	Laptop_Color__c	Picklist	10/5/2015 10:56 PM	☐
Edit | Del | Replace	Laptop Fingerprint Reader	Laptop_Fingerprint_Reader__c	Picklist	10/5/2015 11:00 PM	☐
Edit | Del | Replace	Laptop HDD	Laptop_HDD__c	Picklist	10/5/2015 10:58 PM	☐
Edit | Del | Replace	Laptop RAM	Laptop_RAM__c	Picklist	10/5/2015 10:58 PM	☐
Edit | Del | Replace	Laptop Screen Size	Laptop_Screen_Size__c	Picklist	10/5/2015 10:59 PM	☐

To create product attribute value

1. Go to **Setup > App Setup > Create > Objects** and select **Product Attribute Value**.

   > ⓘ   If you have more than 800 attributes already create Product Attribute Value object, use Product Attribute Value Extension objects. For more details, refer to Creating Custom Product Attribute Value Extensions.

2. Under Custom Fields & Relationships section, click **New**.
3. Select **Picklist** from the Field Type.
4. Click **Next**.
5. Enter **Field Label**.

   > ⓘ   The attribute label must be unique across all the Product Attribute Value and Product Attribute Value Extension objects. Otherwise, CPQ does not provide correct results when executing constraint rules, product attribute rules, attribute value matrices, price rules, and price Matrices that are defined using attributes with duplicate names.

6. Enter values for picklist.

   > ⓘ   Do not use any special characters such as '<' (less than sign) and '>' (greater than sign) in the product attribute value name. If you use such characters, CPQ displays blank values in certain cases on the UI because picklist entries are getting rendered as HTML. Use '&lt' instead of '<' and '&gt' instead of '>'. You can also use other brackets.

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 the Product Attribute Value object.

About Product Attribute Value Extensions

CPQ provides you the ability to create and configure an unlimited number of attributes, overcoming the 800 record limit of Salesforce. The Product Attribute Extension objects are a replica of the Product Attribute Value object. These objects are an extension to the Product Attribute Value object through a combination of master-detail and look-up relationships. The fields from the extension objects are displayed in the Product Console to be associated as attributes with the products. All of the attributes created under the extension object can be used as normal attributes for your products. For example, you can use the extension objects in Attribute Value Matrices, Product Attribute Rules, Constraint Rules, Numeric Expressions, Document Generation, Asset-Based Ordering, and Asset pricing. These attributes can be used when Sales Rep renews, changes, and amends an asset.

The following extension objects that are available to you in addition to the regular Product Attribute Value object:

- Product Attribute Value Ext
- Product Attribute Value Ext 2
- Product Attribute Value Ext 3

You must use the **Product Attribute Value** and Product Attribute Value Extension objects sequentially. Each product attribute value object has a limit of 800 records per object and you must use the next object only after the limit of the object is reached. For example, you must completely use the **Product Attribute Value** object before utilizing the **Product Attribute Value Ext** object. And you must only use **Product Attribute Value Ext 2** after the **Product Attribute Value Ext** object is completely used.

Similar to the **Product Attribute Value** object, Agreement, Asset, Order, and Proposal Attribute Value objects also have respective extensions. CPQ uses the Agreement, Asset, Order, and Proposal Attribute Value object and the respective extension objects to store the values of the attributes associated with the agreement, asset, order, and proposal line items. Along with Product Attribute Value objects, you must create the same attribute fields in Agreement, Asset, Order, and Proposal Attribute Value objects as well, to retain the attribute values in agreements, assets, order, and proposal line items from configuration line items.

If you define attributes in attribute value extension objects, you must enable the **Populate Attribute Extension** custom setting. The setting allows CPQ to retain the attribute values from configuration line items to the agreement, asset, order, and proposal line items upon creation.

To enable Populate Attribute Extension

You must enable **Populate Attribute Extension** checkbox to indicate that the attribute value extension objects contain attribute fields.

1. Go to **Setup > Develop > Custom Settings > Config System Properties > Manage**.
2. Find **System Properties** and click **Edit**.
3. Enable **Populate Attribute Extensions**.
4. Click **Save**.

> ⓘ
> - This enhancement is available in the CPQ Admin UI only.
> - The attributes added in the Product Attribute Value Extension object are not available for Price Dimensions (Pricing engine). Only the first 800 attributes will be visible for the pricing.

You can also create custom Product Attribute Extension objects. For more information, see Creating Custom Product Attribute Value Extensions.

Creating Custom Product Attribute Value Extensions

> ⚠ From the September '22 release, Conga no longer provides support for custom Product Attribute Value Ext objects. Instead, you must use the out-of-the-box Product Attribute Value Ext objects in CPQ. The existing custom Product Attribute Value Ext objects work as-is, but Conga will not provide any enhancements and will only fix critical bugs.
> Conga recommends you migrate all the attributes from the custom Product Attribute Value Ext object to out-of-the-box Product Attribute Value Ext objects in CPQ. You must not migrate the attributes partially, as custom and out-of-the-box Product Attribute Value Ext objects cannot be used simultaneously. At a time, you can use attributes either from custom or out-of-the-box Product Attribute Value Ext objects.

You can also create custom Product Attribute Extension objects. Follow the steps below.

To create a custom extension object

1. Navigate to **Setup > App Setup > Create > Objects** and click **New Custom Object** to create a new extension object.
2. Enter the 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 custom 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 custom extension object

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

ⓘ You cannot use custom extension objects and standard CPQ extension objects simultaneously. You must only add entries in the **Product Attribute Extension Tables** custom setting if you want to use custom extension objects. Otherwise, leave the field blank.

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**.

You can create attributes in Product Attributes Value Extension object the same way you create attributes in the Product Attribute Value object, just select the extension object that you created in the above-mentioned steps. Refer to Creating Product Attribute Values.

Migrating Attributes from Custom PAV Extension Objects to Out-of-the-Box PAV Extension Objects

ⓘ From the September '22 release, Conga no longer provides support for custom product attribute value (PAV) extension objects. Instead, you must use the out-of-the-box product attribute value extension objects in CPQ. The existing custom product attribute value extension objects work as-is, but Conga will not provide any enhancements and will only fix critical bugs.
Conga recommends you migrate all the attributes from the custom product attribute value extension objects to the following out-of-the-box product attribute value extension objects in CPQ:
   • Product Attribute Value Ext
   • Product Attribute Value Ext 2
   • Product Attribute Value Ext 3

> You must not migrate the attributes partially because custom product attribute value extension objects and out-of-the-box product attribute value extension objects cannot be used simultaneously. At a time, you can use attributes either from product attribute value extension objects or out-of-the-box product attribute value extension objects.

Perform the following steps to migrate attributes from the custom product attribute value extension objects to the out-of-the-box product attribute value extension objects:

1. Create the fields that exist in custom product attribute value extension objects, in the out-of-the-box product attribute value extension objects.
2. Enable product attribute extensions by enabling **Populate Attribute Extensions**. For more information, see About Product Attribute Value Extensions.
3. Remove the custom attribute extension data from custom settings. For more information, see Creating Custom Product Attribute Value Extensions.
4. Change any references to the custom product attribute value extension objects and fields from CPQ areas to their out-of-the-box product attribute value extension objects equivalent, such as:
   - Formula fields, field expressions, and rollups
   - Callback classes
   - Lookup fields
   - Attribute groups
   - Attribute rules and matrices
   - Constraint, pricing, and approval rules
   - Custom workflows
5. Reselect or remap existing attribute selections for saved quotes (attribute selection is changed if a saved quote is reconfigured).

For frequently asked questions on migrating custom PAV extensions to out-of-the-box PAV extensions, see Frequently Asked Questions.

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.

Use Case

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. On **App Menu**, click **Conga CPQ Admin** and then click the **CPQ Admin** tab. The new admin console is launched.
2. On the **Products** menu, click **Manage Products**.
3. Search and click any product, preferably the one you want to associate Attribute Value Matrix with.
4. Click **Rules**. The page displays different rules tab.
5. Click Attribute Value Matrix icon(🟩) next to the title, **RULES**. Attribute Value Matric tab is displayed
6. Click the menu icon(⋮) next to **New Constraint Rule** and click **New Attribute Value Matrix**.
7. Fill in the required details.

Field Name	Description
Matrix Name	Enter name of the Attribute Value Matrix.
Active	Flags whether the Attribute Value Matrix is active.

Field Name	Description
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.
Description	Describe the attribute value matrix.
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.
Matrix Fields	Click the plus icon to add attributes to the attribute value matrix. If you don't select any attribute then, **THEN(statements)...** section remains blank.

8. Select the attribute value matrix you just created from the **RULES** section.
9. Fill in the required details under the **IF(scope)...** section.

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.

10. Define the desired matrix under the **THEN(statements)...** section. Click Add Row to add more rows to the matrix. Click **Add Column** to add more attribute in the matrix. You can add or remove the row using the action menu icon(⋮).
11. 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 **LaptopColor** = Black, **LaptopHDD** defaults to *1TB*. Similarly, if you select **Laptop HDD** as *1TB*, the system defaults **Laptop Color** to *Black*.
12. Click **Save**.

If you create or edit the criteria, you must run **Criteria Maintenance > Update Expression Fields**.

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.

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
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. On **App Menu**, click **Conga CPQ Admin** and then click the **CPQ Admin** tab. The new admin console is launched.
2. On the **Products** menu, click **Manage Products**.
3. Search and click any product, preferably the one you want to associate Product Attribute Rule with.
4. Click **Rules**. The page displays different rules tab.
5. Click Product Attribute Rule icon( ) next to the title, **RULES**. Product Attribute Rule tab is displayed
6. Click the menu icon( ) next to **New Constraint Rule** and click **New Product Attribute Rule**.
7. Fill in the required details.

Field Name	Description
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.

8. Select the product attribute rule you just created from the **RULES** section.
9. Fill in the required details under the **IF(scope)...** section. You can leave them blank if the fields are not required.

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.

Field Name	Description
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.
Criteria	Enter the formula to be evaluated. When this formula evaluates to true, the rule is executed.

ⓘ The Scope value for **Product**, **Product Family**, and **Product Group** can be left blank instead of the default value *All*. Leaving the field blank prevents the rule from searching across all records of that type, improving the scalability performance of product attribute rules. For example, the Product scope can be specific for product value, while leaving **Product Family** and **Product Group** blank, so that they are not queried because they are not needed.

10. Fill in the fields under the **THEN(statements)...** to define the Product Attribute Rule.

Field Name	Description
Target Attribute	Select the target attribute. Use type ahead to display a list of applicable fields or attributes.

Field Name	Description
Action Type	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.
Value Expression	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. • separator - the string to use as a delimiter between values.Separatoris 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, update 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.Ext 1Multiselect__c*. This up-gradation is required to work with PAV Extension Objects.

Allowing Values Conditionally 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

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

Configuring Default Values Conditionally for 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'

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.

Hiding Attributes Conditionally

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	

Configuring Attributes Conditionally 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	

Configuring Attributes Conditionally 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 through 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

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
You want to ensure that the user selects Screen Size.	Laptop Screen Size	

Resetting Attributes Conditionally

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 Text Picklist Multi-picklist	The default value. 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.

Scenario	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'

Scenario to Use Product Attribute Rules with Criteria

Product Attribute rules (PAR) effective date based on quote dates

When the effective date of the quote on the product configuration page fulfills the PAR criteria that you defined, a Rule gets executed.

For example,

Rule Criteria: Apply rules where,
Rule Start Date <= Quote Expected Start Date AND Rule End Date >= Quote Expected End Date Date

PAR Effective Date	PAR Expiration Date	Effective Date (On Product Config)	Result
01-11-2017	30-11-2017	15-11-2017	Rule is getting applied

PAR Effective Date	PAR Expiration Date	Effective Date (On Product Config)	Result
01-11-2017	30-11-2017	31-10-2017	Rule is not getting applied

Managing Pricing

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 associate price lists to a product and create price list items for products at the product level as well. You cannot create a price list at the product level.

> ⓘ   Most of the pricing features are available **Admin UI**. However, some features are currently not available in **Admin UI**. In such cases, you must use **CPQ Console**.

- Managing Price Lists and Price List Items
- Managing Price Rules
- Creating Price Matrices and Dimensions
- Managing CPQ Formula Fields
- Configuring Related Pricing
- Enabling Price Ramps
- Enabling Auto Ramp Creation
- Enabling Tiered Pricing
- Running a Criteria Maintenance Job
- Configuring Contract Pricing
- Configuring Pricing Profiles and Pricing Batch Size
- Configuring Email Notifications for Apex Governor Limit Warning
- Managing Cost and Profitability (Cost Breakdown)
- Configuring Unit of Measure (UOM) and Frequency Conversion Rate
- Configuring the Currency Rounding Mode
- Managing Currency Conversion
- Managing Price Waterfall

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.

In the following sections:

- Creating Price Lists
- Associate a Price List with Categories
- Viewing and Publishing Favorite Configurations for a Price List
- Creating Price List Items
- Configuring Price Escalators

Creating Price Lists

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 in 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 with the price list.

To create a price list

1. On **App Menu**, click **Conga CPQ Admin** and then click the **CPQ Admin** tab. The new admin console is launched.
2. On the **Pricing** menu, click **Manage Price Lists**.
3. Click **New PriceList**.
4. Enter the following details and click **Save**.

Field	Description
PriceList Name	Enter the name of the price list. This is a mandatory field.
Short Description	Enter a description for the price list.
Is Active	Move the slider to the right to enable the price list.
Effective Date	Select a date from when the price list must be effective.

Field	Description
Expiration Date	Select a date in the calendar on which the price list should be expired or made inactive.
Based on Price List	Select the name of an existing price list depending on which the new price list is created.

> ⓘ This field is useful in scenarios such as your organization wants to sell the same products in different currencies or in the same currency with some discount.
> For example, your organization sells products in the United States in the USD currency. You have created a price list called USD Price List. For all products, you have created the required price list items, which are associated with USD Price List. Now you want to sell the same products in Great Britain in the GBP currency. You can create another price list with the following details:
> - **PriceList Name**: GBP Price List
> - **Currency**: GBP - British Pound
> - **Based On Price List**: USD Price List
>
> If you sell a product that costs 1 USD in the Unites States, the price of it will be converted to GBP based on the currency conversion rate defined at **Setup** > **Company Profile** > **Manage Currencies**. In this way, you can create a price list based on another price list, without creating corresponding price list items.

Field	Description
Based on Adjustment Type	Select the type of adjustment. The following adjustment types are available: • % Discount: If the Sales Rep applies a % discount type of adjustment, the List Price is discounted by a percentage specified in the adjustment amount field. The new amount is stored on the cart line item. • Discount Amount: If the Sales Rep applies a discount amount of adjustment, the List Price is discounted by a percentage specified in the adjustment amount field. The new amount is stored on the cart line item. • % Markup: If the Sales Rep applies a percentage markup of adjustment, the List Price is increased by the percentage specified in the % Markup field. The new amount is stored on the cart line item. For example, if you want to increase the price of laptops during the month of September, you can apply a markup amount or a percentage markup that adds to the List price. • Markup Amount: If the Sales Rep applies a markup amount of adjustment, the List Price is increased by the amount specified in the markup amount adjustment field. The new amount is stored on the cart line item. • Price Factor: This is a multiplier. Select this and enter an amount in the **Based on Adjustment Amount** field. Using this price factor, CPQ multiplies the List Price on the cart.
Currency	Click the drop-down list and select the currency. ⓘ The currency defined in the price list overrides the currencies that you define in other objects such as price list item, product, quote, quote line item.
Based on Adjustment Amount	Enter a value based on the adjustment type that you have selected.

Field	Description
Type	Select the type of Price list from the drop-down list. The following price list types are available: • Standard: Select standard price list if the quote derives it pricing from price list and its related price list items configured in the CPQ solution. • Contract: Select contract price list if the quote derives its pricing from an agreement. You must also associate a contract number when you select this price list.
Account	Associate an account with the price list.
Cost Model	Search and select a cost model. A cost model is a container holding all cost types (parent and child cost types). You can associate one cost model with multiple price lists. However, one price list can have only a single cost model associated with it. For more information, see Defining Cost Models and Associating Cost Models with a Price List.
Owner	Enter a name of the owner for Pricing activities.
Contract Number	Enter a contract number to which you want to associate this price list. This enables contract pricing in CPQ.

A price list is created and saved. You can associate all related pricing, products, and options with them.

Associate a Price List with Categories

Price list must be associated to an Offering category for the products to be displayed on the catalog.

The ASSOCIATED CATEGORIES tab on 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 control the display of products by category.

Prerequisites

You must have an existing offering category and a price list.

To associate price list to a category

1. On **App Menu**, click **Conga CPQ Admin** and then click the **CPQ Admin** tab. The new admin console is launched.
2. On the **Pricing** menu, click **Manage Price Lists**. A list of existing price lists is displayed.
3. Select a price list and click the **ASSOCIATED CATEGORIES** tab.
4. From the All Categories pane, drag and drop the required categories to the associated categories pane.
5. 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 Configuring Favorite Settings.
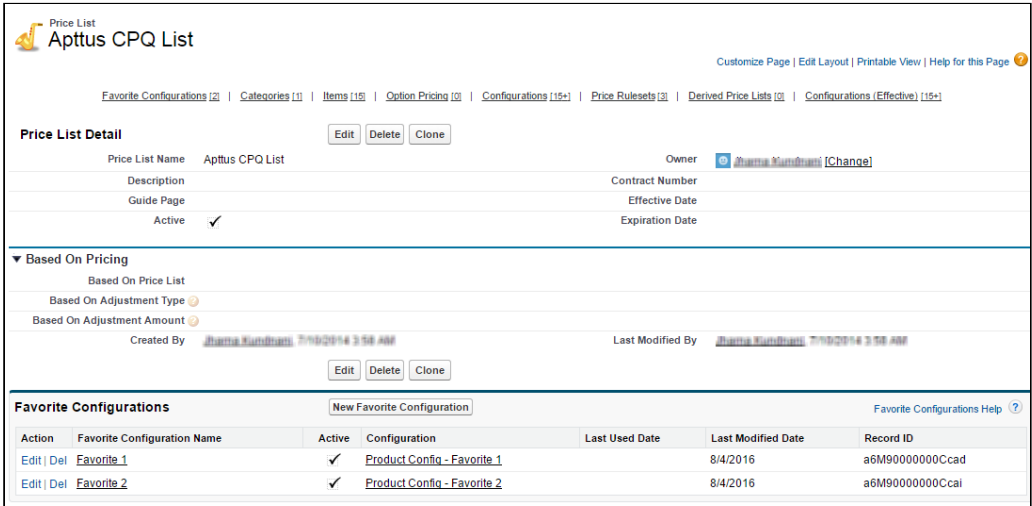
The **Publish** button on your favorite configuration record allows you to publish a favorite configuration record to another price list. This is useful when you want to use the same configuration records across a number of price lists.

Ensure that you have configured Configuring 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

> ⬧ If a field is not available in the Admin UI, you must use CPQ Console to configure it.

> 1. Navigate to the **CPQ Console** tab.
> 2. Click **Manage Price Lists** and select an existing price list.
> 3. Scroll down to the Items related list, and click **New Item**.

1. On **App Menu**, click **Conga CPQ Admin** and then click the **CPQ Admin** tab. The new admin console is launched.
2. On the **Pricing** menu, click **Manage Price Lists**. A list of existing price lists is displayed.
3. Select a price list and click the **PRICELIST ITEMS** tab.
4. From the Products pane, drag and drop the required product to the right side.
5. Enter details in one or more of the following fields:

Field	Description
Charge Type	Select a charge type (mandatory). These are a separate, identifiable element of charges that can be used for pricing. For example: Standard Price, Maintenance Fee, Installation 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 same product (by clicking the ➕ icon) and specify the charge type for each one of the two entries. The charge type will be automatically added to the cart when you add the product so that all charges associated with the product are included in the pricing. Conga does not recommend setting up the bundle and its option to have different frequencies with the same charge type. > ⓘ CPQ does restrict you from creating a PLI without a charge type, but you might not get appropriate results without a charge type.
List Price	Type a list price for a product in the price list item. If the price list item has matrices or may be overridden, enter **0** as the list price. > ❗ You must enter the list price in the accordance with the value defined in the **Currency Field Precision** setting. The precision values for list price must be entered less than or equal to the defined currency precision. If you enter the list price precision values greater than the defined currency precision, CPQ might perform incorrect pricing calculations.

Field	Description
Price Type	Select *One Time*, *Recurring*, or *Usage*. Selecting *Recurring* or *Usage* enables the **Frequency** field. For instance, if you add a subscription product, you will choose 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.
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 List Item fields. • Related Price: Used when the price is calculated as a function of the price of other lines in the same shopping cart. • Tiered Rate: Default price method for Usage Price Type. ⓘ If you select *Usage* in the **Price Type** field, you must ideally select *Tiered Rate* in the **Price Method** field. This is the right combination for the **Price Type** and **Price Method** fields. CPQ does not prevent you from creating other combinations, but you might not get appropriate results.
Active	To make the price list item active, select this check box.

Field	Description
Charge Type Criteria	Click the ⚙ icon and configure a charge type criteria and click **Save**.
	In the CPQ Admin UI, you can use PAV extension fields to create the Charge Type Criteria in price list items. Criteria fields can use attributes only from the out-of-the-box product attribute value (PAV) extension objects. CPQ does not support custom PAV extension objects.
	This is used for conditional charge types that are included only if a certain criteria is met and as part of adding the main product to the cart. Setting this criteria either by Line Item or Attribute field will only apply this price list item under those conditions. For example, this may be a charge type based on workstations; whereas the product may have a charge type based on users.
	ⓘ A product should have at least one Price List Item without any charge type criteria, only then the pricing happens to the product in the Cart. If you are using the **Charge Type Criteria** setting, you must define a criteria for all PLIs. Ensure that you do not define that same criteria for more than one PLI. The criteria must be mutually exclusive so that CPQ does not apply two PLIs while pricing a product.
Percent, Percent Applies To, Related Price List Item	These allow you to set price dependencies such that one price is a percentage of the unit or extended price of another product. Refer Configuring Related Pricing for more information.
Description	Type a brief description to specify the purpose of the price list item.
Price Uom (Unit of measure)	Select an option to specify the unit price you are charging is for an hour, day, year, and so on. This helps the end user while pricing products in the cart. You can modify this in the price list item object.

Field	Description
Min Price/Net Adj %	Enter the minimum price or minimum net adjustment percentage. If the adjustment is a discount, use a negative number (for example, -5). You cannot enter a minimum price or net adjustment percentage that is greater than the maximum price or net adjustment percentage. If you enter, CPQ displays an error message on the Save action. ⓘ When you select an option from the **Min/Max Price Applies To** drop-down, the pricing engine considers the value you enter in this field as the minimum price of the product. When **Min/Max Price Applies To** = *None*, the pricing engine considers the value you enter in this field as the minimum net adjustment percentage. The minimum limit is not applicable to manual adjustments. When the **Min/Max Price Applies To** field is blank, then the **Min Price/Net Adj %** field must have Negative (-ve) value which is used to limit *%Discount.*
Max Price/Net Adj %	Enter the maximum price or maximum net adjustment percentage. You cannot enter a maximum price or net adjustment percentage that is less than the minimum price or net adjustment percentage. If you enter, CPQ displays an error message on the Save action. ⓘ When you select an option from the **Min/Max Price Applies To** drop-down, the pricing engine considers the value you enter in this field as the maximum price of the product. When **Min/Max Price Applies To** = *None*, the pricing engine considers the value you enter in this field as the maximum net adjustment percentage. The maximum limit is not applicable to manual adjustments. When the **Min/Max Price Applies To** field is blank, then the **Max Price/Net Adj %** field must have Positive (+ve) value which is used to limit *%Markup.*

Field	Description
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 the cart. This can be same as the list price or different from the list price, if you have defined a price matrix or a price rule that dynamically arrives at the price of the product based on other factors. • *Extended Price* - This is derived by (Base Price * Quantity * Term) + (Option Price * Quantity) where (Option Price * Quantity) is applicable in case of a bundle scenario. ⓘ When you select an option from the **Min/Max Price Applies To** drop-down, the pricing engine considers the value you enter in the **Min Price/Net Adj %** or **Max Price/Net Adj %** field as minimum or maximum price of the product. When **Min/Max Price Applies To** = *None*, the pricing engine considers the value you enter in the **Min Price/Net Adj %** or **Max Price/Net Adj %** field as a minimum or maximum net adjustment percentage.

CPQ 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.

6. Click **Save**. A price list item is created for a product or option. The price list items created display the products and options to the end user in the pricing cart page.

✅ 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

• Adjusting Options Pricing for a Bundle Manually
  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 the Default Pricing to Products

  You can apply default pricing to a product from the Defaults tab.
- Defining Tax and Billing for Products

  You can define the way your product should be billed and taxed from the Tax & Billing tab.
- Applying Miscellaneous Pricing to Products

  You can apply miscellaneous pricing to a product from the Miscellaneous tab.
- Configuring Charge Type Frequency as Weekly

  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.

Adding Additional Details to a Price List Item

1. Select an existing price list item, click the more icon (⋮), and click **Advanced**.
2. Click the **DETAILS** tab.
3. Enter the following details:

Field	Description
Price List	Displays the price list, which is disabled for editing.

Field	Description
Charge Type Criteria	Click the ⬀ icon and configure a charge type criteria and click **Save**.
	This is used for conditional charge types that are included only if a certain criteria is met and as part of adding the main product to the cart. Setting this criteria either by Line Item or Attribute field will only apply this price list item under those conditions. For example, this may be a charge type based on workstations; whereas the product may have a charge type based on users.
	ⓘ A product should have at least one Price List Item without any charge type criteria, only then the pricing happens to the product in the Cart. If you are using the **Charge Type Criteria** setting, you must define a criteria for all PLIs. Ensure that you do not define that same criteria for more than one PLI. The criteria must be mutually exclusive so that CPQ does not apply two PLIs while pricing a product.
Product	Search and select a product.
Charge Type	Select a charge type. These are a separate, identifiable element of charges that can be used for pricing. For example: Standard Price, Maintenance Fee, Installation Fee, and more.
Active	To make the price list item active, turn on the toggle button.
Sequence	Enter a sequence for the price list item.
List Price	Type a list price for a product in the price list item. If the price list item has matrices or may be overridden, enter **0** as the list price.

Field	Description
Cost	Type a cost of a product.
	ⓘ You can set the cost in the price list currency. If the quote price list is in a different currency, the sales user will see the cost with the correct conversion rate applied. The subsequent fields of cost such as base cost, base extended cost, and extended cost are calculated after the currency conversion. If the **Disable Based On Currency Adjustment** setting is enabled on the price list, the currency conversion is not applied.
Effective Date	Enter an effective date for the price list item.
Expiration Date	Enter an expiry date for the price list item.
Price Type	Select *One Time*, *Recurring*, or *Usage*. Selecting *Recurring* or *Usage* enables the **Frequency** field. For instance, if you add a subscription product, you will choose 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.
Frequency	This is based on your selection from the **Price Type** field. Select an option to indicate how often the product will be charged.

Field	Description
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 List Item fields. • Related Price: Used when the price is calculated as a function of the price of other lines in the same shopping cart. • Tiered Rate: Default price method for Usage Price Type. ⓘ If you select *Usage* in the **Price Type** field, you must ideally select *Tiered Rate* in the **Price Method** field. This is the right combination for the **Price Type** and **Price Method** fields. CPQ does not prevent you from creating other combinations, but you might not get appropriate results.

4. Click **Save**.

Adjusting Options Pricing for a Bundle Manually

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, click the more icon (⋮), and click **Advanced**.
2. Click the **OPTIONS** tab.
3. Enter information in one or more of the following editable columns:

Field	Description
Adjustment Amount	Enter an amount to adjust the options price. ⓘ For Optional bundle, if you apply a discount on the bundle level, it is not populated in the Adjustment Amount and Adjustment Type field on the option. Discount is getting applied on the line Items, however, Adj Type and Amount fields are not getting populated for line items and Adj Type and Amount field are not getting cleared from the bundle product.
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.

4. Click **Save**.

The option price for a bundle is adjusted and saved.

ⓘ **Enabled Group Adjustment Spread and Enforced Min/Max Net %**
  - You can prevent the sales reps from applying any line level adjustments on one or more items in the cart because these items are sold at the price that CPQ calculates.
  - If you have enabled the `Enable Adjustment Spread` custom setting and disabled the `Allow Manual Adjustment` on a line item, then CPQ sets the `Allocate Group Adjustment` as read-only for that line item. Note that

the sales reps cannot use **Mass Update** to update the **Allocate Group Adjustment** flag.
- This feature is based on client side pricing framework, hence you must set `Enable Auto Reprice` to True.
- After configuring the Min Net % and Max Net % on a price list item, you cannot apply the manual price adjustment outside of Min/Max Net % range.
- You can give percentage-based discounts for configuring a quote with a desired price by using group adjustment spread when you have Min/Max Net % associated with the line items.
- This feature can be used only if the bundle has no price of its own. This is because the allocation of group adjustments is only to the options and the bundle's own price will not be adjusted.
- Ability to set Minimum and Maximum Net Adjustment % is enabled only when "Enable Auto Reprice" is turned ON.

> ⓘ To use Min/Max Net %, you must ensure that the **Adjustment Applies To** field is empty.

Conversion of Group Adjustments when bundle and options have ramps

Group Adjustment Spread provides you easiness of applying a group adjustment with the flexibility of a line level adjustment. You can update the adjustment type and amount on a bundle or summary group line and on reprice, the group adjustment gets replaced by line level adjustments on the options or standalone products.

Use Case 1:

When the bundle has three ramps with Start Date and End Date as:

Start Date	End Date
1-Jan-2017	31-Dec-2017
1-Jan-2018	31-Dec-2018
1-Jan-2019	31-Dec-2019

Then suppose there are options with ramp lines with the same dates. In that scenario, if you apply an adjustment to one of the ramp lines, only the corresponding ramp line gets adjusted and the other ramp lines remain the same. If a bundle does not have ramps, but the options have ramps in that case adjustment spreads to all the ramp lines.

Use Case 2:

> If a bundle has a total price of $600, price override of $300, and one of the options
> has three ramp lines with extended prices of $100 each, then each of the option gets
> a price override of $50 (assuming all the options of the bundle are eligible and the
> corresponding option has flat rollup method).
> - You can use the Group Adjustment Spread functionality even if the bundles
>   have sub-bundles.
> - Group Adjusument Spread functionality works for Discount Amount, Markup
>   Amount and Price Override as well for the conversion of amount-based bundle
>   adjustments to option adjustments.
> - You can quicly apply adjustments using the Group Adjustment Spread
>   functionality, althogh you have optional cart line items.

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 - Line Item.

Applying the Default Pricing to Products

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, click the more icon (⋮), and click **Advanced**.
2. Click the **DEFAULTS** tab.
3. Enter the following details:

Field	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. If you select Product Attribute from the **Default Quantity From** drop-down, you can select any field defined on the PAV extension objects for defaulting quantity.

Default Quantity	Enter the default quantity.
	
	ⓘ The default quantity of an option product defined here at the price list item is overridden by the default quantity of the option product defined at the bundle structure level. The precedence for calculating the default quantity is as follows:
	   a. Transactional input provided by the sales representative on the Configuration page
	   b. Default quantity defined at the bundle structure level
	   c. Default quantity defined at the price list item level

Default Price From	Select an option from where default price can be considered. Supported values are: • Advanced Formula • Product Attribute
Default Price Field	Enter the name of the CPQ formula field (it can be normal or based on numeric expression) and select it from the search result. Whatever the value this field returns, that will be the list price of the product. ⓘ You must have selected *Advanced Formula* in the **Default Price From** for this field to work. If you select Product Attribute from the **Default Price From** drop-down, you can select any field defined on the PAV extension objects for defaulting price.

Field	Description
Default Selling Term	Enter the default selling term.
	ⓘ If a product has two PLIs of *Recurring* type and PLI 1 has Default Selling Term mentioned on it while PLI 2 does not have Default Selling Term, then on product configuration, the selling term of PLI 2 gets auto-updated in line with the selling term of PLI 1. When Price Type = Recurring, CPQ calculates the selling term on the cart as follows: • If you provide the Default Selling Term (but the cart line item does not have start date and end date), CPQ calculates the selling term based on the default selling term. • If the cart line item has start date and end date (but you do not provide the Default Selling Term), CPQ calculates the selling term based on the start date and end date. • If you provide the default selling term and the cart line item also has start date and end date, CPQ gives precedence to the start and end dates, and calculates the selling term accordingly.
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.

Field	Description
Auto Cascade Selling Term	Select to automatically cascade selling term. ⓘ Suppose you have one bundle with two options. Option 1 has Auto Cascade Selling Term set to False. Option 2 has Auto Cascade Selling Term set to True. If you change the bundle selling term, it will only cascade to Option 2, but not Option 1.
Is Selling Term Read Only	Select this to make the Selling Term field non-editable on the cart.
Auto Renew	Select this setting only when you want CPQ to automatically close the renewal quote that has an asset with **Auto Renew = True**. On the expiry date of the asset, CPQ automatically processes the renewal quote and renews the asset for the next term. The quote will be automatically set to *Accepted* without any intervention (this is called *Touchless Renewal*). This setting is used to group assets into different renewal quotes, along with the **Renewal Group Fields**. CPQ allows you to configure this setting at the price list item level, which will cascade to the cart and asset level. ⓘ Do not use this setting to create auto renewal quote.

Field	Description
Auto Renewal Type	CPQ allows you to configure an asset as renewable or non-renewable at the price list item level. The supported values are: • *Fixed*: Select this option to make the asset renewable. CPQ can pull the product into Auto and OnDemand renewal quotes. • *Evergreen*: Select this option to define the selling term without an expiration date. This option makes the asset billable forever until the subscription is cancelled (evergreen billing). For example, if the product is of monthly subscription type, the customer keeps receiving the bills for the asset monthly (without requiring renewals). You cannot pull the product into an Auto or OnDemand renewal quote. For such assets, CPQ disables the **Renew** action on the Installed Products page (Sales Representative cannot renew the asset manually also). ⓘ Asset-Based Ordering is not supported for Evergreen assets. • *Do Not Renew*: Select this option to disable CPQ from pulling the asset into Auto or OnDemand renewal quotes. This value makes the asset non-renewable. ⓘ If you do not select any option for **Auto Renewal Type**, CPQ considers the asset as renewable. CPQ allows you to configure this setting at the price list item level, which will cascade to the cart and asset level.
Auto Renewal Term	Enter a value to define the auto renewal term that CPQ must consider while renewing assets. If you do not specify any value, CPQ considers the **Default Renewal Term** specified in Installed Product Settings. If both **Auto Renewal Term** and **Default Renewal Term** are null, CPQ considers the selling term on the original asset during renewal. CPQ allows you to configure this setting at the price list item level, which will cascade to the cart and asset level.

4. Click **Save**. Default pricing is applied and saved.

Defining Tax and Billing for Products

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, click the more icon (⋮), and click **Advanced**.
2. Click the **TAX & BILLING** tab.
3. Enter the following details:

Field	Description
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

Field	Description
Min Usage Quantity	If you defined the **Billing Frequency** to be usage based, enter a value to set the minimum quantity of billable consumption.
Max Usage Quantity	If you defined the **Billing Frequency** to be usage based, enter a value to set the maximum quantity of billable consumption.

4. Click **Save**.

Defining Finance and Revenue for Products

You can configure all finance and billing related information for your product from the FINANCE & REVENUE tab.

To define finance and revenue

You must have an existing price list item.

1. Select an existing price list item, click the more icon (⦙), and click **Advanced**.
2. Click the **FINANCE & REVENUE** tab.
3. Enter the following details:

Field	Description
Revenue Recognition Policy	Enter a revenue recognition policy associated with a product or service.
Revenue Split Policy	Enter an order revenue split policy associated with a product or service.

4. Click **Save**.

Applying Miscellaneous Pricing to Products

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, click the more icon (![icon]), and click **Advanced**.
2. Click the **MISCELLANEOUS** tab.
3. Select the following options as required:

Field	Description
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 Ramps	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.

Field	Description
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 Conga Billing 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. ⓘ CPQ does not support suppression of bundle products and does not encourage suppression of standalone products.
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.
Enable Auto Ramp Creation	Select this check box to auto generate ramp line creation for the 2nd charge type, if price ramp is enabled for the first charge type.
Allow Proration	Select to allow proration. On the Cart page, to make Selling Frequency editable by the user, this check box must be selected.
Disable Cost Model	Select this check box to disable cost model and therefore, cost line items are not created when a product is added to the cart.

4. Click **Save**.

Miscellaneous pricing items applied to a product and saved.

Configuring Charge Type Frequency as Weekly

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.

Disabling Time Adjustment in Price List Item Fields

You can disable the time zone adjustment on the **Effective Date** and **Expiration Date** fields in the Price List Item. By default, the time on the **Effective Date** and **Expiration Date** fields is adjusted based on the Time Zone of the Locale in the org. If you disable the adjustment, CPQ does not change the time you manually configured in the **Effective Date** and

Expiration Date fields. Refer to Adding Additional Details to a Price List Item for information about the Price List Item fields.

To disable time adjustment in Price List Item fields

1. Go to **All Tabs** > **Admin**
2. Click **New**, to create a new record. The New Admin page is displayed.
3. Fill in the following details

Field	Value
Name	*APTS_DisablePLITimeZoneAdj*
Value	The following are the valid values: • *true*: disables time zone adjustment • *false*: enables time zone adjustment
Code	Leave the field blank

4. Click **Save**.

Configuring Price Escalators

Price Escalator is a new mechanism to define a markup on the different prices for a product. The markup can be defined on a previous or first ramp line on the cart.

On which prices is it applicable?

The markup is applied on the List Price, Base Price or Extended Price.

Applicable for which products?

The Price Escalator can be applied on Bundle and Option products. Please note that the bundle level escalator does not cascade to the option products.

Where is this applied?

The markup is applied on the first or previous ramp line (defined by the **Period** field) of bundle or option products on the cart.

When is this applied?

For bundle products, the price escalator is applied after the Price Matrices.

For option products, the price of the ramps is calculated depending on the value in the **Product Option Price Order** field.

> ✅ **Usecase**
>
> If the matrix gives a base price is 60 and if the YOY ramp markup has been defined as 10% over base price on period 2 and period 3, then the first ramp line will be $60, second will be $66 and the third will be $72.60.

How is price escalator applied on ramp lines during ABO operations?

CPQ applies Price Escalator on new ramp lines created during ABO operations regardless of the **Default Asset Pricing Indicator?** setting (true or false). A new ramp line always gets its value from the price list item (PLI) pricing. For more information, see Use Case: Applying Price Escalator on New Ramp Lines During ABO Operations.

To configure a Price Escalator

1. Navigate to your Price List Item record and search for **Price Escalator** related list.
2. Click **New Price Escalator** and enter the following details:

Field	Description
Period Number	Defines the ramp line number.
Adjustment Source	Defines the ramp on which the adjustment is applied.
Adjustment Applies To	Values for the different prices to which the adjustment is applied. Valid values are List Price, Base Price and Extended Price.
Adjustment Type	Values for manual adjustments whether you want to apply discount or markup on the price.
Adjustment Amount	Field to enter in the amount or percentage of manual adjustment.
Price List Item	This is a lookup to the Price List Item, containing the PLI Id by default.

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 Rulesets

  Price Rulesets allow for pricing adjustments across a range of products through 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.

- Managing Price Rules

  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 Rulesets

Price Ruleset allows pricing adjustments across range of products through 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. You can achieve different types of pricing strategies and capabilities that can be achieved using Price Rulesets such as Discounting rules, promotional pricing, adjustments in aggregate based on a Product Family, and more.

For example, if a customer has a Gold Rating, and you want to automatically apply a 30% discount to specific line items on a proposal, you can achieve this using a Price Ruleset.

To create a price ruleset

You must have an existing price list.

1. On **App Menu**, click **Apttus CPQ Admin** and then click the **CPQ Admin** tab. The new admin console is launched.
2. On the **Pricing** menu, click **Manage Price Ruleset**.
3. Click **New Price Ruleset**.
4. Enter details in one or more of the following fields, as required:

Field	Description
Price Rule Set Name	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
Is Active	Select this to set the ruleset as active.
Enable Currency	Select this to enable currency for the price ruleset.
Currency	Select the currency for the price ruleset. ⓘ If the price ruleset and a line item have different currencies, CPQ applies the currency conversion rate (defined at **Setup** > **Company Profile** > **Manage Currencies**) when applying the rule adjustment.
Short Description	Enter a description for the ruleset.
Effective Date	Not required as long as the price ruleset is Active, but can be used for promotional rules, and more.
Price List	Select a price list. The ruleset will only source product prices with the selected price list and then apply adjustment criteria.
Product Category	Select a category. The ruleset will only source product prices within the selected category.
Category	Select All, Agreement, or Proposal. This indicates if the ruleset is relevant to Agreements, Proposals, or both.

Field	Description
Application Level	Select a level to which this ruleset will be applied. The supported values are: • *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 the adjustment is displayed in the Totals tab in the shopping cart. • *Aggregate:* Selecting this enables the **Application Method** field.
Application Method	This field is enabled if you selected *Aggregate* from the **Application Level** drop-down list. 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. The supported values are: • *Apply to Line Items*: Applies the adjustment to line items on the cart. • *Create Summary Lines*: Applies the adjustment as a summary line in the Totals tab on the cart.
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. In CPQ Admin UI, you can use attributes from the out-of-the-box product attribute value (PAV) extension objects in the following ways: • To create price dimensions, which can be used to create price rulesets • To create Ruleset Criteria in price rulesets ⓘ CPQ does not support custom PAV extension objects.
Stop Processing more Rules	Selecting this indicates that if the ruleset is satisfied, the system will not process any more rules.

Field	Description
Enable Date Range	Selecting this enables you to set the effective date and expiration date on the rule entry. ⓘ When you hover the info icon, it will display the message: *Enable this to have a start and end date for the rule entries.*
Expiration Date	Select an expiration date.
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 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.

5. Click **Save**.

By filling out these criteria, the source products and prices are then designated and the ruleset can apply adjustments through price rules.

Creating Price Rules

You can associate multiple rules to a rule set 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 a 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.

In the CPQ Admin UI, you can use PAV extension fields to create price rules of type criteria. Criteria fields can use attributes only from the out-of-the-box product attribute value (PAV) extension objects. CPQ does not support custom PAV extension objects.

To create a price rule

You must have an existing Price Ruleset.

1. On **App Menu**, click **Conga CPQ Admin** and then click the **CPQ Admin** tab. The new admin console is launched.
2. On the **Pricing** menu, click **Manage Price Ruleset**.
3. Select an existing price ruleset.
4. On the **PRICE RULES** tab click **New Price Rule**.
5. On the **DETAILS** tab, enter the following details.
   a. In **Rule Name**, enter a mandatory rule name.
   b. In **Short Description**, enter a description for the rule.
   c. Selecting **Stop Processing More Rules** indicates that if the ruleset is satisfied, the system will not process any more rules.
   d. From **Rule Type**, select a type. The supported values are *Dimension* and *Condition*. In setting a dimensional price rule, the process is exactly the same as creating a price matrix for a product.

   > ⓘ The values on the **Rule Type** drop-down are based on what you add in **Setup** > **App Setup** > **Create** > **Objects** > **Price Rule** > **Rule Type** > **Values**. CPQ supports only *Dimension* and *Condition* rule types.

   e. From **Adjustment Applies To**, select an option to which the adjustment is applied. This is for line item adjustments only. The supported values are Base Price and Base Extended Price.
   f. By default, the price rule inherits the value of **Currency**, from the price ruleset.
   g. The **Sequence** is auto-generated.
   h. To make the rule active, select **Active**.
   i. 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.
   j. To allow complete removal of a price adjustment, select **Allow Removal of Adjustment**.
6. On the **DIMENSIONS** tab, enter the following details.
   a. 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. For example, you can select a dimension on quantity.
   b. 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.

7. Click **Save**. The price rule is successfully created and listed under the Price Rule list.
8. Click the add icon (✚) in the price rule details section to add an adjustment to the price rule. You can add more than one price rule rows by clicking ✚ and delete rows by clicking ➖. You can also move a row up or down by clicking 🔼 or 🔽.
9. Enter the following details and click **Save**.
   a. In the ***<Dimension>*** column, enter a value for the dimension you created.
   b. In the **Adjustment Amount** column, enter a percentage or a number to adjust the price.

   > ⓘ In this column you enter the adjustment amount directly. Use this column if you are not using the **Adjustment Amount Source** column.

   c. In the **Adjustment Type** column, select the kind of adjustment required.
   d. In the **Adjustment Amount Source** column, select an adjustment source.

   > ⓘ Use this column if you are not using the **Adjustment Amount** column. In this column, you can select an adjustment amount that will be fetched from CPQ formula fields created on the required business objects and fields. CPQ fetches the adjustment value from those fields instead of fetching the adjustment amount directly from the **Adjustment Amount** column. However, you must select such formula fields that return only numeric values. CPQ applies the adjustment on the list price to derive the base price.

   e. In the **Effective Date** column, select the date from when the price rule entry will be effective.

   > ⓘ This column is available only if the **Enable Date Range** is turned on. For more information, see Creating Price Rulesets.

   f. In the **Expiration Date** column, select the date on which the price rule entry will expire.

   > ⓘ This column is available only if the **Enable Date Range** is turned on. For more information, see Creating Price Rulesets.

g. In the **UOM** column, enter the UOM for the price rule entry so that CPQ applies the UOM conversion rate if the price rule entry and the line item have different UOMs.

> ⓘ You can set the UOM of a price rule entry when the **Adjustment Type** is Discount Amount, Markup Amount, or Price Override. Also, you must have configured the conversion rate for UOMs. For more information, see Configuring Unit of Measure (UOM) and Frequency Conversion Rate.
> The UOM you enter in the price rule entry here must be the source during conversion (**From Uom**).

Creating Price Matrices and Dimensions

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 a price dimension

1. On **App Menu**, click **Conga CPQ Admin** and then click the **CPQ Admin** tab. The new admin console is launched.

2. On the **Pricing** menu, click **Manage Price Dimensions**. A list of existing price dimensions is displayed.
3. Click **New Price Dimension**.
4. In the **Name** field, enter a mandatory name for the dimension.
5. From the **Context Type** drop-down, select a context type.

> ⓘ CPQ supports the out-of-the-box context types described in the following table. You can manage these context types at **Setup** > **App Setup** > **Create** > **Objects** > **Price Dimension** (object) > **Context Type** (custom field) > **Values** section.
>
> For a context type where the **Cumulative Dimension** field is applicable, you must use this field for cumulative range. Therefore, on the **Cumulative Dimension** drop-down, select **Quantity** only because the cumulative range is supported only for quantity.

Context Type	Description
Line Item	Allows you to choose any field on the line item shopping cart. For example: Quantity, Term, and more. When you select this context type, the following fields are enabled. • **Business Object**. This field is populated by default based on the Context Type selection. You cannot modify it. • **Field**: Select a line item field. • **Relation Type**: Select a relation type between the context type and the line item field. • **Cumulative Dimension**: Cumulative Dimension is applicable for Asset operations such as increment quantity. If you want to use a value from another dimension with the current dimension and use the resultant value as the final value, select a *Dimension Name* from the drop-down. The value from another dimension will be added to the value of the current dimension during evaluation. For example, you are creating a dimension on quantity and want to add the value of existing asset quantity to get the total quantity (Asset Quantity + Current Quantity) value, which will be the value evaluated when your current dimension is evaluated.

Context Type	Description
Product Attribute	Allows you to choose any product attribute that has been associated to a product through an attribute group. From **Attribute**, click to select an attribute group. When you select this context type, the following fields are enabled. • **Business Object**. This field is populated by default based on the Context Type selection. You cannot modify it. • **Attribute Group**: Search and select an attribute group. • **Attribute**: Select an attribute in the attribute group selected above. • **Cumulative Dimension**: Cumulative Dimension is applicable for Asset operations such as increment quantity. If you want to use a value from another dimension with the current dimension and use the resultant value as the final value, select a *Dimension Name* from the drop-down. The value from another dimension will be added to the value of the current dimension during evaluation. For example, you are creating a dimension on quantity and want to add the value of existing asset quantity to get the total quantity (Asset Quantity + Current Quantity) value, which will be the value evaluated when your current dimension is evaluated.

Context Type	Description

Custom	Allows you to define a custom syntax for CPQ to retrieve the value of a field. 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.INSERTYOURFIELDAPINAMEHERE`
	
	-or-
	
	For Agreement Header field:
	
	`Apttus_Config2__ConfigurationId`
	`r.Apttus_CMConfig__AgreementId__r.INSERTYOURFIELDAPINAMEHERE`
	
	> ⓘ The field must be the fields API name and not the field label or name.
	
	When you select this context type, the following fields are enabled.
	
	• **Business Object**. This field is populated by default based on the Context Type selection. You cannot modify it.
	• **Business Formula**: Define a formula field.
	• **Cumulative Dimension**: Cumulative Dimension is applicable for Asset operations such as increment quantity. If you want to use a value from another dimension with the current dimension and use the resultant value as the final value, select a *Dimension Name* from the drop-down. The value from another dimension will be added to the value of the current dimension during evaluation. For example, you are creating a dimension on quantity and want to add the value of existing asset quantity to get the total quantity (Asset Quantity + Current Quantity) value, which will be the value evaluated when your current dimension is evaluated.

Context Type	Description
Formula Field	Allows you to create a formula field containing the syntax based on the custom syntax (refer to the context type *Custom*). 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. When you select this context type, the following fields are enabled. • **Business Object**. This field is populated by default based on the Context Type selection. You cannot modify it. • **Formula Field**: Select a predefined formula field. • **Cumulative Dimension**: Cumulative Dimension is applicable for Asset operations such as increment quantity. If you want to use a value from another dimension with the current dimension and use the resultant value as the final value, select a *Dimension Name* from the drop-down. The value from another dimension will be added to the value of the current dimension during evaluation. For example, you are creating a dimension on quantity and want to add the value of existing asset quantity to get the total quantity (Asset Quantity + Current Quantity) value, which will be the value evaluated when your current dimension is evaluated.

Context Type	Description

Asset Line Item	Applicable for Service CPQ.
	
	Allows you to choose any field on the asset line item. When you select this context type, the following fields are enabled.
	
	• **Business Object**. This field is populated by default based on the Context Type selection. You cannot modify it.
	• **Field**: Select an asset line item field.
	• **Cumulative Dimension**: Cumulative Dimension is applicable for Asset operations such as increment quantity. If you want to use a value from another dimension with the current dimension and use the resultant value as the final value, select a *Dimension Name* from the drop-down. The value from another dimension will be added to the value of the current dimension during evaluation. For example, you are creating a dimension on quantity and want to add the value of existing asset quantity to get the total quantity (Asset Quantity + Current Quantity) value, which will be the value evaluated when your current dimension is evaluated.
	
	ⓘ For Service CPQ, you must manually add the following values for using a price dimension:
	    • **Object**: Price Dimension (Apttus_Config2__PriceDimension__c)
	    • **Picklist Field**: Context Type (Apttus_Config2__ContextType__c)
	    • **Value:** Asset Line Item

Context Type	Description
Asset Attribute	Applicable for Service CPQ. Allows you to choose any asset attribute that has been associated to an asset. When you select this context type, the following fields are enabled. • **Business Object**. This field is populated by default based on the Context Type selection. You cannot modify it. • **Attribute**: Select an asset attribute. ⓘ For Service CPQ, you must manually add the following values for using a price dimension: • **Object**: Price Dimension (Apttus_Config2__PriceDimension__c) • **Picklist Field**: Context Type (Apttus_Config2__ContextType__c) • **Value:** Asset Attribute

6. From the **Type** drop-down, select one of the following types of price dimensions:

> ⓘ The field **Type** is related to promotions. Based on the selected value, CPQ creates a dimension type for promotion. For example, you can create a quantity-based dimension by selecting *Quantity* from the **Type** drop-down. For more information, see Creating Price Dimensions for a Promotion.
> To set up values for the **Type** field, go to **Setup** > **Create** > **Objects** > **Price Dimension** (Apttus_Config2__PriceDimension__c) > **Type** (Apttus_Config2__Type __c) field.

Type	Description
Amount	Select this if the price dimension is based on purchase amount.
Equipment	Applicable for Service CPQ. Select this value if the price dimension is for Service CPQ.
Quantity	Select this if the price dimension is based on quantity.
Standard	Select this for a standard price dimension.

Type	Description
Term	Select this if the price dimension is based on term.

7. In the **Description** field, enter a description for the price dimension.
8. Click **Save**.

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.

As an administrator, you can associate a cost (custom field) to each price matrix entry because the cost varies with the price and hence the margin is affected. The price matrix entries that are applied on a line item can be accessed in the line item. A price list can have more than one price matrix. CPQ supports multiple matrix entries within a matrix (cumulative range matrix) to be applied.

In CPQ Admin UI, you can use attributes from the out-of-the-box product attribute value (PAV) extension objects in the following ways:

- To create price dimensions, which can be used to create price matrices
- To create price matrices of type criteria

> ⓘ CPQ does not support custom PAV extension objects.

To create price matrices

You must have existing price dimensions.

1. Select an existing price list item, click the more icon (⣿), and click **Advanced**.
2. On the **MATRICES** tab, click **New Matrix**.
3. On the **DETAILS** tab, enter the following details.
   a. **Matrix Name**: Enter a name for the matrix.
   b. **Sequence**: Enter a sequence number for the matrix.
   c. **Short Description**: Enter a description for the matrix.
   d. **Matrix Type**: Select *Dimension*. If you choose *Condition*, note that the option or attribute pricing at the 1st level sub-bundle is not supported.

> ⓘ If **Price Type** is *Usage* on a price line item, then the **Matrix Type** should also be *Usage* while creating a price matrix. CPQ supports only this combination.
> If **Price Type** is *Usage* on a price line item and **Matrix Type** is *Dimension*, and you try to use the **Adjustment Amount Source** field, it will not work. CPQ does not support this combination.

   e. **Stop Processing More Matrices**: Select this checkbox to stop the system from evaluating further matrices if a match is found within this matrix.

   f. **Enable Date Range**: Select this checkbox to enable the start and end date for price matrix to execute.

> ⓘ In Classic CPQ, the price matrix date is mapped with the start date of line item. If the start date of the line item on the Cart falls under the price matrix entry, CPQ applies the correct price matrix.

4. On the **DIMENSIONS** tab, enter the following details.
   a. 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.
   b. From **Dimension Value Type**, select one of the following:

Option	Description
Cumulative Range - Line Item	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.

ⓘ As an administrator, you define step pricing so that users (sales representatives) can quote correct prices to customers. You can create a matrix where the Dimension Type is Cumulative Range. If the admin setting *APTS_DisableCumulativeQuantityAcrossCart* is set to True, CPQ applies the cumulative range matrix such that it does not span multiple line items of the same product. The Admin Setting *APTS_DisableCumulativeQuantityAcrossCart* is not available in Admin tab by default. You can add the entry by following the steps below:

    i. Go to **All Tabs** > **Admin**
    ii. Click **New**, to create a new record.
    iii. Fill the following details

Field	Value
Name	*APTS_DisableCumulativeQuantityAcrossCart*
Value	*true* or *false*
Code	Leave the field blank

    iv. Click **Save**.

When you have applied *Cumulative Range - Line Item price* matrix on a product, if you create ramps for that line item, those ramps will have the matrix applied by default. Ramps will inherit the matrix from that line item. (If you create five ramps, all five ramps will have the matrix.)

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.

5. Click **Save**. The price matrix is successfully created and listed under Matrices.

6. Click the add icon (+) in the matrix detail section to add an adjustment to the price matrix. 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 🔽.

7. Enter the following details and click **Save**.

   a. In the ***<Dimension_Name>*** column, enter a value for the dimension you created. 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.

   b. In the **Adjustment Amount** column, enter a percentage or a number to adjust the price.

      > ⓘ In this field you enter a static adjustment amount value. If you do not want the adjustment amount to be static, use the **Adjustment Amount Source** column.

   c. From **Adjustment Type**, select the kind of adjustment required, if the criteria is met. An example is an override to a list price, markup, or discount.

   d. In the **Adjustment Amount Source** column, select a CPQ formula field.

      > ⓘ Use this column if you want to make the adjustment amount as a variable that is evaluated in run-time. If you want to provide a static adjustment amount, use the **Adjustment Amount** column.
      > In the **Adjustment Amount Source** column, you can select a CPQ formula field that fetches a value from a numeric or currency type of custom field on the line item object. For more information, see Managing CPQ Formula Fields.

> However, you must select such formula fields that return only numeric values. CPQ applies the adjustment on the list price to derive the base price. For more information on how to use a formula field in creating a price matrix, see Use Case: Using a CPQ Formula Field in Price Matrix.

Use Case: Using a CPQ Formula Field in Price Matrix

In this use case, you will create a price matrix using a CPQ formula field in the **Adjustment Amount Source** column. You want

1. Create a formula field *Auto_Pricing_FormulaPricing_NumericField* on the **Number of Beds** (custom) field on the line item.
2. Create a price matrix with **Matrix Type** as *Dimension*.
3. From the **Dimension** drop-down, select a price dimension on quantity.
4. From **Dimension Value Type**, select *Range* and save the matrix.
5. In the ***<Dimension_Name>*** column, enter 10.
6. From **Adjustment Type**, select *% Discount*.
7. In the **Adjustment Amount Source** column, select *Auto_Pricing_FormulaPricing_NumericField*.

If the Sales Representative enters 20 in the **Number of Beds** field, the value of the formula field is 20. CPQ populates the same value of 20 in the **Adjustment Source Amount** field. CPQ applies 20% discount on the quantity from 1 to 10.

Managing CPQ Formula Fields

You can create a CPQ formula field using a simple formula or using the Conga Numeric Expressions Framework, so that you can use the formula field for configuring prices and adjustments. You can create a formula on any numeric or currency type of custom field on the line item object and use that formula as a dimension while calculating adjustments. Instead of providing the adjustment amount as a static number, you can use a CPQ formula to make the adjustment amount as a variable that is evaluated in run-time. You can use the CPQ formula field while creating a price rule, price matrix, related pricing, and default pricing (as a list price).

For example, you can create a formula field on the **Number of Beds** field on the line item object as follow:

- Name of the formula field: *Auto_Pricing_FormulaPricing_NumericField*
- Business Object: *Apttus_Config2__LineItem__c*
- Field: *Number_of_Beds__c*

You can use this formula field *Auto_Pricing_FormulaPricing_NumericField* in the **Adjustment Amount Source** field while creating a price matrix. If the Sales Representative enters 10 in the **Number of Beds** field, the value of the formula field is 10. CPQ populates the same value of 10 in the **Adjustment Source Amount** field.

> ⓘ To use numeric expressions while creating CPQ formula fields, you must add the Picklist value **Expression**, to the Picklist field **Use Type**. Go to **Setup** > **Create** > **Objects** > **Formula Field (CPQ)** > **Use Type** field.

1. On **App Menu**, click **Conga CPQ Admin** and then click the **CPQ Admin** tab. The new admin console is launched.
2. On the **Pricing** menu, click **Manage CPQ Formula Fields**.
3. Click **New Formula Field**. The DETAILS tab is displayed.
4. Enter the following details and click **Save**.
   a. **Name**: Enter a name for the formula field.
   b. **Business Object**: Select a business object on which you want to create the formula field.
   c. **Description**: Enter a description for the formula field.
   d. **Usage**: Select the type of usage for the formula. Supported values are:
      - **Reference** (default): If you select this option, the **Formula** field is displayed.
      - **Expression**: If you select this option, the **Expression** field is displayed.
   e. **Formula**: Enter a formula as a text.
   f. **Expression**: Click the expression lookup. The Edit Expression dialog box is displayed. Configure a formula using numeric expressions and click **Ok**. For more information, see Configuring Numeric Expressions.

Configuring 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 maintenance 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

1. On **App Menu**, click **Conga CPQ Admin** and then click the **CPQ Admin** tab. The new admin console is launched.
2. On the **Pricing** menu, click **Manage Price Lists**.
3. Click **New Price List**.
4. Select **Price Method** as *Related Price*.
5. Click the **DETAILS** tab. The different fields used for related pricing, such as **Related Adjustment Amount** and **Related Adjustment Type** are displayed.
6. Enter the following details and click **Save**.

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. ⓘ In this field you enter the adjustment amount directly. Use this field if you are not using the **Related Adjustment Amount Source** field.
Related Adjustment Type	Indicates the type of adjustment that you want to apply on the related product. Available values are: *%Discount*, *%Markup, Discount Amount*, *Markup Amount*, *Percentage*, *Price Factor*, and *Price Override*. 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 Extended Price*, *Base Price*, *Extended Price*, *List Price*, *Net Price*, and *Net Unit Price*. In our example, choose *Base Price*.

Field	Description
Related Adjustment Amount Source	Indicates the adjustment amount source that you want to apply on the related product. ⓘ Use this column if you are not using the **Related Adjustment Amount** column. In this column, you can select an adjustment amount that will be fetched from CPQ formula fields created on the required business objects and fields. CPQ fetches the adjustment value from those fields instead of fetching the adjustment amount directly from the **Related Adjustment Amount** column. However, you must select such formula fields that return only numeric values. CPQ applies the adjustment on the list price to derive the base price. In our example, if the formula returns 10, CPQ applies 10% discount to the base price of the related product.

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.

1. Navigate to Related Price List Item (To) related list and click **New Related Price List Item**.
2. 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.

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. 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.
Related Product Group	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 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*.

Currently, the combination with Related Charge Type and Related Price List Item is supported and the combination with Price List Item and Related Price List Item is not supported.

To configure the context for Related Pricing

You can set the related pricing context for Line Items using the 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.

Use Cases: Related Pricing

When **Related Price Scope** = Cart, CPQ calculates the price of the dependent product based on the instances of the primary product in the entire cart. CPQ checks for the primary product in the whole cart.

When **Related Price Scope** = Bundle, CPQ calculates the price of the dependent product based on the instances of the primary product in the same bundle. CPQ checks only for the bundle and its options.

In all these use cases, if you add the dependent product to the cart first, its price is zero because there are no primary products in the cart. If there is only one primary product, the dependent product's price is calculated based on one primary product. If there are two primary products, the dependent product's price is calculated based on two primary products. If you delete primary products, the price of the dependent product is zero.

Use Case 1: Standalone Products

Primary Product 1 = USD 1000

Primary Product 2 = USD 1000

Dependent Product 1 is related to both Primary Product 1 and Primary Product 2 (at 10% Discount)

When **Related Price Scope** = Cart, if you add all three products to the cart, CPQ calculates the price of Dependent Product 1 based on the instances of Primary Product 1 and Primary Product 2 in the entire cart, at 10% discount on the total. Sum of Primary Product 1 and Primary Product 2 = 2000 and 10% discount on 2000 is 200. So the price of Dependent Product 1 = USD 1800 (2000 – 200).

When **Related Price Scope** = Bundle, if you add all three products to the cart, CPQ calculates the price of Dependent Product 1 based on the instances of Primary Product 1 and Primary Product 2 in the same bundle Dependent Product 1 (be it a bundle or option). CPQ looks only for that bundle where Dependent Product 1 exists and there is no instance of

Primary Product 1 and Primary Product 2 in the bundle. So the price of Dependent Product 1 = USD 0.

Use Case 2: Bundle and Option Products

Bundle 1 = USD 1000

Option 1 = USD 100

Option 2 is related to both Bundle 1 and Option 1 (at 10% Discount)

When **Related Price Scope** = Cart, CPQ calculates the price of Option 2 based on the instances of Bundle 1 and Option 1 in the entire cart, at 10% discount. If you add Bundle 1 to the cart once, CPQ calculates the price of Option 2 based on the instances of Bundle 1 and Option 1, at 10% discount. Because there is only one instance of Bundle 1 and Option 1 in the cart, the price of the Option 2 is (1000 + 100) – 110 = 990. If you add Bundle 1 to the cart a second time, the price of Option 2 in both instances is 1980, because there are two instances of Bundle 1 and option 1 in the cart (990 * 2 = 1980).

When **Related Price Scope** = Bundle, if you add Bundle 1 to the cart, CPQ calculates the price of Option 2 based on the instances of Bundle 1 and Option 1 (in Bundle 1), at 10% discount on the total. Sum of Bundle 1 and Option 1 = 1100 and 10% discount on 1100 is 110. So the price of Option 2 = 990 (1100 – 110). If you add Bundle 1 to the cart a second time, the price of Option 2 remains 990 because CPQ looks inside Bundle 1 only (though this is the second instance of Bundle 1).

Use Case 3: Discount Amount

Primary Product 1 = USD 1000

Primary Product 2 = USD 1000

Dependent Product 1 is related to both Primary Product 1 and Primary Product 2 (at 10 Discount Amount)

When **Related Price Scope** = Cart, CPQ calculates the price of Dependent Product 1 based on 10 discount amount on the instances of Primary Product 1 and Primary Product 2 in the entire cart. If Primary Product 1 and Primary Product 2 are not added to the cart, but only Dependent Product 1 is added, the price of Dependent Product 1 is USD – 10.

When **Related Price Scope** = Bundle, CPQ calculates the price of Dependent Product 1 based on 10 discount amount on the instances of Primary Product 1 and Primary Product 2 in the same bundle where Dependent Product 1 exists. There is no instance of Primary Product 1 and Primary Product 2 in the bundle where Dependent Product 1 exists, so the price of the Dependent Product 1 is USD – 10.

Use Case 4: More Scope (Product Group, Product Family, or Charge Type)

On related price, you can define the scope such as product group, product family, or charge type.

Primary Product 1 = USD 1000 (Charge Type = Standard Price)

Primary Product 2 = USD 1000

Dependent Product 1 is related to both Primary Product 1 and Primary Product 2 (at 10% Discount) and Charge Type = Standard Price

When **Related Price Scope** = Cart, if you add Primary Product 1 and Primary Product 2 to the cart, CPQ considers only those products that have the same charge type for calculating the price of Dependent Product 1. In this case, only Primary Product 1 has Charge Type = Standard Price. So the price of Dependent Product 1 is 10% discount of the instances of Primary Product 1 in the cart. 10% discount of 1000 is 100. So the price of Dependent Product 1 = USD 900.

CPQ skips Primary Product 2 even though it is related to Dependent Product 1.

Use Case 5: Custom Group

On related price, you can create a custom group by defining some formula fields.

Primary Product 1 = USD 1000 (PLI_CustomField = 10)

Primary Product 2 = USD 1000 (PLI_CustomField = 20)

Dependent Product 1 is related to both Primary Product 1 and Primary Product 2 (at 10% Discount) - You can create a custom field where (PLI_CustomField = 20)

When **Related Price Scope** = Cart, if you add Primary Product 1 and Primary Product 2 to the cart, CPQ considers only those products that have the same custom field for calculating the price of Dependent Product 1. In this case, only Primary Product 2 has PLI_CustomField = 20. So the price of Dependent Product 1 is 10% discount of the instances of Primary Product 2 in the cart. 10% discount of 1000 is 100. So the price of the Dependent Product 1 = USD 900.

CPQ skips Primary Product 1 even though it is related to Dependent Product 1.

Use Case 6: Locations

If primary product and dependent product have different locations, related pricing does not work. In that case, the net price of the dependent product will be 0. For related price to work, both products must be in the same location.

- Dependent Product 1 is related to Primary Product 1 (at 10% Discount)
- Related Adjustment Applies To: Base Price or Net Price

When you add both products to the cart:

Product	Net Price	Base Price	Location
Primary Product 1	1000	1000	Australia
Dependent Product 1	100	100	Australia

After you change the location of Dependent Product 1:

Product	Net Price	Base Price	Location
Primary Product 1	1000	1000	Australia
Dependent Product 1	0	0	India

> ⓘ If there is a requirement for related price to work even if the location is different for both products, you must use a custom location field.

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.

> When you have applied *Cumulative Range - Line Item* price matrix on a product, if you create ramps for that line item, those ramps will have the matrix applied by default. Ramps will inherit the matrix from that line item. (If you create five ramps, all five ramps will have the matrix.)

To enable price ramps

You must have an existing price list item.

1. On **App Menu**, click **Conga CPQ Admin** and then click the **CPQ Admin** tab. The new admin console is launched.
2. On the **Pricing** menu, click **Manage Price Lists**. A list of existing price lists is displayed.
3. Select a price list and click the **PRICELIST ITEMS** tab.
4. Select an existing price list item, click the more icon (⋮), and click **Advanced**.
5. Click the **MISCELLANEOUS** tab.
6. Select the **Enable Price Ramps** checkbox.
   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.

> ⓘ **Support for Price Ramp on the Cart Grid UI**
> Cart Grid UI shows a Price Ramp icon. When you click the icon, it opens an Auto Ramp wizard if you have enabled auto cascade ramp setting on the PLI.

Enabling Auto Ramp Creation

As a sales representative, you might want to auto generate ramp line creation based on general information, like Start Date, End Date, and Term. In the previous releases, when you created price ramps, every line was created individually. Also, the sales representative might want to set up some custom date to adjust charge lines for customers. To address the first scenario, check the **Enable Auto Ramp Creation** under the Miscellaneous tab on a Price List Item page to enable automatic price ramp on the other charge type.

Consider the following scenarios:

> ✅ **Scenario 1**

Consider a scenario where a product has two charge types. When using price ramp on any one of the charge types, the price ramp should be created for the other charge type too. In the previous releases, the system could create a price ramp for only one charge type with the help of an API. This issue has been addressed in the current release, with the introduction of **Enable Auto Ramp Creation** in the Price List Item under the Miscellaneous tab to enable automatic price ramp on the other charge type.

✅ **Scenario 2**

As a sales representative, you might want to auto generate ramp line creation based on general information, like Start Date, End Date, and Term. Also, you might want to set up some custom date to adjust charge lines for customers. To auto generate ramp line creation, check the **Enable Auto Ramp Creation** in the Price List Item page under the Miscellaneous tab to enable automatic price ramp on the other charge type. To set up custom date, you can set the start date, end date, terms and frequency period during ramp creation. You can select the frequency period as 'Monthly', 'Quarterly, 'Yearly' or as 'Custom'. If you select 'Custom' as frequency period, then mention the number of ramps in the text field. The start date, end date, and terms are available when added to the cart. When you click 'save', all the lines are created at the same time. When you select 'Monthly', 'Quarterly, or 'Yearly', the system creates a number of ramps based on Start Date, End Date, and Term. If 'Custom' is selected as the frequency, the system creates a number of lines based on the number of ramps.

To enable Auto Ramp creation

You must have an existing price list item.

1. From CPQ Console, go to your target price list.
2. Under the related list for *Items*, click **Edit** for the required price list item. The Price List Item Edit page is displayed.

3. On the **Miscellaneous** tab, select the **Enable Auto Ramp Creation** check box.

This auto generates the ramp line creation based on general information, such as Start Date, End Date, Term, and Frequency period.

Enabling Tiered Pricing

Price Tiers enable an end user to view or set up tiers on 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 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.

> ⓘ 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 **App Menu**, click **Conga CPQ Admin** and then click the **CPQ Admin** tab. The new admin console is launched.
2. On the **Pricing** menu, click **Manage Price Lists**. A list of existing price lists is displayed.
3. Select a price list and click the **PRICELIST ITEMS** tab.
4. From **Price Type**, select *Usage*.
5. From **Price Method**, select *Tiered Rate*. If Tiered Rate is not available as a choice, you can add it to the custom field object.
6. To enable a user to modify a usage tier.
   a. Click the more icon (⋮), and click **Advanced**.
   b. On the **MISCELLANEOUS** tab, select the **Is Usage Tier Modifiable** checkbox.
7. 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 pricing cart page.

You can create or modify tiers for pricing on 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 Conga 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*.

   > ⓘ 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 the price list, price list item, price matrix, price rule, a 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. On **App Menu**, click **Conga CPQ Admin** and then click the **CPQ Admin** tab. The new admin console is launched.
2. Click the menu icon(⋮) on the main menu bar at the top.
3. Click **Criteria Maintenance**. The Criteria Maintenance popup is displayed.
4. 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%.

> ⓘ **Incremental Update for Criteria Maintenance Job**
>
> The Criteria Maintenance Jobs are incremental instead of rebuild so that existing records are not impacted.
> - When you execute the constraint field maintenance job, CPQ generates new constraint field records as an insert operation.
> - When you execute the pricing fields maintenance job, CPQ generates new pricing fields records as an insert operation.
> - When you execute the expression maintenance job, CPQ generates new expression records as an insert operation.
> - Existing records remain as is.
> - CPQ does not have downtime for quotes with existing records.

Configuring Contract Pricing

As a Pricing Administrator, you can store contracted prices in CPQ. The contract pricing feature offers a way to negotiate a price for a product that is not being purchased in the

current quote. CPQ uses a set of agreed upon prices for a customer for all the subsequent transactions. CPQ provides the following options to implement contract pricing.

CPQ supports contracted price lists specific to customers. Each customer might have different pricing information. Whenever customers order products or services with the same contracted price list, agreed prices will be given to them.

CPQ can maintain contract price lists (customer specific price lists) to keep a track of negotiated prices for each customer. The customer provides a set of agreed upon prices using an offline process or mechanism. You must set up this information as a price list using the standard CPQ capability. After you set up this price list, any new quote or order coming into CPQ will specify the customer specific price list as the price list on the quote or order.

When to use Contract Price Lists?

- Low customer base: Customer specific price lists make perfect sense, when the number of customers is not huge, as creating customer specific price lists creates a lot of data in CPQ and over a period of time, with changes and renewals, the number can grow exponentially.
- Low number of SKUs in catalog: You must use customer specific price lists when the number of products being negotiated with a customer is of the magnitude of less than 1000 products. If the product catalog contains a large number of products, customer specific price lists do not offer the flexibility as price programs. In such cases, price programs make better sense.
- Better suited for standalone products: If the product catalog contains a large number of highly complex configurable products, customer specific price lists do not make sense as each product can be configured a lot of different ways.

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 inherit the contract price list from the bundle. If an option is included in two different bundles in a quote, CPQ adds the option twice to the Contract Price List (resulting in two contract price list items). CPQ then applies pricing on the quote based on the selection of the bundle and options, and calculates the price of the option from the price list item that has relation to the current bundle under which it is being configured.

> ⓘ The contract pricing for bundles works fine for a single-level bundle and same option, however it does not work for same sub-bundles.
> Contract pricing does not work for when an option is part of different bundles and inherit price from different price lists.

Configuring Contract Price Lists

To enable contract pricing in CPQ, you must create a price list and store the contract prices. You must set the Type = Contract and associate the price list a *Contract number*.

To configure a contract price list

You must have an existing price list. You must have an existing contract number or agreement number.

1. On **App Menu**, click **Conga CPQ Admin** and then click the **CPQ Admin** tab. The new admin console is launched.
2. On the **Pricing** menu, click **Manage Price Lists**. A list of existing price lists is displayed.
3. Click **New PriceList**.
4. From the **Type** drop-down list, select Contract. The quote derives its pricing from a contract.
5. In the **Contract Number** field, enter a contract number or an agreement number.
6. Enter values in the required fields.
7. Click **Save**.

A contract price list is created.

Defining Contracts to be Used in a Quote

You must define (explicitly) which contracts to be used in quotes.

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. You must create a custom button and add the parameter **cntrNbr_1** to the button formula. CPQ determines the contract pricing for quotes based on the parameter in the cart.
- Quote Line Item Level: Sales Representatives can add the contract number at the quote line item level. The **Contract Numbers** field is available on the Line Item record. This applies contract pricing on specific line item only. This parameter is useful when users want to use different contracts for different products. They can apply contract pricing on specific set of products or some other criteria based on condition and not on all products.

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.

Prerequisites

- The contract must be active.
- Contract has the contract number populated and a Price List associated with it (the Related List will have the Price List).
- The contract number on the price list must be the same as the contract number on the Contract.

To define a contract to be used in quote

You must be in the Quote/Proposal object.

1. In **CPQ Console**, 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 Step 4. Establish field-level security page, make the necessary changes and click **Next**.
9. On 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="&Contract_price_list_1_
_c,IMAGE ("/resource/Apttus_QPConfig__Button_Configure", "Configure Products"),"_self
"), NULL)
```

Defining Multiple Contracts to be Used in a Quote

Having multiple contracts on one quote

CPQ supports passing multiple price lists on a Quote/Agreement/Order to see both Non-Contracted and Contracted Products. Sales Representatives may want to display contracted and/or non-contracted products. In this case, you can pass a list of comma-separated contracted price lists (sequence is important because CPQ follows the same sequence as you pass it).
CPQ uses the default price list whenever a user creates a proposal. CPQ creates an account-specific price list when a user creates an agreement. CPQ tries to search if any account specific price list available or not. If the account-specific price list is available, CPQ uses it; if the account-specific price list is not available, CPQ uses the default price list.

> ⓘ   A proposal can have the default price list but the price list item on line item record displays the applied price list item information.

Use Case

A customer has multiple accounts based on locations.

- ABC New York
- ABC California
- ABC USA Operations (parent account)

If you are quoting from ABC New York and a product's pricing is available in ABC USA Operations, the pricing should be selected from ABC USA Operations and not from ABC

New York. If there is parent account for ABC USA Operations, then look for applicable parent account pricing.

Solution

There are different price lists, that is, ABC New York, ABC California, and ABC USA Operations with contract number populated on each price list.

- ABC New York Price List contract number is 23456789.
- ABC California Price List contract number is 12345678.
- ABC USA Operations Price List contract number is 01234567.

You must set the parameter **cntrNbr_1** in the correct sequence using the **Configure Products** formula button as follows:

```
IF ( LEN( Apttus_Config2__PriceListId__c ) > 0 , HYPERLINK("/apex/
Apttus_Config2__OrderConfiguration?id=" &Id &"&cntrNbr_1=23456789&cntrNbr_2=12345678"&
 "&flow=NGDefault", IMAGE("/resource/Apttus_Config2__Button_Configure", "Configure
Products"),"_self"), NULL)
```

You can provide multiple contract numbers separated by ampersand (&). If the product is found in contract number 23456789, pricing will be applied from ABC New York Price List. If not, CPQ looks for another contract number 12345678. If pricing is found in contract number 12345678, pricing will be applied from ABC California Price List. If not, pricing will be applicable from the default price list.

Configuring Contract Pricing through Price Agreements

CPQ can store pricing information when users accept a proposal or some other event occurs, as per client requirement. CPQ can retrieve the same pricing information when the same user creates a proposal for the second time or updates the same price list in case of agreement amendment. Many customers prefer old pricing during renewals or amendments.

When the user creates a quote with Intent = Price Agreement, finalizes the quote, and accepts it, CPQ automatically creates a contract price list with the products from the quote, according to the prices negotiated in the quote. The contract price list is a subset of the standard price list. CPQ assigns a contract number to the new price list generated. The user can then use the contract number and pass it to the future quotes for using the

contract pricing feature. The prices in the new quote will be retrieved from contract price list and those prices are the negotiated prices of the original quote.

For more information, see Creating Quotes from Opportunities.

Configuring Pricing Profiles and Pricing Batch Size

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, CPQ 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
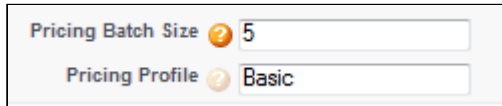
1. In **CPQ Console**, go to **Setup** > **Develop** > **Custom Settings**.
2. Click **Manage** for **Config System Properties**, and click **Edit**.
3. Select **Defer Pricing**.

   > ⓘ To use **PricingProfile** and **PricingBatchSize**, **DeferPricing** must always be selected.

4. For **Pricing Profile**, type *Basic* or *Advanced*.

   > ⓘ If this field is left blank, the default value is Advanced.

5. For **Pricing Batch Size**, type a value.

   | Pricing Batch Size ❓ | 5 |
   | Pricing Profile ❓ | Basic |

   > ⓘ 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.

Configuring Email Notifications for Apex Governor Limit Warning

When 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 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.
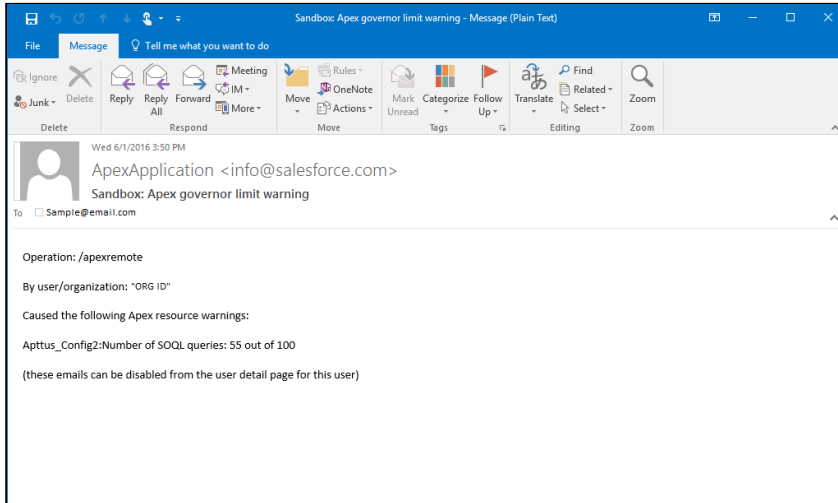
Managing Cost and Profitability (Cost Breakdown)

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.
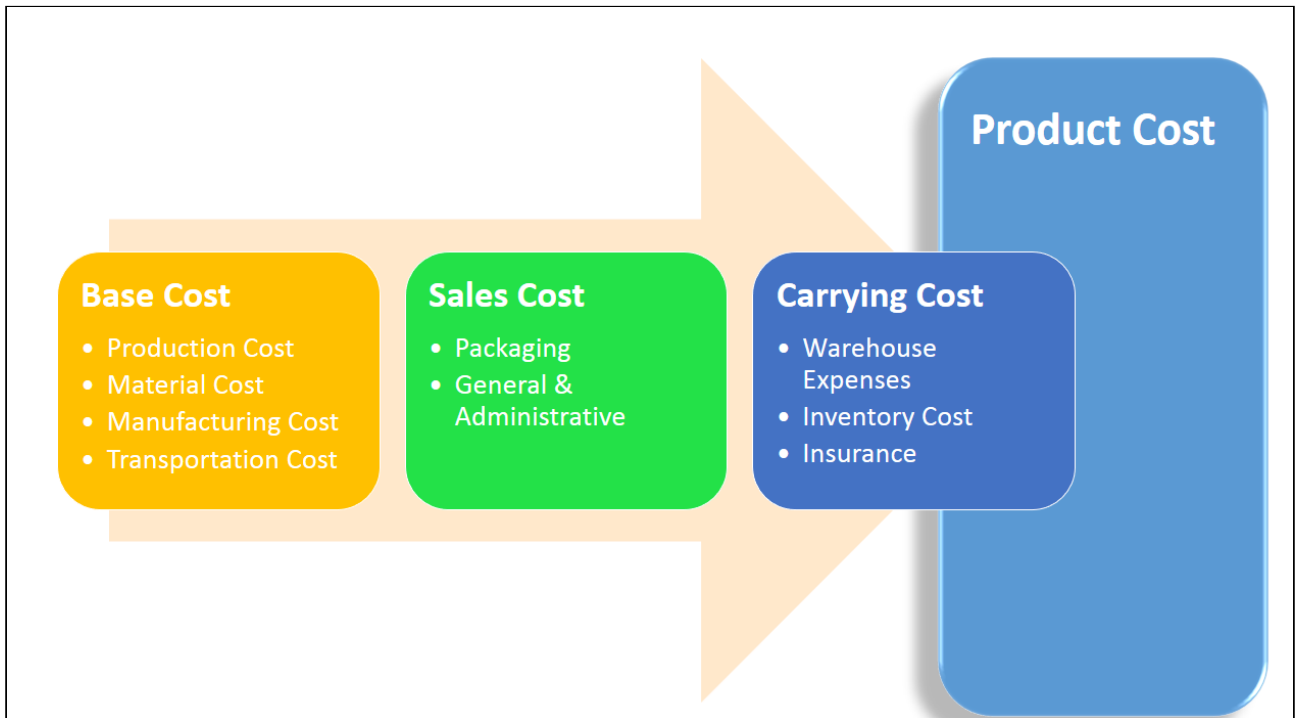
The cost models, as you can see in the above illustration, are nothing but the structure of cost elements and sub-elements participating in deriving the final cost. The cost hierarchy gives you the granular level of details regarding the various aspects of product costing. With CPQ, you can create complex 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.

Term	Description
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.
  For more information on how you can configure and store the granular costs associated with your product, see Storing the 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. For more information on how you can create a Transfer Price mapping, see Configuring Transfer Pricing.
- **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 more information, Analyzing Quotes.

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 Cost Models

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.
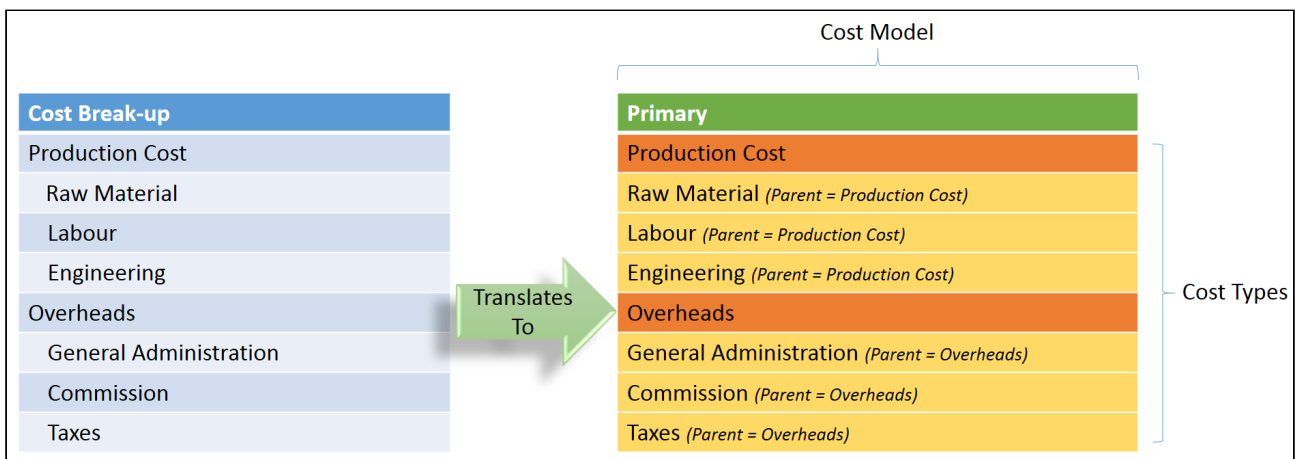
1. In **CPQ Console**, go to **+ (All Tabs)** > select **Cost Models**.
2. Click **New**. The New Cost Model page is displayed.
3. In the **Cost Model** field, enter a name for the cost model.
4. Click **Save**.

Your cost model is created and you can start adding cost types.

Define Cost Types

Cost Types correspond to individual cost entities. You can define *parent* for a Cost Type thereby creating Cost Type hierarchies.

1. From the Cost Types related list, click **New Cost Type**.

2. Enter the following details and click **Save**:

Field	Description
Cost Type	The name for a 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 selected, allows you to override costs on the **Analyze Quote** page. On clicking Reprice, the base price is recalculated with the new costs.
Currency	The Currency specific to this Cost Type.
Hide Child Cost Types in Price Waterfall	If selected, allows you to hide a cost category and its children from being displayed on the Analyze Quote page.

Associating Cost Models with a Price List

You can associate one cost model with multiple price lists. However, one price list can have only a single cost model associated with it.
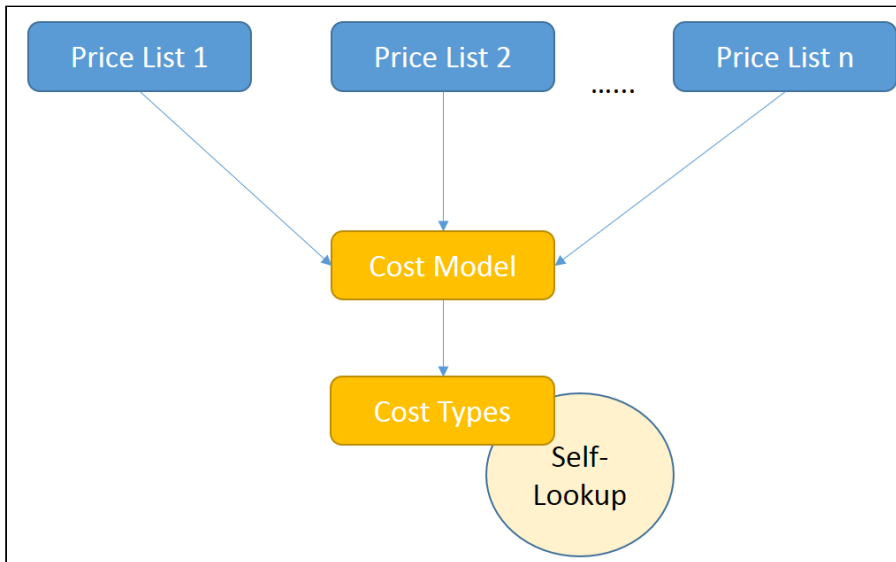
To associate a cost model with a price list

1. On **App Menu**, click **Conga CPQ Admin** and then click the **CPQ Admin** tab. The new admin console is launched.
2. On the **Pricing** menu, click **Manage Price Lists**.
3. Select the required price list
4. In the Cost Model field, search and select a cost model of your choice.
5. Click **Save**.

Configuring the 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
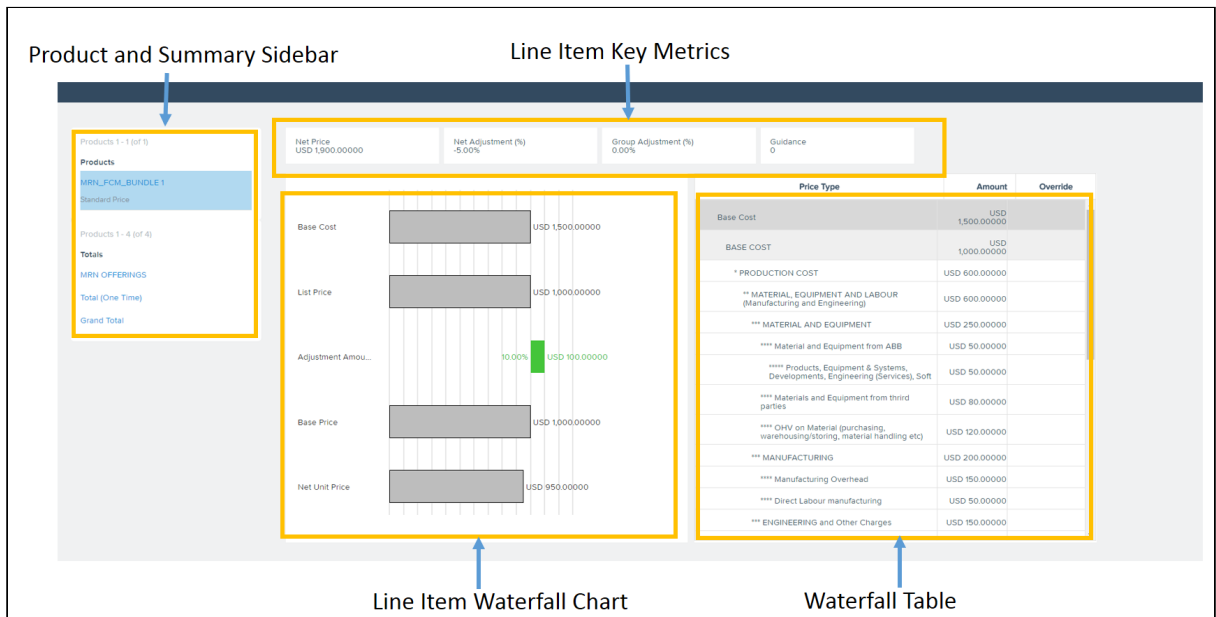
You can opt to display the fields of your choice in the **Line Item Key Metrics** and **Line Item Waterfall Chart** screen elements for both **Line Items** and **Summary Group options**.

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 the 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)

- Conga X-Author™ For Excel Document Generation (1.0)
- Conga 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.
3. For information on how to create an app, see Creating Apps with Quick Apps.
4. Ensure the Cost Type Label and Cost ID matches with the Cost Names specified in the org.
5. On the App, add Parent Code (from the Salesforce URL) for the child Cost Items.
6. **Validate** and **Save** the App.
   Go to your Salesforce org and access Apps tab to see your app.
7. Edit the app and select **Activated** to activate it.
8. Create a **Price List** and associate it with the Cost Model.
9. 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("/apex/
Apttus_QPConfig__ProposalConfiguration?id=" &Id
+"&flow=NGDefault"&"&useAdvancedApproval=true&useDealOptimizer=true"&"&prestoApp
pId=a5t2C000000GnBe",
IMAGE("/resource/Apttus_QPConfig__Button_Configure", "Configure Products"),"_se
lf"), NULL)"
```

10. Go to **+ (All Tabs)** > **Apps**, open your app and copy the App Id from the URL.
11. Create a Quote, add the product to the cart and click Reprice. To see the price breakdown Click **Analyze Quote** button.

Configuring 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
Target Field	Requested_ManufacturingNetPrice__c
Target Object	Apttus_Config2__LineItem__c

Configuration

1. In **CPQ Console**, go to **+ (All Tabs)** > select **Cost Models**.
2. Select the required cost model.
3. From the Transfer Prices related list, click **New Transfer Price** to create a Transfer Price mapping.
4. Enter the following details and click **Save**.

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.

Configuring Unit of Measure (UOM) and Frequency Conversion Rate

Unit of measure (UOM) is a way to quantify resources. It is a key factor in pricing and discounting the products on the cart. With the new object called **Frequency/UOM conversion rate**, you can define a set of unit of measures for your product families. Products

under the same family can be priced differently, based on its unit of measure. Depending on the available unit of measures for the product family, the sales reps can define different discounts for different UOMs for the same product.

The current design requires all products under the same product family to share the same UOM conversion rates. So an implementation tip is to use product category to group products by business needs, but use product family to group products that share the same conversion rates. For example, different types of coffee can be grouped under a single category called Coffee, but if you want to apply the same conversion rate to dark roast and mild roast coffee products, you can create a new product family called Roast Coffee.

Frequency/UOM conversion rate is a new object, along with fields such as **Conversion Factor** and **Type**. Depending on the type of conversion type, you must define the field values for either the Frequency conversion or UOM conversion. You can also control whether to apply these conversions on a product family and product category.

You can only define UOMs and UOM conversion rates for product families; one product can exist in multiple categories, but only in one product family.

Use case: Creating UOM and conversion rate for products

Description: This use case describes how to create frequency conversion rates and unit of measurements for pricing the products on the cart. You might use this functionality differently, depending on your business case.

Suppose you are a coffee vendor and want to sell your product by considering the customer requirements in terms of kilograms and boxes. As a system administrator, you can set up the type of measurement as kilogram and box, and define the conversion rate for these measurements.

In the following example, the sales administrator creates a conversion type for the coffee product.

The above setup represents the following UOM conversion rates for the product family Coffee

1 Box = 10 Each

1 Kilogram = 20 Each

1 Case = 5 Each

1 Kilogram = 2 Box

Pricing admin can create product list for any products under the product family Coffee and use above UOMs captured in the UOM conversion table.

For example, the pricing admin can create prices list in the following UOMs:

Table 1

Product	Price	UOM
Dark Roast	$10	Each
Dark Roast	$95	Box
Dark Roast	$50	Case
Medium Roast	$10	Each
Medium Roast	$90	Box
Medium Roast	$45	Case

And the sales rep can create the following manual adjustments on the cart:

Table 2

Product	Discount	UOM
Dark Roast	$2 off	Each
Dark Roast	$23 off	Box
Dark Roast	$12 off	Case
Medium Roast	5% off	Each
Medium Roast	8% off	Box
Medium Roast	5% off	Case

In the cart, the pricing engine converts the quantities from the Pricing UOM to the Selling UOM and the discounting UOM to the Selling UOM.

To create a Frequency Conversion Rate

Perform the following steps to create a new record of frequency conversion rate:

1. In **CPQ Console**, click ⊞ (All Tabs) and search for **Frequency/UOM Conversion Rates**.
2. Click **Frequency/UOM Conversion Rate** tab and click **New** to create a new record.
3. The record is divided into 4 sections and the fields in each section are explained below:

Information	
Field	**Description**
Type	Defines the currency metric type. Available values are *Frequency* and *Unit Of Measure*. Depending on your selection, you must enter the values in either Frequency or UOM Conversion table.
Product Family	Defines the Product family to which you want to apply the conversion rate. This helps you filter the group of products on which you want to apply the conversion rate. Ensure that you have created a product family record so as to see it populated in this picklist.
Product Category	Defines the Product category to which you want to apply the conversion rate. This helps you filter the group of products on which you want to apply the conversion rate. Ensure that you have created a product category record so as to see it populated in this picklist. In our example, the product category is *Coffee*.

Frequency Conversion > applicable for Type = *Frequency*	
Field	**Description**
From Frequency	Defines the starting frequency for a conversion rate from which the rates are valid.
To Frequency	Defines the ending frequency for a conversion rate until which the rates are valid.

UOM Conversion > applicable for Type = *Unit of Measure*	
Field	**Description**
From UOM	Defines a unit of measure that can be used as a source for rate conversion. For example, if you want to measure a coffee box in terms of Box to Kilogram conversion, use *Box* in **From UOM** and *Kilogram* in **To UOM** field.
To UOM	Defines a unit of measure that can be used as a destination for rate conversion.

Rate	
Field	**Description**
Conversion Rate	Defines the rate at which you want to convert the product. In our example, for UOM conversion from Box to Kilogram for 1 box = 2 kilograms, enter 2 in this field.

4. Click **Save**.

    You must create these records for each UOM that you want to configure. In our example, you must create one record for each row of Table 1.

Configuring the Currency Rounding Mode

This section provides information on configuring currency rounding in pricing calculations. This feature rounds adjustments before calculating the base price.

Prerequisites

Enable Custom Rounding must be True. For more information, see *Config System Properties* in Configuring Custom Settings.

To configure currency rounding

Perform the following steps to configure currency rounding.

1. In **CPQ Console**, go to **Setup** > **Develop** > **Custom Settings** > **Config System Properties** > **Manage**.

2. Click **Edit** next to **System Properties**.
3. In the **Rounding Mode** field, enter one of the following values.

Value	Description
UP	Rounds the currency to the next number. For example, 21.2 is rounded to 22.
DOWN	Rounds the currency to the previous number. For example, 21.8 is rounded to 21.
HALF_UP	Rounds the currency to the next number if the decimal is equal to or greater than 5. For example, 21.5 to 21.9 is rounded to 22.
HALF_DOWN	Rounds the currency to the previous number if the decimal is equal to or smaller than 5. For example, 21.1 to 21.5 is rounded to 22.
HALF_EVEN	Rounds the currency to the nearest even number. For example, 23.5 is rounded to 24 and 22.5 is rounded to 22.

4. Click **Save**.

Managing Currency Conversion

You can set the exchange rate between the currency of your organization and another currency for a certain period.

> ⓘ If the Sales Representative uses the conversion rate on a date that is not within the defined period for currency exchange rate, CPQ displays an error message on the Cart.

Prerequisites

1. You must set the value of the **useAdvancedCurrency** parameter to *True* to enable dated currency conversion.
2. A price list should be created where the **currency** field should be associated with the specific currency value based on your requirement for the currency conversion.

To manage currency conversion

1. On **App Menu**, click **Conga CPQ Admin** and then click the **CPQ Admin** tab. The new admin console is launched.
2. On the **Pricing** menu, click **Manage Currency Conversion**. The Dated Currency Exchange Rates page with a list of existing exchange rates is displayed.
3. Click **Add New Conversion**. A new item is added at the end of the list.
4. Enter the following details:

> ⓘ If the exchange rate is already available between the same currencies and you enter the exchange rates between the same currencies for the same period, CPQ displays an error message.

Field	Description
Currency From	CPQ selects the source currency automatically based on the currency you have configured for your organization.
Currency To	Select the target currency to which you want to convert the source currency.
Exchange Rate	Enter the exchange rate between two currencies.
Rate Type	Select the rate type for currency conversion. Currently, Period End Date is supported.
Start Date	Select the start date from when the currency conversion must be effective.
End Date	CPQ enters the End Date automatically after you provide the Start Date for the next entry for the exchange rate between the same currencies. For example, you provided the first rate between USD and EURO from 01/01/2020. Now when you provide the rate between the same currencies from 01/04/2020, the End Date for the previous entry is taken automatically by as 31/03/2020.
Currency	

5. Click **Save**.

6. Click the more icon (⋮) and select **+ Add** to add a new conversion or select **X Remove** to delete an existing conversion.

Managing Price Waterfall

Price Waterfall is the industry standard for Price and Margin calculation. Price waterfall defines a sequence to handle different prices with the target of reaching the net price.

> ⚠ This feature is available only in the TurboPricing flow. For more information, see About TurboPricing.

A well-defined Price Waterfall provides transparency and high visibility of pricing calculation and helps protect margins and avoid incorrect pricing.

Price Waterfall comes with features for transparency and high visibility in pricing calculation and protecting margins, avoiding incorrect pricing. In addition, Price Waterfall defines a sequence to treat different prices with the end goal of arriving at the net price charged to the customer.

Price Pipeline enables administrators to configure the price execution flow and set the groundwork for Conga's Price Waterfall feature to display detailed pricing steps for end-users to visualize. Price waterfall is a tool that provides users with detailed information on how much revenue the company is generating from a transaction and the pocket margin of that transaction. The waterfall is typically represented in a column chart consisting of critical price or margin points and any adjustments that were applied to the list price to arrive at the final price or margin.

The advantages of the price waterfall functionalities are as follows:

- Provides Sales Representatives with the ability to drill down on any price point visible to them so that they can make better decisions and better communicate the deal to customers giving them more leverage during a negotiation.
- Provides Approvers and Deal Desk with a complete picture of the different price points, preapproved and non-standard adjustments, and costs to better analyze the health of the deal.
- Provides Cost Accountants and Controllers' ability to drill down on the cost (if cost is included as part of the Price Pipeline) to ensure that the true profitability is calculated for the deal to ensure the company's bottom line goals are met.

Configuring Price Waterfall

To configure the price waterfall, perform the following steps:

St ep	Task	Description
1	Define Pricepoints	Create new price points as picklist values in the "Price Points" field in the Price Rule object in Salesforce so that these price points can be used when defining any price pipeline.
2	Configure Price Pipeline	Define price points and configure price pipelines.
3	Configure a Price Waterfall to a Price Pipeline	Configure a price waterfall to a price pipeline.

The following chapters describe the steps to configure price waterfall:

- Defining Price Points
- Configuring a Price Pipeline
- Configuring Price Pipeline Ruleset

Defining Price Points

For creating new price points, Admin must add them as new picklist values in the "Price Points" field in the Price Rule Object in Salesforce.

To add new price points as picklist values:

1. Go to **Setup** > **App Setup** > **Create** > **Manage your custom objects** > **Price Rule**.
2. Under the **Custom Fields & Relationships** section, click **Price Points**.
3. Under the **Values** section, click **New.**
4. Enter new price points in the textbox and click **Save.**

Configuring a Price Pipeline

A price pipeline is a sequence of price points.

You can add multiple price points in a price pipeline definition, arrange them in a sequence to define the price pipeline, and map one price as net price. Base Price and List Price are mandatory price points, and hence those will already be added to every price pipeline by default. When the pricing engine is pricing a line item, it calculates the per-unit amount associated with each price point and the gross amount (amount incorporating the quantity and term) if the transaction is a quote or order. The price point definition describes how it should be calculated and whether it is treated specially (for example, identifying a particular price point as the Net Price point for integration with downstream systems such as billing.)

Pricing administrators or CPQ administrators define price points and configure price pipelines. The price pipeline definition is further used in the Price Waterfall chart to determine whether certain features (such as manual adjustment to any price point) are allowed to the end-user or not. For more information, refer to Defining the Price Points.

To add a price pipeline:

1. On **App Menu**, click **Conga CPQ Admin** and then click the **Admin UI** tab. The new admin console is launched.
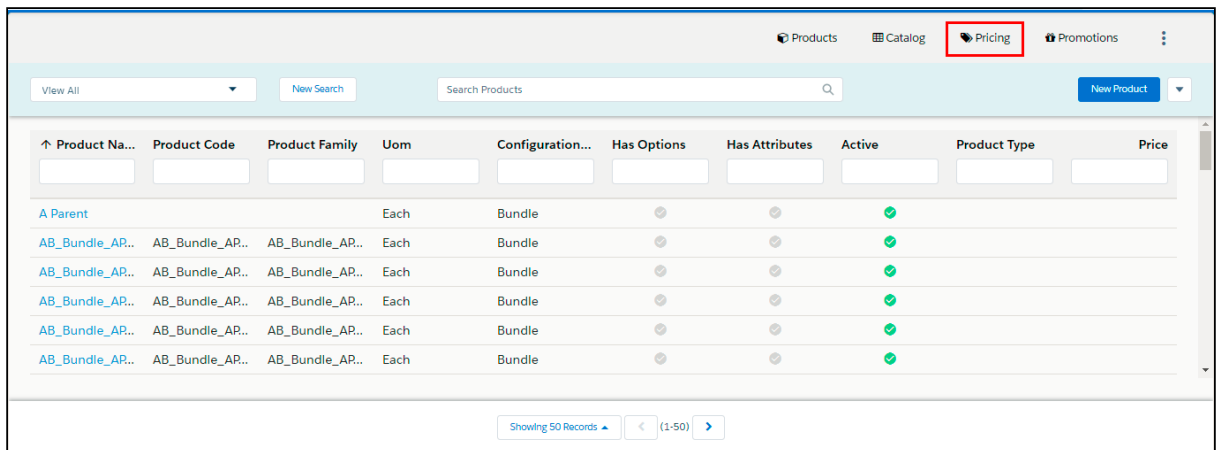
2. Select **Pricing** > **Manage Price Pipeline**.
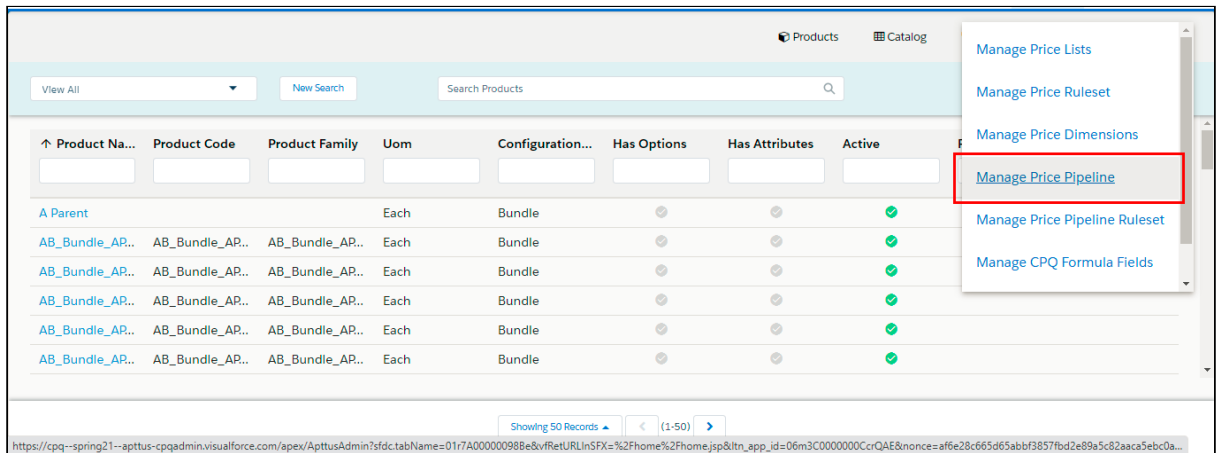
The Manage Price Pipeline page is displayed.
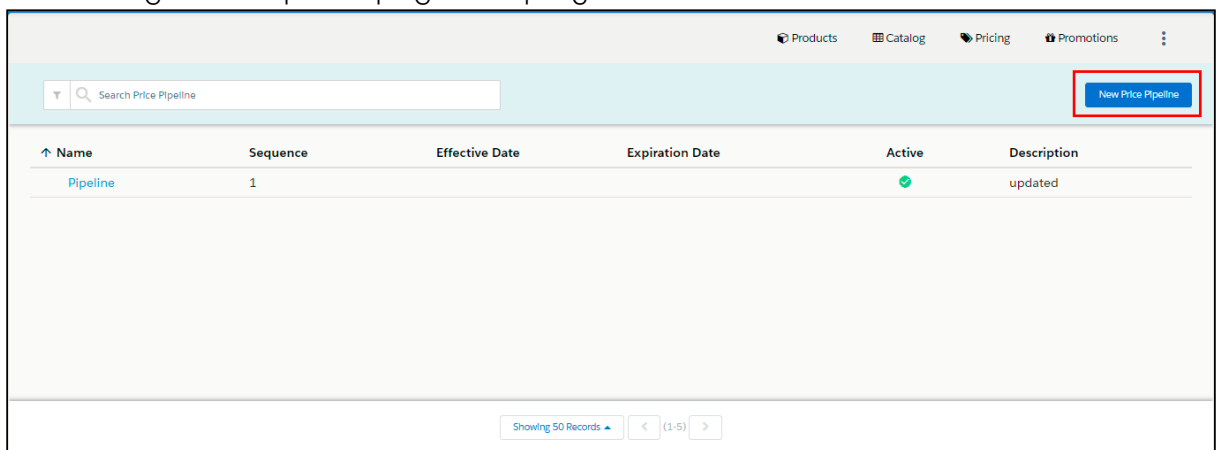
3. Click **New Price Pipeline** and update the following fields:

Field Name	Description
Price Pipeline Name	Name of the price pipeline
Sequence	The sequence allows you to determine the correct price pipeline to apply if multiple price pipelines match the criteria for a line item. The lowest matching sequence will have the highest priority.
Is Active	Slide the slider to enable the price pipeline
Description	Description of the price pipeline
Effective Date	Enter a date to associate an effective start date with the price pipeline

Field Name	Description
Expiration Date	Enter a date to associate an effective end date with the price pipeline
Pipeline Criteria	Define the criteria for the price pipeline. The price pipeline is used based on the criteria defined by the user. If the given criteria are not met for any line item, then the price waterfall chart is not created. Define the criteria for Line Items and Product Attributes as follows. • For Line Items, click the icon > Line Items > Add New Criteria > enter the desired Field (Select the field that must be used for the adjustment), Operator, and Value > Save. • For Product Attributes, click the icon > Product Attributes > Add New Criteria > enter the desired Field (Select the field that must be used for the adjustment), Operator, and Value > Save.
Define Price Points:	Define price points as follows. The list price and Base Price are set by default. a. Click the text box in the name column and select the desired price point from the list. If the price point is not available, create a new price point. b. Is Modifiable: This allows you to modify the price point on the Waterfall page. For example, If you enable the Is Modifiable option for a price point such as *Invoice Price*, the sales representative can only modify the *Invoice Price* of the product by adding manual adjustment to it on the Price Waterfall chart. c. Is this Net Price: This allows you to make a price point as a net price. Move the slider to enable the option. Any manual adjustments that the user performs on the cart appear on the waterfall chart against this price point (which is marked as the net price). d. Click the '+' icon to add new price points.

4. Click **Save.**

ⓘ You must set at least one Price Point as a Net Price.

To Modify a Price Pipeline

1. On **App Menu**, click **Conga CPQ Admin** and then click the **Admin UI** tab. The new admin console is launched.
2. Select **Pricing** > **Manage Price Pipeline**.
3. Click the required Price Pipeline > Modify the details and click **Save.**

Configuring Price Waterfall to a Price Pipeline

After configuring the price pipeline, you must configure the price waterfall to a specific price pipeline.

To configure a price waterfall:

1. In the **App Menu**, click **Conga CPQ Admin** and then click the **Admin UI** tab. The new admin console is launched.
2. Select **Pricing** > **Manage Price Pipeline**.
3. Click the required Price Pipeline > Click the **WATERFALL SETUP** tab.
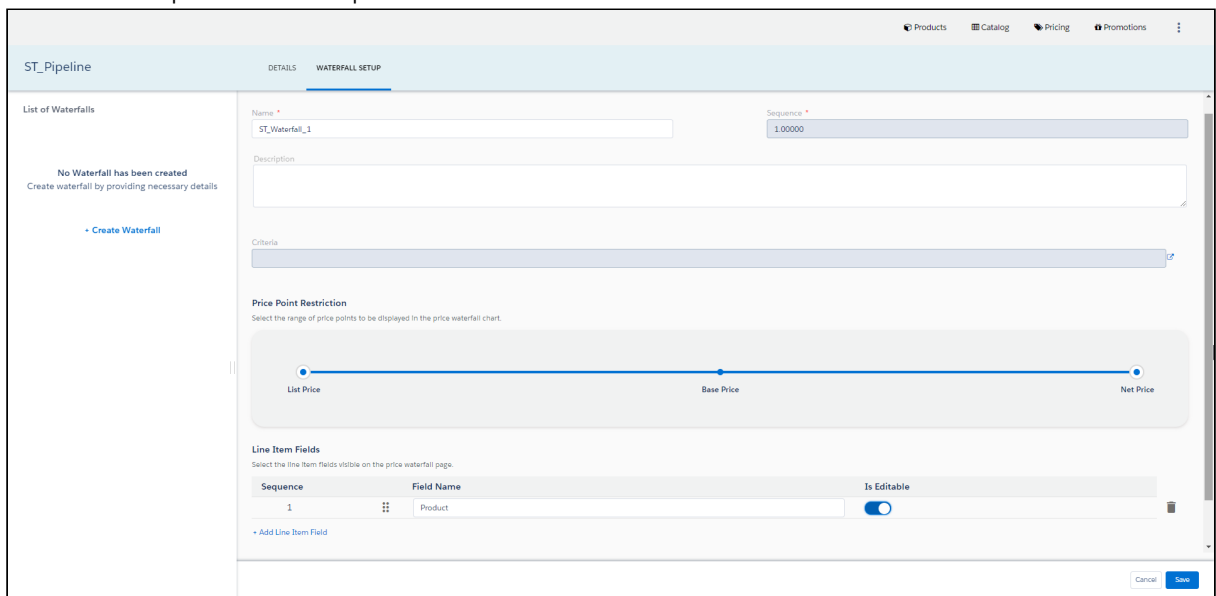
4. Update the following details and click Save.

Field	Description
Name	Name of the price waterfall.
Description	Provide a description.

Field	Description
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 the icon to fill in your criteria.
Price Point Restriction	Set the visibility restriction on price points. This option allows you to control the visibility of price points for the user.
Cart Line Items Field	a. Click the text box in the name column and select the desired Line Item field from the list. b. Is Editable: This enables you to modify the price point on the *Waterfall* page. For example, if you enable this option for the cart line item *Quantity,* then the sales representative can modify the quantity of the product on the waterfall chart and apply it to the price. Move the slider to enable the Is Editable option. c. Click the '+' icon to add new field names. d. Click **Save.**

Configuring Price Pipeline Ruleset

Price Pipeline Ruleset allows pricing adjustments across a range of products through price rules. Price Pipeline Ruleset manages adjustments to specific price points, based on particular line item criteria or product attribute criteria. Price Pipeline rulesets are very similar to standard price rulesets, except that these contain price pipeline rules instead of the standard price rules. A price pipeline ruleset can contain one or more price pipeline 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.

# To create a price pipeline ruleset

You must have an existing price list.

1. On **App Menu**, click **Conga CPQ Admin** and then click the **CPQ Admin UI** tab. The new admin console is launched.
2. On the **Pricing** menu, click **Manage Price Pipeline Ruleset**.
3. Click **New Price Pipeline Ruleset**.

4. Enter details in one or more of the following fields, as required:

Field	Description
Price Pipeline Ruleset Name	Enter a mandatory price pipeline 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
Is Active	Select this to set the ruleset as active.
Enable Currency	Select this to enable currency for the Price Pipeline Ruleset.
Currency	Select the currency for the Price Pipeline Ruleset. ⓘ If the Price Pipeline Ruleset and a line item have different currencies, CPQ applies the currency conversion rate (defined at **Setup** > **Company Profile** > **Manage Currencies**) when applying the rule adjustment.

Field	Description
Short Description	Enter a description for the ruleset.
Effective Date	Not required as long as the Price Pipeline Ruleset is Active, but can be used for promotional rules, and more.
Expiration Date	Select an expiration date.
Price List	Select a price list. The ruleset will only source product prices with the selected price list and then apply adjustment criteria.
Product Category	Select a category. The ruleset will only source product prices within the selected category.
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. The supported values are: • *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 the adjustment is displayed in the Totals tab in the shopping cart. • *Aggregate:* Selecting this enables the **Application Method** field.
Application Method	This field is enabled if you selected *Aggregate* from the **Application Level** drop-down list. 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. The supported values are: • *Apply to Line Items*: Applies the adjustment to line items on the cart. • *Create Summary Lines*: Applies the adjustment as a summary line in the Totals tab on the cart.
Enable Date Range	Selecting this enables you to set the effective date and expiration date on the rule entry.

Field	Description
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 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.
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 the ⬀ icon to fill in your criteria.

5. Click **Save**.

By filling out these criteria, the source products and prices are then designated and the ruleset can apply adjustments through price rules.

Creating Price Pipeline Rules

A *Price Pipeline Rule* enables you to provide target adjustments to specific price points in the price waterfall chart. Although the standard price rules are applied to the *Base Price*, you can now target any custom price point that you want by using the price pipeline rule. When defining any price pipeline rule, you must associate at least one price point to a price pipeline rule. These rules are will be executed (subject to fulfilling other criteria) and shown as adjustments specific to only those price points in the price waterfall chart

You can associate multiple pipeline rules to a price pipeline ruleset or through to a product. Price matrices can be defined in price rules as well as conditional price pipeline rules (rules that use Boolean logic such as AND/OR). The price pipeline rule can be dimensional, which uses a similar concept to Price Matrices or Conditional, that relates to fields for pricing adjustments.

One ruleset can have multiple rules and is evaluated in order of the rule sequence. A rule can be dimensional or conditional. A price pipeline rule determines the actual price adjustment made.

To create a price pipeline rule

You must have an existing Price Ruleset.

1. In the **App Menu**, click **Conga CPQ Admin** and then click the **CPQ Admin** tab. The new admin console is launched.
2. On the **Pricing** menu, click **Manage Price Ruleset**.
3. Select a price ruleset.
4. On the **PRICE PIPELINE RULES** tab click **New Price Price Pipeline Rule**.
5. On the **DETAILS** tab, enter the following details.
   a. In **Rule Name**, enter a mandatory rule name.
   b. In **Short Description**, enter a description for the rule.
   c. Click **Target Price Points** and select the price point from the list to associate with the rule.
   d. From **Rule Type**, select a type. The supported values are *Dimension* and *Condition*.
   e. From **Adjustment Applies To**, select an option to which the adjustment is applied. This is for line-item adjustments only. The supported values are List Price, Previous, and Prior Price Point.

Value	Description
List price	The source price for adjustment calculation is the List Price.
Previous	The source price for adjustment calculation is the current running price.
Prior Price Point	The source price for adjustment calculation is the prior price point.

   f. By default, the price pipeline rule inherits the value of **Currency**, from the price ruleset.
   g. The **Sequence** is auto-generated.
   h. To make the rule active, select **Active**.
   i. From **Adjustment Charge Type**, select an option to set adjustment as a charge type. You can change this in the price pipeline rule object.
6. Select the **DIMENSIONS** tab, enter the following details.
   a. 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. For example, you can select a dimension on quantity.

    b. From **Dimension Value Type**, select **Discrete, Range, Cumulative Range,** or **Cumulative Range-Line Item.**

7. Click **Save**. The price pipeline rule is created and listed under the price pipeline rule list.
8. Click **Save**.

Managing Quotes or Proposals

In this section,

Enabling Quote Collaboration in the 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 (up to 5 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.

For detailed flow and use case around Quote Collaboration, see About Quote Lifecycle Collaboration.

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
   - Apttus_Config2__CollaborationRequestId__r.Owner.Name
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. Also, for Queue assignment, ensure that you have listed *Product Configuration* and *Collaboration Request* in the **Supported Objects** list.
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 define Permissions to use Quote Collaboration Feature

The administrator must define proper access permissions for user profiles to use Quote Collaboration. Otherwise, the user cannot see the collaboration feature on the Cart page. Follow the steps below to define access permissions for the user.

1. Go to **Setup> Administrator Setup> Manage Users> Profiles**.
2. Find the profile for which you want to set permissions and click **Edit**.
3. Find **Collaboration Request** under the section **Custom Object Permissions**.
4. Select one or more permission from the list below to define access for the profiles.

Name	Impact on the Cart Page
Read	A disabled collaboration icon is displayed on the Cart page.
Create	The collaboration icon is enabled. The actions fields in the collaboration pop-ups are enabled and the user can assign a line item to a collaborator.
Edit	The **Update** and **Merge** actions are enabled on collaboration pop-up and the user can edit a configuration assigned to them.
Delete	**Cancel** action is enabled on the collaboration pop-up.
Modify All	The **Read**, **Delete**, and **Edit** permissions are selected automatically. The **Update**, **Merge**, and **Cancel** actions are enabled on the collaboration pop-up.

Actions performed on the collaboration request with the **Modify All** permission have a different impact on the collaboration status as opposed to when the **Modify All** permission is not selected. If **Modify All** is selected, when the user abandons the cart, cancels the collaboration request, or deletes the line item, the collaboration record is deleted. Otherwise, the collaboration status changes. If you do not select any permission, the collaboration icon remains disabled on the Cart page.

For more information on collaboration status, see *About Quote Lifecycle Collaboration*.

5. Click **Save**.

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 define field set settings

You must define **Collab Create Fields** and **Collab Merge Fields** in Field Set Settings. For more information, see Configuring Field Set Settings.

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.

Template Name	Description
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 record is created, and every time it's edited	`OR (AND (ISCHANGED(Apttus_Config2__Status__c), ISPICKVAL(Apttus_Config2__Status__c, 'Submitted')), AND (ISCHANGED(OwnerId), ISPICKVAL(Apttus_Config2__Status__c, 'Submitted')))`
QuoteCollabNotifyOnUpdate	Collaboration Request	Evaluate the rule when a record is created, and every time it's edited	`AND ( NOT(ISCHANGED(OwnerId)), NOT(ISBLANK(Apttus_Config2__ChildConfigurationId__c)), NOT(ISCHANGED(Apttus_Config2__ChildConfigurationId__c)), ISPICKVAL(Apttus_Config2__Status__c, 'Submitted'), ISCHANGED( LastModifiedDate ) )`

Workflow Rule Name	Object	Evaluation Criteria	Rule Criteria
| QuoteCollabNotifyOnComplete | Collaboration Request | Evaluate the rule when a record is created, and every time it's edited | ```
AND
(ISCHANGED(Apttus_Config2__Status__c),
ISPICKVAL(Apttus_Config2__Status__c,
'Completed'))
``` |
| QuoteCollabNotifyOnAccept | Collaboration Request | Evaluate the rule when a record is created, and every time it's edited | ```
AND
(ISCHANGED(Apttus_Config2__Status__c),
ISPICKVAL(Apttus_Config2__Status__c,
'Accepted'))
``` |
| QuoteCollabCancelNotificationOnChangeOfOwner | Collaboration Request | Evaluate the rule when a record is created, and every time it's edited | ```
AND (
ISCHANGED(OwnerId),
ISCHANGED( LastModifiedDate ),
ISPICKVAL(PRIORVALUE( Apttus_Config2__Status__c ), 'Submitted' )
)
``` |

Workflow Rule Name	Object	Evaluation Criteria	Rule Criteria
QuoteCollabNotifyOnUpdateAfterCompletion	Collaboration Request	Evaluate the rule when a record is created, and every time it's edited	```AND (
NOT(ISCHANGED(OwnerId)),
AND(ISCHANGED(Apttus_Config2__Status__c
),
ISPICKVAL(Apttus_Config2__Status__c ,
"Submitted"),
OR(ISPICKVAL(PRIORVALUE(Apttus_Config2_
_Status__c), "Completed"),
ISPICKVAL(PRIORVALUE(Apttus_Config2__Sta
tus__c), "Accepted")
)
),
ISCHANGED(LastModifiedDate)
)```

For the corresponding Workflow Rules, create the appropriate email alert action (email template, and the recipient) from the **Workflow Actions** section.

Configuring the Collaboration Dialog Box

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 the 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.

Name	CollaborationRequest__c.User.OwnerId ⓘ		Object Name	CollaborationRequest__c
Lookup Field Name	OwnerId.User		Display Columns	Name
Filter Criteria	LocationId__c = LineItem__c.LocationId__c		Default Flag Field	
Source Field	UserId__c		Source Object	Junction__c

Configuring 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.

> ⚠ When there are more than five line items, then the the **Async Collaboration Merge** enables background processing for collaboration requests.

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

Use cart locking when editing the cart to lock other sales representatives out of editing the cart. They can still access the cart in read-only mode. The cart is unlocked when the Sales rep currently editing the cart saves, finalizes, abandons, or closes the cart. Sales reps in read-only mode can only gain edit access after these operations are performed. Cart locking helps prevent data loss while a Sales rep is working on a configuration. You can use **Enable Cart Locking for Concurrent Access** in Config System Properties to enable cart locking.

If the cart is left idle by the Sales rep currently in edit mode for a predefined time limit, the cart is unlocked automatically. The cart is considered idle if the sales rep does not change line items. Changes include updating a value, or adding or deleting a line item. Other actions on the cart, such as updating cart views, viewing the totals, or applying filters, are not considered modifications to the line items, and time spent is counted as idle time. The time limit is defined in the **Cart Edit Access Idle Timeout in Minutes** config system property. After the time limit is exceeded the cart is unlocked allowing any Sales rep to open the cart in edit mode, locking the cart to the other Sales reps. If you do not define a time limit the cart remains locked until the Sales rep in edit mode does not saves, finalizes, abandons, or closes the cart.

To enable cart locking for concurrent access and setting an idle cart time limit

1. Go to **Setup** > **App Setup** > **Develop** > **Custom Settings** > **Config System Properties** > **Manage.**
2. Select **Enable Cart Locking for Concurrent Access**.
3. Enter a number of minutes in **Cart Edit Access Idle Timeout in Minutes**.
4. Click **Save**.

The **Current User field** in the Product Configuration object contains the ID of the user accessing the cart in the edit mode.

Enabling PDF Security for Generated Proposal Documents

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:
    a. Select the **Enable PDF Security** check box.
    b. 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>
<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 > **Conga 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.

Configuring Quick Quote Mode

The Quick Quote feature allows Sales reps to create a simple quote with fewer clicks from opportunity. The Sales rep can directly go to the Catalog page from Opportunity, configure products, and finalize the cart without having to create a proposal in turn reducing the number of clicks. Quick Quote feature is especially meant for creating budgetary quotes where Sales Reps can begin the quoting process by populating the quote header with limited values.

To enable the Quick Quote feature, you must create and add a custom button to the opportunity layout. CPQ populates the critical header fields in the new quote based on the parameters passed in the formula of the button. You can define variable or static parameters. In the case of variable parameters, CPQ retrieves the values from opportunity. You can use the following parameters in the URL as described:

Fields	Required	URL Parameter	Description
Opportunity ID	Yes	*businessObjectId*	Add the ID of the opportunity to the URL.

Fields	Required	URL Parameter	Description
Price List	Yes	*priceListId*	Add the ID of the price list to the URL. You must use one of the following ways to pass the price list. • Add a static value of the price list ID the URL. This is applicable when the quotes are always created with a fixed price list. • Create a Price List lookup field on the Opportunity object and a formula field to store the ID of the selected price list. Pass this formula field value in the URL.
Account	Yes	*NA*	This field is copied by default.
Expected Start Date	No	*expectedStartDate*	You must create the field on the Opportunity object with datatype as *Date* and add the field in the URL.
Expected End Date	No	*expectedEndDate*	You must create the field on the Opportunity object with datatype as *Date* and add the field in the URL.
Existing Configure Products URL parameters	No	• *flow* • *useAdvancedApproval* • *asyncFinalize* • *deferPricingUntilCart*	Add the parameters based on your business needs.

The custom fields with the same API name, that are common between both opportunity and quote are also populated.

To create a quick quote custom button on Opportunity

1. Go to **Setup** > **App Setup** > **Customize** > **Opportunities** > **Buttons, Links, and Actions**.
2. Click **New Button or Link**.
3. Fill in the details as explained below.

Fields	Description
Label	Enter the name of the button. For example, *Create Quick Quote*.
Name	This field is populated automatically when you click the text box.
Description	Enter a description for the button.
Display Type	Select **Detail Page Button**.
Behavior	Select the behavior based on where you want to display the button
Content Source	Select **URL**.

4. In the Code Snippet block, type the following:

```
/apex/Apttus_Config2__Cart?businessObjectId={!Opportunity.Id}
&recordTypeName=Proposal&priceListId={!Opportunity.PriceListIdId__c}
&useAdvancedApproval=true&flow=QuotingCartGrid&expectedStartDate={!
Opportunity.expectedStartDate__c}&expectedEndDate={!
Opportunity.expectedEndDate__c}
```

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

5. Click **Save**.

To add the quick quote button

After you create the custom button, add the Quick Quote custom button to the layout of the opportunity.

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.

For more information on creating a quick quote, see Creating Quick Quotes.

Customizing 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 through the CPQ *Config Settings* tab.

- Configuring Display Column Settings

  The Display Column Settings enable you to display only those key fields needed to quickly and easily configure your quote/proposal.

- Configuring Display Action Settings

  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.

- Configuring Custom Actions for Auto-Synch of Cart Lines

  The Custom Actions feature has been enhanced to auto-synch cart lines with proposal lines and navigate to other pages.

Configuring Display Column Settings

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
   - Related Purchases
   - 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
   - Mass Edit (Assets)

> ⓘ Mass Edit (Assets) is an invalid selection for Display Type.

- Collaboration Parent Line Item
- Collaboration Child Line Item
- Line Item Key Metrics
- Line Item Waterfall
- Summary Group Key Metrics
- Summary Group Waterfall
- Optional Cost and Profitability Fields
- Cart Summary
- Asset Termination

> ⓘ If you add only the **Termination Date** field, CPQ displays:
> - The Confirm Swap pop-up with the **Effective Date** field instead of the Confirm Installed Products Swap page during asset swap.
> - The Confirm Termination pop-up with the **Termination Date** field instead of the Confirm Termination page during asset termination.
>
> When the Conga Billing package is not installed in your org, it is recommended to display the Confirm Swap or Confirm Termination pop-up. Otherwise, all billing fields will be blank on the Confirm Installed Products Swap or Confirm Termination page.

4. Select your desired flow from the **Flow** picklist.
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.

> ⓘ You should not add Option as a column as Product and Option columns are identical in the CartDetailView page.

The supported values are:
- Termination Date
- Start Date
- End Date
- Original Start Date
- Net Price
- Billed Through Date
- Total Billing
- Invoiced
- Pending Billing

- Estimated Credit
- Asset TCV
- Current Contract Start Date
- Current Contract Value
- Current Contract Billing (Invoiced)
- Current Asset Billing (Invoiced)
- Current Contract Pending Billing
- Current Asset Pending Billing
- Custom asset line item fields

> ⓘ You can allow Sales Representatives to edit the custom asset line item field values on the Confirm Termination page instead of waiting to go to the Cart page to edit these values. For example, they may want to capture the reason for terminating assets. However, CPQ does not support text area type of fields. CPQ carries these edited field values from the Confirm Termination page to the Cart page where Sales Representatives can make further changes to those field values, if required. You can select these fields only if you have added the same to the **Editable Fields for Cancelled Lines** setting. After selecting one of these fields here, if you remove the same field from the **Editable Fields for Cancelled Lines** setting, CPQ automatically removes that field from **Display Column Settings**. You must execute the maintenance job. For more information, see Configuring Installed Products Settings.

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.

Configuring Display Action Settings

The Display Action Settings page provides an 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 change a custom label

1. Go to **Setup > Create > Custom Labels** and search for **Change**" label with Installed Products Change Action as the short description.
2. Click the Change label. The system displays the Custom Label Detail page.
3. Click the button, **New Local Translations/Overrides** in the Local Translations/ Overrides section. The system displays the New Translation page.
4. Select the appropriate language from the drop-down.
5. Enter the Translation Text as an Add-on in the text box.
6. Click **Save**.

The custom label is changed.

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 page where you want to place the custom action.
4. From the **Flow** drop-down, select the flow you want to configure.
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. The Action and Action Label Name display the default actions.
6. You can create a 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.
7. From the **Display As** column, select one of the following:

Value	Description
Action	Select this to set the action as an action button
Task	Select this to set the action as Task menu item in the Status bar.
Action and Task	Select this to set the action as both an action button and a Task menu item.

Value	Description
Sidebar Action	Select this to set the action as an action button on the sidebar of the Configuration page. You can define a maximum of 3 actions in the sidebar. Only the first 3 sidebar actions in the list are valid, CPQ does not display the remaining sidebar actions on the Configuration page at all.
Nav Link	Select this to set the action as a hyperlink in the Navigation bar.
Help	Select this to set the action as a Help menu icon.

For the new UI, the behavior is as follows:

Display As	Behavior
Action	Listed as a primary button. Depending on the sequence that you have defined, only one button is placed on the top and the rest of the buttons are placed in the drop-down list.
Task	Listed as a secondary button. A maximum of 4 secondary buttons are allowed on any page.

For example, on the catalog page displayed below, **Go to Pricing, Installed Products**, **Save,** and **Compare Products** have **Display As** = *Task,* whereas **Abandon**, **Update Price**, **Quick Save** and **Finalize** have **Display As** = *Action.*

8. From the **Action Area** column, select one of the followings to position the action:

Value	Description
Center	Select this for center alignment of the button
Left	Select this for left alignment of the button
Right	Select this for right alignment of the button
More	Select this to add the button as an item under the More action button

> ⓘ **Note**
>
> Note that the alignment settings are not applicable for Nova and Grid UI.

9. In the **Action Style Class** column, type one of the following available style class name for the action.

Style	Description
apt-selected-btn	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.

10. 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

11. 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*.

12. From the **Action Type** column, select one of the following:

- To redirect to 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.

13. To enable or disable actions, select or clear the **Is Enabled** check box.
14. 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.

15. Click **Save** to save the changes you made on this page.

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.

Use Case

You can configure action buttons displayed on the Summary panel on the Configuration page. You can add a maximum of 3 buttons on the Summary panel. If you configure more than 3 buttons, only the first 3 buttons are displayed on the Summary panel. The following action buttons are displayed on the Summary panel by default:

- Go to Pricing
- Update Price
- Validate

To configure action buttons on the Summary panel on the Configuration page

1. Go to **All Tabs** > **Config Settings** > **Display Action Settings.**
2. From the **Display Type** drop-down, select **Attribute Page**.

> ⓘ You can select **Display Type** as **Bundle Page** to configure actions on the sidebar for bundle page in case of sub-bundles.

3. From the **Flow** drop-down, select the flow you want to configure.
4. Find the buttons you want to display on the Summary panel and define the following fields as described:

Field	Value
Display As	*Sidebar Action*
Action Area	*Center*

5. Select the following checkboxes:
   - **Is Enabled**
   - **Always Display**
6. Click **Save**.

To hide default action buttons on the Configuration page

1. Go to **All Tabs** > **Config Settings** > **Display Action Settings.**
2. From the **Display Type** drop-down, select **Attribute Page**.
3. From the **Flow** drop-down, select the flow you want to configure
4. Find the buttons you want to hide and clear the following checkboxes:
   - **Is Enabled**
   - **Always Display**
5. Click **Save**.

Configuring Custom Actions for Auto-Synch of 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.

> ⓘ   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.

Disabling the Clone Icon on the Cart Page

You can now choose to disable the clone icon on the Cart page. Select the **Hide Copy Action** check box in **Config System Properties** to disable the clone icon for in the cart

To disable the clone icon for on the cart page

1. Go to **Setup** > **Develop** > **Custom Settings** > **Config System Properties** > **Manage** > **System Properties**.
2. Click **Edit**.
3. Select the **Hide Copy Action** check box.
4. Click **Save**.

Configuring Flows

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

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=Sample 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.

Configuring Enterprise QTC Profile Value for Quote

When you have enabled TurboPricing in the org to process large-sized carts, Conga recommends you to create such quotes with the **QTC Profile** value as *Enterprise*. In a quote with *Enterprise* QTC Profile, CPQ processes downstream operations asynchronously for a maximum of 10000 lines. When the Sales Representative selects the **QTC Profile** as *Enterprise* during quote creation, CPQ performs the following downstream operations in async mode:

- Finalizing the cart
- Cloning a quote with Line items
- Synchronize quote with opportunity
- Creating an agreement with line items (single agreement)
- Creating an order (single order)
- Creating assets

> ⓘ You must finalize Enterprise quotes using the **finalizeCart()** API if you have more than 1000 line items. For more information on **finalizeCart()** API, see Finalizing Configuration in Enterprise and Smart Cart. You must also use a custom **Submit For Approval** button for any approvals for Enterprise quotes.

You must manually add the picklist value of *Enterprise* in the **QTC Profile** field of the Quote/Proposal, Product Configuration, and Agreement objects.

To add Enterprise as a QTC Profile value

You must add the value *Enterprise* in the **QTC Profile** field of the Quote/Proposal, Product Configuration, and Agreement objects so that CPQ displays the value *Enterprise* in the **QTC Profile** drop-down on the quote for the Sales Representative.

1. Go to **Setup** > **App Setup** > **Create** > **Object**.
2. Click **Quote/Proposal**.
3. In **Custom Fields & Relationships**, Click **QTC Profile**.
4. On **QTC Profile(Managed)** page, click **New** on the **Values** section.
5. Enter *Enterprise.*
6. Click **Save**.

> ⓘ Repeat steps from 3 to 6 for the **Product Configuration** and **Agreement** objects. For the **Product Configuration** object, in Step 3 you must select the **Business Object Profile** field instead of **QTC Profile**.

To create a custom Submit For Approval button

You must use the custom Submit for Approval buttons for all Enterprise quotes. Follow the steps:

1. Go to All Tab > **Config Settings** > **Display Action Settings**.
2. Select *Cart Page* in **Display Type**.
3. Select the flow you configured to use TurboPricing.
4. Add a new row and defined the fields as described in the following table

Fields	Value
Action Label Name	Provide a name for the button. For example, *CustomSubmitForApproval.*
Display As	Select **Action** from the dropdown.
Action Area	Select **Center** from the dropdown.
Action Page	Enter *Apttus_CQApprov__CartApprovals.*
Behavior	Select **Self** from the dropdown.

Fields	Value
Action Type	Select **Quick Redirect** from the dropdown.
Is Enabled	Enable this checkbox.
Always Display	Enable this checkbox.

5. Click Save.

For more information on how to create quotes, [Creating Quotes from Opportunities](#).

Configuring Progress Tracker for Async Operation

When you enable this feature from the Quote/Proposal, Agreement, or Order object , CPQ displays notifications on the Quote Detail, Agreement Detail, and Order Detail page tracking different stages of async operations. The tracker informs the Sales Rep of the progress in real-time. It also immediately informs the Sales Rep in case of errors and provides detailed information about the error, helpful for debugging. For example, when a Sales Rep finalizes a cart asynchronously and returns to the Quote Detail page, a progress bar displays the progress of the *Finalize* operation.

You can use this feature to track the progress of the following tasks that CPQ processes asynchronously:

- Finalizing a cart.
- Cloning a quote or agreement.
- Accepting quote to create order.
- Creating an agreement with line items from Split and Enterprise quotes.
- Activating agreement to create order.

The feature tracks the following stages of the async operation:

- The progress of the async operation.
- The errors encountered while processing the operation.
- Completion of the async operation. Upon completion, the Quote page is refreshed automatically.
- Success notification after the completion of the operation (only supported for Lightning Experience).

The Sales rep can use this feature for async operations for quotes of all profiles namely Regular, Split, and Enterprise.

Follow the steps below to enable the tracker.

To enable the tracker for Salesforce Classic UI

1. Go to **Setup** > **App Setup** > **Create** > **Objects**.
2. Click **Quote/Proposal** or **Agreement** object based on the page you want to enable the tracker for.
3. In the **Buttons, Links, and Actions** section, click **Edit** next to the **View** record.
4. Define the fields as described in the below table:

Field	Description
Label	The label is populated automatically.

Field	Description
Name	The name is populated automatically.
Default	The default has already been selected.
Salesforce Classic Override	Select the **Visualforce page** radio button. From the picklist next to the **Visualforce page** radio button, select • For Quote/Proposal object: **ClassicProposalRecordDetail [Apttus_QPConfig__ClassicProposalRecordDetail]** • For Agreement object: **ClassicAgreementRecordDetail [Apttus_CMConfig__ClassicAgreementRecordDetail]** • For Order object: **ClassicOrderRecordDetailpage [Apttus_Config2__ClassicOrderRecordDetail]**
Lightning Experience Override	Select the **Lightning Page** radio button.
Mobile Override	Leave the field as-is.
Comment	Add comments with a description of the action.

5. Click **Save**.

To enable the tracker in Lightning Experience

1. Go to **Lightning Experience**.
2. Click the wrench icon (⚙). Click **Edit Page**. Lightning App Builder page is displayed.
3. From the **Pages** dropdown, select the Proposal or Agreement record page based on the page you want to enable the tracker for.
4. Under the **Components** tab, find and expand the **Custom - Managed** list.
5. Drag and drop the following components from the **Custom - Managed** list to the proposal record page:

Component	Description
proposalBanner	Add this component to the Proposal record page.
agreementBanner	Add this component to the Agreement record page.

Component	Description
BannerRefreshView	You must hover the cursor over the name and select the correct component as described below: • For Proposal record page: *BannerRefreshView (Apttus_QPConfig)*. • For Agreement record page: *BannerRefreshView (Apttus_CMConfig)*. • For Order record page: orderRefreshView (*Apttus_Config2*).

> ⓘ Conga recommends placing the above components between the **Highlights Panel** and **Path** components.

6. Click **Save**.

You can also translate or override the labels and messages in the progress bar based on your Locale and Language. For more information, see Managing Translation of Fields and Values in Different Language.

Turning off Async Operation Email Notification

You can use the Admin Setting **APTS_DisableCartAsyncNotification** to turn off the email notifications you receive after completion of any async operation such as cart finalization and pricing in every type of quote or agreement. In the Split quote or agreement, the notification for cart splitting is also disabled along with finalization and pricing. If you enabled the Progress Tracker on the quote or agreement page to monitor the progress of all the async operations, email notifications are redundant and you can use this feature to avoid them.

To turn off async operation email notification

1. Click ➕ and click **Admin**.
2. To create a new Admin Setting, click **New**.
3. Enter **Name**, **Value**, and **Code** for the Admin Setting as described in the below table:

Name	*APTS_DisableCartAsyncNotification*
Value	*true*
Code	Leave the field blank

4. Click **Save**.

Configuring Push Alerts

You can enable Salesforce Push Alerts to track async operation status when you are not on the Quote or the Agreement page. You must select **Enable Notification Feed** in Config System Properties. If you enable push alerts, the Sales rep receives notification for the following asynchronous CPQ operations:

- Finalizing a cart
- Cloning a quote or agreement
- Accepting quote to create order
- Creating an agreement with line items from Split and Enterprise quotes.
- Activating an agreement to create an order

To enable push alerts

1. Go to **Setup > Develop > Custom Settings > Config System Properties > Manage**.
2. Find **System Properties** and click **Edit**.
3. Select **Enable Notification Feed** checkbox.
4. Click **Save**.

Configuring Document Generation for Quote

You can you Document Generation(Doc Gen) feature to generate a document for the line items in the quote. You can use this functionality by clicking **Send Proposal** button on the Quote/Proposal Detail or Cart page. After clicking the button you are redirected to Doc Gen Wizard page. The Sales Rep can generate in a particular format, merge documents, preview, and send the documents to customers on the Doc Gen Wizard page. To use the feature, you must configure the following settings.

To configure output format for the profile

You must define an output format for the profile, in case you want to generate documents automatically.

1. Click the All Tabs icon ( ) and click **Proposal Document Output Formats**.
2. Select the profile you want to update.
3. Select one of the following:
    - Prompt
    - Doc
    - Docx
    - PDF
    - RTF
4. Click **Save**.

To add endpoint URL

You must add the endpoint for the document URL that you want to list on the Proposal Doc Gen. Follow the steps below:

1. Go to **Setup** > **Security Controls** > **Remote Site Settings**.
2. Add a New Remote Site.
3. Click **Save**.

To enable Salesforce Files

1. Go to **Setup** > **Develop** > **Custom Settings** > **Proposal System Properties** > **Manage**.
2. Find **System Properties** and click **Edit**.

3. Select **Enable File**.
4. Click **Save**.

To configure preview of File documents on doc gen page

When you are using Files in your org, CPQ does not automatically enable previews for files. You must disable **Enable clickjack protection for customer Visualforce pages with headers disabled** fields, to enable file document preview on the Doc Gen page. Follow the steps below to enable preview on the Doc Gen page:

1. Go to **Setup > Administrator Setup> Security Controls > Session Settings> Manage**.
2. Find **Clickjack Protection**.
3. Disable **Enable clickjack protection for customer Visualforce pages with headers disabled**.

To add Send Proposal button on the quote page

1. On the Quote/Proposal Detail page, click **Edit Layout**.
2. Select **Fields** on the left-hand side menu.
3. Search **Send Proposal** using Quick Find.
4. Drag and drop the field on the quote.
5. Click **Save**.

To sort the generated documents generated on the doc gen page

You must enable the **Sort Documents Descending** proposal system property to sort the generated documents in descending order of Created Date on the doc gen page. If this setting is not enabled, the documents are sorted in ascending order of creation date, by default. For more information, see Proposal System Properties. In addition, enabling the **Sort Documents Descending** proposal system property, CPQ previews the document that was generated last.

> ⓘ If you are upgrading from the May '22 or September '22 to February '23 Preview release of CPQ, you must remove the **APTS_SortDocumentsDesc** admin setting. The admin setting is no longer supported. For more information, see Configuring Admin Settings.

Configuring Fast Doc Gen

You can use this feature to generate the default document automatically. When the Sales Rep clicks on **Send Proposal** button, a document is generated automatically using a queried or default template. The generated document is in output format defined and is displayed on the Doc Gen page. You can set up default document generation for the overall org or for specific scenarios as the document generation depends upon the templates that are defined.

> ⓘ In the case of large quotes, Conga recommends that you must not use Fast Doc Gen as CPQ generates default documents in sync mode.

CPQ generates documents automatically based on the template that is defined in **Query Template Filter** and the **Default Template Name** setting. The automatic generation of documents using a particular template is based on the below-mentioned scenarios:

Scenario	Result
When a template is defined in Default Template Name but not in Query Template Filter	The document is not generated.
When the same template is defined in both Default Template Name and Query Template Filter	The document is generated using that template.
When a single template is defined in Query Template Filter but no template is defined in Default Template Name	The document is generated using that one template.
When multiple templates are defined in Query Template Filter but no template is defined in Default Template Name	the document is not generated.

Prerequisite

An output format must be selected for the profile. Refer to section *To configure output format for the profile*.

To enable fast doc gen

1. Go to **Setup > Develop > Custom Settings > Proposal System Properties > Manage**.
2. Find **System Properties** and click **Edit**.
3. Define the fields mentioned in the table:

Field	Description
Enable Fast Doc Gen	Select the checkbox to enable fast document generation.
Default Email Template	Enter the name of the template to set it as default.

4. Click **Save**.

Configuring Large Document Generation for Quote

You can use the Large Document Generation feature to generate documents for quotes with a large number of line items. The document generation for a large quote is processed asynchronously in small chunks based on the threshold and batch size and the resultant document is attached to **Notes and Attachments**. This feature is available in the **Generate** button and not available in the DocGen, **Send Proposal** button.

You must define the following settings to allow the Sales rep to generate a document for a quote with a large number of line items. You need to define a threshold and a batch size for the number of line items in the quote. When the number of line items in the quote exceeds the threshold, the quote is considered a large quote. The line items in the large quote are processed in smaller chunks based on the batch size you defined.

You must take into consideration the template that the Sales rep uses to generate the document before defining the batch size. For example, if the template includes information about only line item fields is considered a simple template and can have large batch size. Similarly, a template that has too many line item fields, attribute information, adjustment line details would be considered a complex template and should have a small batch size.

To define the threshold and batch size.

1. Go to **Setup > Develop > Custom Settings > Proposal System Properties > Manage**.
2. Find **System Properties** and click **Edit**.
3. Define the fields mentioned in the table

Feild	Description
Enable File	Select the checkbox to enable files in the quote
Enable Submit Merge Call	Select the checkbox to enable the **Submit** button to identify a quote as large doc quote based in the **Large Doc Threshold**.
Large Doc Threshold	Enter a number in the field. ⓘ The default and the maximum value of the **Large Doc Threshold** is *10000*. You can define a value lower than 10000. Setting a value of *-1* for the Large Doc Threshold invalidates the setting.
Large Doc Process Batch Size	Enter a number in the field. ⓘ The default value of the **Large Doc Process Batch Size** is 1*000* and a maximum of *2000*. You can define a value lower than 2000, based on template complexity.

4. Click **Save**.

To define APTS_LargeDocSObjects Admin Setting

By default, CPQ considers Proposal Line Item and Agreement Line Item as Large Objects for document generation. In case you used any other object in the template, you must use this setting to define such objects as Large objects. You can specify multiple objects as Large Objects and also provide batch size for each of these objects. You must define the Admin Setting **APTS_LargeDocSObjects** by following the steps below:

1. Go to **All Tabs** > **Admin.**

2. Click **New**, to create a new record.
3. Fill the following details:

Field	Value
Name	APTS_LargeDocSObjects
Value	Leave the field blank
Code	*Apttus_QPConfig__ProposalLocation__c=5* *QTC_Proposal_Related_Line_Item_c=1000* ⓘ Use the format Auto Applied *<Object_API_name=<Batch Size>>* to defined the objects.You can add more objects separated by a new line. Decide the batch size based on the complexity of cart and template.

4. Click **Save.**

Configuring Conga Sign for eSignature

You can integrate CPQ with eSignature solutions like Conga Sign. Conga Sign provides the ability to send the quote or proposal to the customer to gather signatures.

To Install Conga Sign

You must install the Conga Sign package in your org to use the Conga Sign features. For more information, see Installing Conga Sign Package.

To configure Conga Sign

Once you install the package in the org, you must complete the admin configuration in Conga Sign and set up integration with CPQ Proposal object. For more information, see Configuring Conga Sign.

To edit the layout of the Quote Detail page

After you configure Conga Sign, the **Send with Conga Sign** button is displayed on the Quote Details page. If the button is not displayed, you must edit the layout of the Quote Details page, and add the button on the header. You can also add Conga Sign Recipient and Conga Sign Transaction History related lists on the Quote Details page.

For more information, Sending Documents for eSignature Using Conga Sign.

Creating Promotional Banners

The banner settings feature has been deprecated.

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 }
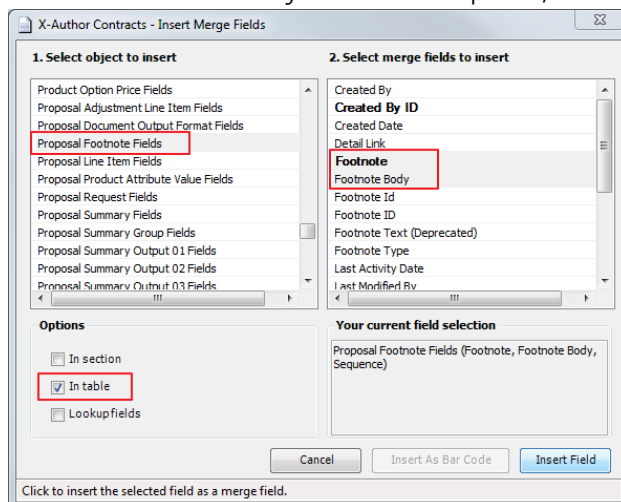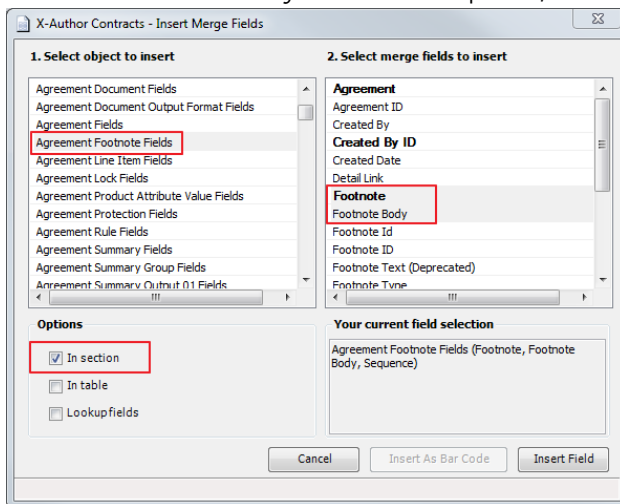
{ MERGEFIELD AgreementFootnote_FootnoteId \* MERGEFORMAT }

{ MERGEFIELD AgreementFootnote_FootnoteBody \* MERGEFORMAT }

{ MERGEFIELD AgreementFootnote_Sequence \* MERGEFORMAT }

{ MERGEFIELD Apttus_CMConfig__AgreementFootnote_section_end \* MERGEFORMAT }
```

Retaining the Order of Line Items in a Generated 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:

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 > Conga 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.

Updating the sequence of options in bundles that are reordered might cause CPU time out error. Conga recommends refraining from updating the option sequence after reordering the bundle. You must add an admin entry to disable the sequencing of options.

> ⚠ Updating the sequence of options in bundles that are reordered might cause CPU time out error. Conga recommends refraining from updating the option sequence after reordering the bundle. You must add an admin entry to disable the re-sequencing of options by following the steps below:
>
> 1. Click All Tabs icon (✚) and click **Admin.**
> 2. Click **New** to create a new Admin entry.
> 3. Fill the following details
>
> | Field | Value |
> |---|---|
> | **Name** | *APTS_IncludeBundleOptionsForResequencing* |
> | **Value** | *false* |
> | **Code** | Leave the field blank |
>
> 4. Click **Save**.

Adding a Link to Quote in Cart Finalize Notification Email Template

CPQ sends a notification email to the quote owner after completing the cart finalization. You configure the template of the notification email to include a link that opens the associated quote for quick access. You must only configure this feature if you are upgrading CPQ to September '22 release or later. Otherwise, if you are installing CPQ for the first time, the quote link is available by default.

To add proposal link to cart finalize notification email template

Perform the following steps.

1. Go to **Setup** > **Administration Setup** > **Communication Templates** > **Classic Email Templates**.
2. Select the **Apttus CPQ Email Templates** folder.
3. Click **SplitCartFinalizeSuccess**. Click **Edit Template**.
4. Copy and paste the given code into the **Email Content** tab.

```
1   <!--
2   Apttus Config & Pricing
3   SplitCartFinalizeSuccess
4   Email template for sending out finalize success email
5   @2018 Apttus Inc. All rights reserved.
6   -->
7   <messaging:emailTemplate subject="{!relatedTo.Name} ({!
    relatedTo.Id}) - Task Successful"
8   recipientType="User"
9   relatedToType="Apttus_Config2__ProductConfiguration__c">
10
11  <messaging:htmlEmailBody >
12      Hi {!relatedTo.LastModifiedBy.FirstName} {!
    relatedTo.LastModifiedBy.LastName},
13      <br /> <br />
14      The configuration has been finalized successfully -
     <c:RecordLink recordId="{!
    relatedTo.Apttus_Config2__BusinessObjectRefId__c}" recordName=""/>.
15  </messaging:htmlEmailBody>
16  </messaging:emailTemplate>
```

5. Click **Save**.

Managing Guided Selling

The Guided Selling feature has been deprecated since the CPQ Spring '21 release. The chapters related to the features have been removed as well.

Managing Promotions

The Conga Promotions package is available as an add-on with the CPQ license and package.

With the Conga Promotions package, you can incentivize the sales of products in your inventory. Promotions add value to product through discount pricing and add reward for purchase. Promotions provide value for money for your customers and contribute to better sales for your organization.

If you are a marketing manager or are responsible for promoting some products over others, you use historical transaction and sales data to create promotions. You can use promotions to:

- build traffic to the storefront
- create extra interest in product offering
- introduce new product, generate trial, or increase penetration in a specific market or segment
- increase sales of products during down-period
- provide competitive pricing to products during heavy competition
- increase revenue margins
- enhance customer retention
- clear stocks of a product that is about to reach end-of-life
- sell excess Inventory
- motivate staff

A promotion is a marketing technique that you apply to reduce the list price of a product or a service. A promotion consists of all non-tech activities that marketers use to try to increase sales over a specific period of time. You can create such promotions and restrict their scope, limit, and benefits so that your sales representatives can apply promotions to specific products, for specific customers, and for a limited period.

For more information on how promotions work, see the Promotions Workflow.

Usage of Promotions

In the Managing Promotions section, 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 the Impact of promotion on quotes and orders
- Analyze the effectiveness of promotions

If you are a sales representative, see Applying Promotions on Line Items in the Cart, to learn how to

- View the available Pull promotions
- Add a promotion code and apply promotions

For more information on Promotion Management, see About Promotions Workflow.

About 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. For more information, see Types of Promotions.

Types of Promotions

The following are the major categories of promotions that you see on the Incentives page:

- Normal promotion: This is the default promotion that CPQ allows you to create. Promotions are applicable on the Cart.
- Order-level promotions: Order-level promotions are called rebates. For more information, see *Rebates Management on Salesforce Administrator Guide*.
- Sales promotion: A marketing manager or pricing manager can define promotions that can be grouped with the adjustments defined by the Sales Representative. A sales promotion allows the Sales Representative to modify the promotion in the shopping cart and structure discounts for deals. A sales promotion can be bucketed with the adjustments defined by the Sales Representative. Adjustments that are made to the Cart for a sales promotion are listed in the adjustment bucket.

To create a sales promotion

Perform the following steps to create a sales promotion:

1. Go to **Setup** > **Create** > **Objects** > **Incentives**.
2. In the **Buttons, Links, and Actions** related list, click **New Button or Link**. The New Button or Link page is displayed.
3. Enter the following details:
    a. **Label**: Enter a label for the button, for example, *New Sales Promotion*.
    b. **Name**: Enter a name for the button. This field is auto-populated based on the Label when you click inside the field.
    c. **Description**: Enter a description for the button.
    d. **Display Type**: Select a display type for the button, for example, *List Button*.
    e. **Behavior**: Select what should happen when the user clicks the button or link.
    f. **Content Source**: Select a content source, for example, *URL*.
    g. In formula editor, enter the following formula for the button:

    ```
    /apex/Apttus_Config2__IncentiveManager?
    DisplaySetting=Promotion&UseType=Promotion&SubUseType=Sales&DataSource=Apt
    tus_Config2__OrderLineItem__c&ApplicationMethod=Buy X Get X
    ```

4. Click **Save**.
5. Add the new button to the layout of the Incentives page.

An action button is created on the Incentives page. You can now create a new sales promotion.

Furthermore, you can create the following types of promotion for each major category:

> (i) You can configure how the benefit must be applied on a selected product. For example, a promotion can be a discount in price or an increased quantity of product.

Promotion Type	Description	Example
Buy X Get X	When the customer purchases a product, you offer some benefit on the same product.	As a marketing manager of a mobile phone company, you need to promote a new smart phone. You create a promotion where you offer your customers 20% discount on the price of the phone that they buy.
Buy X Get Y	When the customer purchases a product, you offer some benefit on a different product.	As a marketing manager of an e-commerce company, you need to increase the sales of a refrigerator. You create a promotion where you offer your customers a free voltage stabilizer when they buy the refrigerator.
For every X Get X	When the customer purchases a certain quantity of a product, you offer some benefit on the same product.	As a marketing manager of a telecommunications company, you need to retain existing enterprise customers. You create a promotion where you offer your customers 10% discount on the price of each subscription for every 10 subscriptions that they buy.
For every X Get Y	When the customer purchases a certain quantity of a product, you offer some benefit on a different product.	As a marketing manager of a software company, you need to provide competitive pricing to a software as a service product during heavy competition. You create a promotion where you offer your customers free training for every 10 licenses that they buy.

Configuring Promotions

With the Promotions Management application, you can now manage, execute, and analyze promotions using the 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 Conga Promotions to create and manage promotions like,

- buy X get X promotions
- tiered promotions
- automatically applied promotions
- coupon-based promotions

You can apply promotions on line item and on order levels. If you have applied promotion on order level, the promotion type is restricted to Buy X get X only.

> ⚠ - Line item promotions (Allow User Override = true) should not display incentives with Context Type = Summary Group.
> - If the promotions is "push" promotion, the promotions are auto applied to the matching quote total or total summary group.
> - If the promotions is "pull" promotion, the user can select the promotion and have it applied to the matching grant group total or total summary group.
> - When the user override is enabled, only display Promotions, of which the context type is "line item.

For more information on how promotions work, see About Promotions Workflow and Types of Promotions

Preparing Data for Promotions

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 promotion plan that guarantees fulfilling 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 promotions
- Create data roll-up

This data must already be available before you can implement or roll-out your promotion. This section explains how you can access data and then include or reference this data in the Incentives that you create.

Defining User Profiles and Permissions

If you are a CPQ administrator, you must first define who can access the Promotions configuration page and what actions they can perform on this page. You must create users and define permissions for each user of each profile as listed in the following table:

User	Profile	Permissions
Marketing Manager	• Create or edit promotion programs • Submit promotions for approval • Analyze impact of promotion on quotes or orders • Analyze promotion effectiveness	• Read or Write access on all objects used for promotions management • Read access on adjustment line items, Limit Data table, rollup data, and other related objects
Marketing Executives	• Analyze impact of promotion on quotes or orders • Analyze promotion effectiveness	• View access on Promotion objects, quotes, and proposals • Read access on adjustment line items, Limit Data table, rollup data, and other related objects
Promotion Approvers	• Approve or reject promotions • View or analyse promotions	• Read or Write access on promotion objects • View access on all objects used for promotions
Sales Representatives	Using promotions on quotes • View available Pull promotions • Add promotion code and apply promotion • View promotion benefit	View access on promotion objects, quotes, and proposals, adjustment line item objects, and Limit Data table

User	Profile	Permissions
Customers	Using promotions on the e-commerce orders • View available Pull promotions • Add promotion code and apply promotion • View promotion benefit	View access on promotion objects, quotes, and proposals, adjustment line item objects, and Limit Data table

For more information of user profiles and permissions, see About Permission Sets.

Configuring Incentive Records

With the help of custom settings, you can show or hide sections 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 > Develop > Custom Settings > 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. Enter *Promotions* here.

Field	Description
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. 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. Enter *Promotion*.

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 > Create > Objects > Incentive.**
2. Go to the **Field Sets** section and 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**.
3. Search for your field, add or remove it from the fieldset.
4. Click **Save**.

Creating Price Dimensions 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. On **App Menu**, click **Conga CPQ Admin** and then click the **CPQ Admin** tab. The new admin console is launched.
2. On the **Pricing** menu, click **Manage Price Dimensions**. A list of existing price dimensions is displayed.
3. Click **New Price Dimension**.
4. Enter the values for fields described in the following table.

Field	Description
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. • 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.

Field	Description
Type	Select from one of the following options:
	ⓘ The field **Type** is related to promotions. Based on the selected value, CPQ creates a dimension type for promotion. For example, you can create a quantity-based dimension by selecting *Quantity* from the **Type** drop-down. To set up values for the **Type** field, go to **Setup** > **Create** > **Objects** > **Price Dimension** (Apttus_Config2__PriceDimension__c) > **Type** (Apttus_Config2__Type__c) field.
	• **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 • **Equipment** - if the if the scope of your promotion is based on equipment (Service CPQ)
Description	Describe the purpose of this price dimension so others in your organization can use it effectively.

5. Click **Save**.

For more information on Price Dimensions, see Creating Price Dimensions.

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** > **Update** Pricing Fields.

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 **Conga 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, Related 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	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 Approval Workflows for Promotions

You can define a hierarchy so that Promotions beyond a certain dollar amount or percentage require a manager's approval. CPQ allows you to selectively turn on or off approvals at a flow level. You must set the **isCartApprovalDisabled** parameter to True in the URL that you have configured on the **Configure Products** field of the respective quote. For more information, see Creating Custom Buttons for Different Flows.

To create an Approval workflow for Promotions,

1. From the force.com app menu, select **Conga 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.
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, see Approval Functional End User Workflow.

Configuring Data Rollups for Promotions

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. Go to the Salesforce App Menu and select **Apttus Incentive Setup**.
2. Click **Incentive Rollups**. The Incentive Rollups page with existing rollups is displayed.
3. Click **New**. The Rollup Rules page is displayed.
4. In the **Rollup Information** section, enter values for the fields described in the following table.

Field	Description
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 rollup, describe the purpose and function of this data rollup.
Active	Select the **Yes** radio button to activate this data roll-up and make it available for reference in your Promotions.

5. In the **Rollup Value** section, enter values for the fields described in the following table.

Field	Description
Rollup Type	Select **SUM** from the drop-down menu.
Business Object	Select from one of the following options. • Asset • Asset Line Item • Product • Product Attribute Value • Product Configuration
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.
Rollup Field	Select from one of the following options. • Order Line Item • Cart Line Item

6. In the **Rollup Duration** section, enter values for the fields described in the following table.

Field	Description
Rollup Frequency	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.
Rollup Period Source	The source of the period depends is the **Business Object** you defined in the Rollup Value section.

Rollup Start Date Type	If you select *Constant,* only the **Rollup Start Date** is visible. Enter a start date for the rollup. If you select *Reference Value Source,* the following fields are visible. Enter values for those fields.

Field	Description
Data Source	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**.
Field	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
Offset Type	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 • Offset (+) Months • Offset (-) Months By default, the Value is constant.
Offset Value	Enter the number of days or months by which you want to offset the start date of the data roll-up.

Rollup End Date Type	If you select *Constant,* only the **Rollup End Date** is visible. Enter an end date for the rollup.
	If you select *Reference Value Source,* the following fields are visible. Enter values for those fields.

Field	Description
Data Source	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**.
Field	Select from one of the following options. • Created Date • Effective Date • Expiration Date • Last Modified Date • Last Referenced Date • Last Viewed Date • System Modstamp
Offset Type	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 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.

7. In the **Rollup Group By Parameters** section, enter values for the fields described in the following table.

Field	Description

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.

8. In the **Scope** section, enter values for the fields described in the following table.

Field	Description
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.
Region	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.

9. In the **Inclusion Criteria** section, enter values for the fields described in the following table.

Field	Description
Field	Select an option from this list of fields to which the promotion must be applied. The fields available depend on what **Reference Data Source** you selected while defining the information.

Operator	If the field you defined is a pick-list or a lookup field, you can select one of the following operators. • equal to • not equal to • in • not in For any other field you defined, select one of the following operators. • equal to • not equal to • less than • less than or equal to • greater than • greater than or equal to • contains (displayed only if you selected a field of type text or area such as Description)
Value Type	If you select *Constant* (default), only **Value** is displayed. If you select *Field Value*, **Data Source** and **Field** are displayed. 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. If you selected *Product* from the **Field** drop-down, you can search a product by its name or code in the **Value** field.
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.
Criteria Expression	Apttus Incentive Management populates this field when you enter values for the fields that defines the scope.

10. Click **Save**.

Defining Promotions

Every promotion you create has the following aspects that you must define:

- Promotion Information
- Promotion Scope
- Promotion Criteria
- Promotion Benefits
- Promotion Limits

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 the prerequisite tasks before creating your first promotion plan.

Defining the Information of a Promotion

To define a new promotion,

1. Go to the Salesforce App Menu and select **Conga Incentive Setup**.
2. Click **Incentives** > **New**.
3. On the Incentives information page, define values for the fields described in the following table.

Field	Description
Promotion Name	User-specific name for the promotion. This name is displayed to the user at any place the promotion 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.
Promotion Type	Different types of promotions. Supported types are: • Buy X Get X • Buy X Get Y • For Every X Get X • For Every X Get Y To configure what options to be displayed in this drop-down, go to **Setup** > **App Setup** > **Create** > **Objects** > **Incentive** > **Application Method** (API name: Apttus_Config2__ApplicationMethod__c) > **Values** section.
Reference Data Source	You can apply promotions on line item and on order levels. Select from the following options: • Line Item: To apply a promotion on line item level. This line item is the primary object to which your promotion is applied. • Summary Group: To apply promotion on the order level. Note: For Summary Group, the promotion type is Buy X get X only.
Promotion Code	This is the code that is used for applying promotion. This code is captured on the order or proposal line items when a promotion is applied to the line item. 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?** checkbox is cleared, the user can use the incentive code to avail a promotional offer.
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.
Effective Date	Date from which the promotion is applicable. Cannot be blank. By default, it is set to current date (date when you create the promotion record).
Expiration Date	Date on which the promotion expires. Can be blank. Needs to be greater than Start Date. By default, it is set to 1 year from the Effective Date.

Field	Description
Promotion Group	This is a generic grouping of the Incentives for analyzing and tracking the incentives. The user can select a promotion group for each promotion. The list of Promotion 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.
Combine with Other Promotion?	Enables you to combine the new promotion with an existing promotion.
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 set up). Approvals set up would be similar to custom object approvals for any object.
Description	User-entered description for the promotion.

4. Click **Save** to save the changes or click **Save & Continue** to move to the next tab.

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**. If you do not select a value, CPQ performs evaluation on all items for each field and that can impact the performance.
> 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.
Product Family	Select the Product Family that you want to apply the promotion to.

Field	Description

Product	Select the specific products that you want to apply the promotion to. You can search a product by its name or code.
	
	If you selected **Promotion Type** as *Buy X Get X* on the Information page, you will see the following options:
	
	    • **Include (OR)**: CPQ applies promotion on all products (in scope), which are available in the cart.
	    • **Include (AND)**: CPQ applies promotion on all products (in scope) only if the products are added to the cart.
	    • **Include Bundle Options (AND)**: CPQ applies promotion to a bundle and options only if the bundle is configured with specified options.
	    • **Exclude**: The promotion will not be applied to the products specified.
	
	ⓘ To see these options on the UI, you must update **Setup** > **App Setup** > **Create** > **Objects** > **Price Rule** object > **Product Criteria Oper** field.
	    • To use **Include (OR)** scope, update the **Product Criteria Oper** field as "in".
	    • To use **Include (AND)** scope, update the **Product Criteria Oper** field as "all".
	    • To use **Include Bundle Option (AND)** scope, update the **Product Criteria Oper** field as "all in bundle".
	
	If you selected **Promotion Type** as *Buy X Get Y* on the Information page, you will see the following options:
	
	    • **Include (OR)**: If one of the specified products is included in the cart, CPQ applies the promotion to the benefit product.
	    • **Include (AND)**: If all specified products are included in the cart, CPQ applies the promotion to the benefit product.

Field	Description
	• **Include Bundle Options (AND)**: If the bundle and all specified option products are included in the cart, CPQ applies the promotion to the benefit product. • **Exclude**: CPQ does not apply the promotion to the products specified.
Region	Select geographical regions of the customers that you want to make this promotion available to.
Product Group	Select the Product Group that you want to apply the promotion to.
Country	Select 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.
Accounts	Select the accounts for which you want to make this promotion available.

3. Click **Save** to save the changes or click **Save & Continue** to move to the next tab.

After the scope, you must now define the criteria for this promotion.

Defining the Criteria of a Promotion

You must define the criteria for a promotion before making 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 based on:

- any the past purchases of customers or rolled up metrics of past purchases.
- 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 applies to.

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.

Field	Description
Field	Select an option from this list of fields to which the promotion must be applied. The fields available depend on what **Reference Data Source** you selected while defining the information.
Operator	If the field you defined is a picklist or a lookup field, you can select one of the following operators. • equal to • not equal to • in • not in For any other field you defined, select one of the following operators. • equal to • not equal to • less than • less than or equal to • greater than • greater than or equal to • contains (displayed only if you selected a field of type text or area such as Description)

Field	Description
Value Type	If you select *Constant* (default), only **Value** is displayed. If you select *Field Value*, **Data Source** and **Field** are displayed. 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	This field is visible only if you select *Constant* from the **Value Type** drop-down. 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. If you selected *Product* from the **Field** drop-down, you can search a product by its name or code in the **Value** field. The search results display the product name and code wherever applicable. If a product has a code defined, the search result displays name and code; else the search result displays only the name.
Data Source	This field is visible only if you select *Field Value* from the **Value Type** drop-down. 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	This field is visible only if you select *Field Value* from the **Value Type** drop-down. Select one from the list of fields from the child filter object.

3. Click **Save** to save the changes or click **Save & Continue** to move to the next tab.

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 work pane, click **Benefits**.
2. From the **Benefits** section, enter the following details:

> ⓘ In the Benefits section, the **Rule Name**, **Product Family**, and **Product** fields are available only if you selected the *Buy X Get Y* promotion.
> In the Benefits section, the **For Every X Criteria**, **Benefit Product**, and **Benefit Type** sub-sections are available only if you selected the *For Every X Get X* and *For Every X Get Y* promotions.

   a. In the **Rule Name** field, enter a name for the benefit rule.
   b. From the **Product Family** lookup, search and select the required product families.
   c. From the **Products** lookup, search and select the required products. You can search a product by its name or code. The search results also display product name and code wherever applicable. If a product has code defined, the search result displays name and code; else the search result displays only the name.

   > ⓘ For Buy X Get Y promotions, if the charge types of scope and benefit products are different in price list items, you must select a charge type for the benefit product in the **Charge Type** multiselect box. Otherwise, CPQ does not apply the promotion on the benefit product. CPQ applies

> the Buy X Get Y promotion to the benefit product regardless of the charge type present on the product you have selected in the Scope and Benefit sections.

Use the following operators to define the benefit for multiple products:

- **Or**: CPQ applies promotion only to the first product that matches the criteria (selected by default).
- **And**: CPQ applies promotion to all products on the benefit criteria. All benefit products must exist on the cart to apply the promotion.

d. In the **Charge Type** multiselect box, select the required charge type for the benefit products if the charge types of scope and benefit products are different in price list items.

> ⓘ The **Charge Type** multiselect box is applicable only for Buy X Get Y type of promotions. The charge type of the product you selected from the **Product** drop-down in the Scope section does not have any impact on the charge type you select for benefit product in the **Charge Type** multiselect box in the Benefit section.

- If you select multiple benefit products from the **Products** lookup, you must select multiple charge types in the **Charge Type** multiselect box depending on their charge types in price list item.
- If you select only one benefit product from the **Products** lookup and its charge type is same of the charge type of the scope product, CPQ applies promotion on the benefit product even if you do not select a charge type in the **Charge Type** multiselect box.
- If you select only one benefit product from the **Products** lookup and its charge type is different from the charge type of the scope product, you must select a charge type in the **Charge Type** multiselect box. Otherwise, CPQ does not apply promotion on the benefit product. For example, the charge type of the scope product is Standard Price and the charge type of the benefit product is License Fee, you must select License Fee for the benefit product in the **Charge Type** multiselect box.
- If the benefit product has two charge types in price list item, you must select only one of the charge types in the **Charge Type** multiselect box depending on which charge type you want CPQ to apply the promotion on the benefit product.

e. Enter the details in the following subsections:

> ⓘ In the Benefits section, the **For Every X Criteria**, **Benefit Product**, and **Benefit Type** sub-sections are available only if you selected the *For Every X Get X* and *For Every X Get Y* promotions.
>
> You can create a promotion of type *For Every X Get X* where X can be multiple products or any product from a group. For example, Mobile Phones is a group of products consisting ABC 13, ABC 13 Pro, and ABC 13 Max phones. You create a *For Every X Get X* promotion for Mobile Phones as "for every 5 mobile phones get 20% discount on the 6th mobile phone". When the users add any product from Mobile Phones (that are defined in the criteria of the promotion) to the Cart; CPQ applies the defined *For Every X Get X* promotions to those products. For example, if the user adds 6 ABC 13 and 6 ABC 13 Max phones to the Cart, CPQ provides 20% on the 6th 6 ABC 13 and 6th ABC 13 Max phones. However, the users must fulfil the quantity criteria for each product to get the benefit for that product. For example, the users must add 5 ABC 13 mobile phones to get 20% discount on the 6th ABC 13. The users will not get any benefit if they add 4 ABC 13 and 2 ABC Pro phones to the Cart.
>
> From Conga CPQ May '22 release, CPQ does not add the criteria product on the Price Rule entry object. All products in scope are eligible for promotions. After upgrading Conga CPQ to May '22, the existing or previously defined *Every X Get X* promotions do not work with multiple products. You must create a new promotion to support multiple products. However, the existing *Every X Get X* promotions will work for a single product according to their current implementation.

   i. **Criteria Value**: Enter the criteria value on the scope product for the promotion to be applied on the benefit product. The criteria value determines how many quantities of the scope product the user must buy for the promotion to be applicable on the benefit product.
- **Type**: Select Quantity.
- **Metric**: Select Quantity.
- **Value**: Enter a numerical value.

  ii. **Benefit Product**: Select the benefit product on which promotion must be applied.

 iii. **Benefit Type**: Select Price and Quantity.

 iv. **Benefit Quantity**: Enter a numerical value. The benefit quantity determines on how many benefit products the promotion is applicable.

3. From the **Rate Type** drop-down, select a value.

ⓘ On the **Rate Type** drop-down, CPQ supports only *Single Rate, Rate Table - No Tiers*, and *Rate Table - With Tiers* rate types.

Rate Type	Description
Single Rate	Selecting this option allows you to define lump sum benefits which do not depend on transaction volume or amount. This is an ideal option when your promotion is not dependent on the amount or quantity of the line items. Select this value if your promotion calculation has no dependency on the amount or quantity ordered. Example, a seasonal promotion that offers 15% off for smartphones. If you select this Rate Type, enter the following values:

Rate Type	Description

Field	Description
Rate Value Type	Select a value to define how you want to apply the benefit. The supported values are: • % Discount: A percentage discount specified in the **Adjustment Amount** field is applied on the option selected in **Adjustment Applies To**. • Discount Amount: A discount amount specified in the **Adjustment Amount** field is applied on the option selected in **Adjustment Applies To**. • % Markup: A percentage markup specified in the **Adjustment Amount** field is applied on the option selected in **Adjustment Applies To**. For example, if you want to increase the price of laptops during the month of September, you can apply a percentage markup that adds to the List price. • Markup Amount: A markup amount specified in the **Adjustment Amount** field is applied on the option selected in **Adjustment Applies To**. For example, if you want to increase the price of laptops during the month of September, you can apply a markup amount that adds to the List price. • Price Override: The amount specified in the **Adjustment Amount** field is overrides the option selected in **Adjustment Applies To**.
Adjustment Applies To	The value of this field controls to which price point the adjustments are applied. For example, whether the price discounts are applied to the total net price or the starting price.

Rate Type	Description

Field	Description
Adjustment Amount	Enter the adjustment amount.
Benefit Uom	Select the benefit unit of measure for the product.

Rate Type	Description
Rate Table - No Tiers	Allows you to conditionally define promotional amount. Examples of conditions are products, regions, account types. If you select this Rate Type, enter the following values:

Field	Description
Benefit Uom	Select the benefit unit of measure for the product.
Adjustment Applies To	The value of this field controls to which price point the adjustments are applied. For example, whether the price discounts are applied to the total net price or the starting price.
Enable Benefit Effectivity Dates	Select **Yes** to define the start and end dates for the benefit. Select the dates in the **Effective Date** and **Expiration Date** lookup fields.
Rate Table Columns	This field is used to define the conditions to be used with the rate table. If you need to define different promotions for different products, make sure the product column is added. If you would like to add another condition, make sure to select the value from the picklist and click the **Generate** button.

Rate Type	Description
Rate Table - With Tiers	Selecting this option allows you to define promotional amount that depends on transaction volume or amount. If you select this Rate Type, enter the following values:

Field	Description
Rate Table - Tier Type	The value of this field controls how the promotion is calculated. *Rate Per Each Tier* - Promotion is calculated at different pricing levels. The promotion amount for each unit depends on which tier the unit falls under. This is the most common scenarios for customer promotion calculation. *Highest Tier Rate* - Promotion is calculated based on the total volume or revenue. Each unit receives the same promotional amount.
Tier Metric Type	The value of this field controls which metric will be used to qualify for tiers. *Volume* - Volume or quantities are used to qualify for tiers. Select this value if your promotion varies based on the purchase quantities but has no dependencies on the dates. *Volume with Dates* - Volume and dates are used to qualify for tiers. Select this value if your promotion varies based on the purchase quantities as well as the when they purchased. *Revenue* - Total quote amount is used to qualify for tiers. Select this value if your promotion varies based on the revenue amount but not on the dates. *Revenue with Dates* - Total quote amount and dates are both used to qualify for tiers. Select this value if your promotion varies based on the purchase amount as well as the when they are purchased.

Rate Type	Description		
	Field	**Description**	
	Benefit Uom	Select the benefit unit of measure for the product.	
	Metric Calculation Period	Select a period over which the tier metric is aggregated. The supported value is *No Rollup*, which means the tier metric is based on the value from individual transaction only.	
	Metric Value Source	The value from this field determines the source based on which the volume or revenue is calculated. The calculated volume or revenue amount is used to determine the tier. Select this value if the Tier Metric Type is Volume. Once Quantity is selected, you will see that in the Rate Table Structure, the "from" and "to" columns become "From Quantity" and "To Quantity".	
	Adjustment Applies To	The value of this field controls to which price point the adjustments are applied. For example, whether the price discounts are applied to the total net price or the starting price.	
	Rate Table Columns	This field is used to define the conditions to be used with the rate table. If you need to define different promotions for different products, make sure the product column is added. If you would like to add another condition, make sure to select the value from the picklist and click the **Generate** button.	

4. Click **Save** to save the changes or click **Save & Continue** to move to the next tab.

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*. You can select specific values for all the objects: • Account: All, Bill To, and Ship To • Contact: All • User: All, Created By ID, and Last Modified By ID
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.
Max Limit Value Type	Specifies that a limit will be a constant value.

Field Name	Description
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. If you enable **Setup** > **Build** > **Develop** > **Custom Settings** > **Incentive System Properties** > **Manage** > **Enforce Limits**, when a quote is finalized, CPQ applies the incentive limit across quotes in the same account. For example, if **Limit Value** is *5* and there are two quotes with three cart line items each, CPQ applies the incentive limit on three items in the first quote and two items in the second quote. If you disable **Setup** > **Build** > **Develop** > **Custom Settings** > **Incentive System Properties** > **Manage** > **Enforce Limits**, only when a quote is accepted, CPQ applies the incentive limit on each quote of an account. For example, if **Limit Value** is *5* and there are two quotes with five cart line items each, CPQ applies the incentive limit on five items in the first quote and five items in the second quote. CPQ resets the incentive limit for each quote. ⓘ The incentive limit is applicable when users finalize the cart. If they abandon the cart where promotion is applied, CPQ resets the incentive limit.
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*.

4. Click **Save** to save the changes or click **Save & Continue** to move to the next tab.

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.
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 for a Promotion

Coupons are generally alphanumeric codes that correspond to a promotion. With Conga Promotions, 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.

> ⓘ You can generate coupons only if you defined **Auto apply?** = **No** on the Information page. If **Auto apply?** = **Yes**, the **Generate Coupons** button is disabled.

To generate coupon codes for a promotion that you have created,

1. On the left work pane, click **Coupons**.
2. In the **Quantity** field, enter the number of coupons you want to generate and click **Generate Coupons**.
3. Enter values for the fields described in the following table:

> ⓘ These fields act as filter criteria as well. For example, the coupon assign-to account or user needs to match the account field value that is specified on the quote or the logged-in user respectively. If the Account field value on the quote is missing, the coupon assign-to account needs to match the account linked to the opportunity.

Field	Description
Active	Select the checkbox to activate the coupon.
Start Date	The start date for the coupon.
End Date	The last date this coupon is valid.

Field	Description
Coupon Limit	The number of times you can use this coupon.
Assigned To Dimension	The level at which the coupon is valid. Select from one of the following options. • Incentive User • Account
Assignee	Select the user or account to which the coupon is applicable.

4. Click **Save** to save the changes or click **Save & Continue** to move to the next tab.

> ⓘ After coupons are generated for a promotion, the Apply Promotions dialog does not list the promotion. The user must enter the coupon code to apply coupons instead.

5. Select specific coupons in the list and click **Apply to Selected Coupons**. You can also mass edit coupons by selecting the checkbox at the header and clicking **Apply to Selected Coupons**.

6. To create multiple coupons for one incentive, use the following sample code:

```
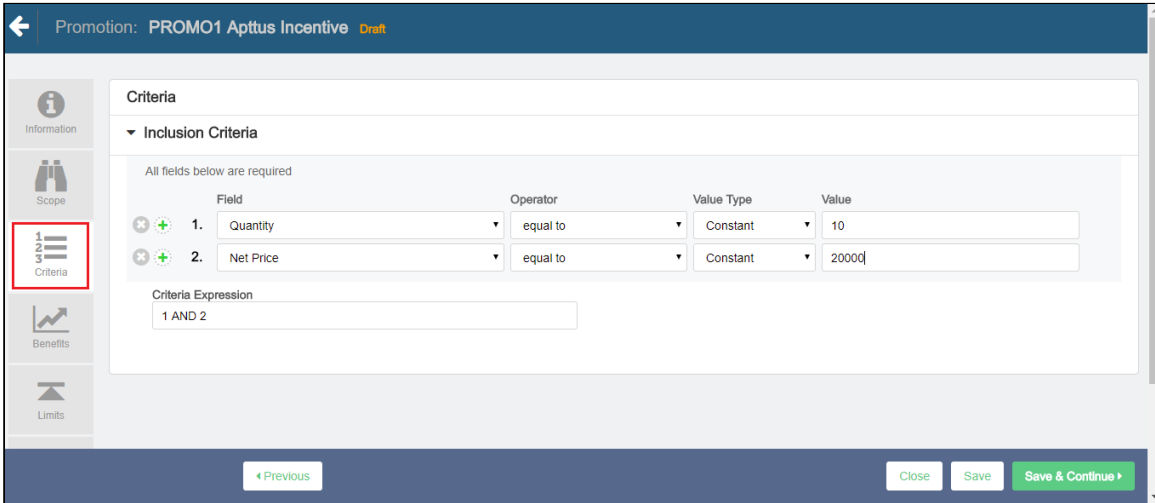List<IncentiveCoupon__c> coupons =
IncentiveCouponFactory.createSObjectsForIncentive(incentiveSO.Id, 5);
        System.assertEquals(coupons.size(), 5);
        for (IncentiveCoupon__c couponSO : coupons) {
            System.debug('couponCode=' + couponSO.CouponCode__c);
        }
```

You can view the resulting list of coupons on the Accounts page.

> ⚠ The Coupon Code is displayed in the product details page. If there are multiple coupons, all coupons codes are saved as comma separated.

Viewing the Summary of a Promotion

After you have configured the promotion record, you can use the Summary tab to view all the details. Something similar to the image below is displayed when you click Summary tab.

Defining Batch Size to Process Line Items with Promotions

During the pricing, you can process the promotions in batches for a cart with a large number of line items. You can add the **APTS_IncentivePricingBatchSize** Admin Setting to define the batch size of line items. The batch processing is only supported for **Buy X Get X** and **Buy X Get Y** promotions with **Auto Apply?** enabled. You can use this feature for both regular and async pricing.

To define batch size to process line items with promotions

1. Go to **All Tabs** > **Admin**
2. Click **New**, to create a new record. The New Admin page is displayed.
3. Fill in the following details

Field	Value
Name	*APTS_IncentivePricingBatchSize*

Field	Value
Value	Enter the number of line items to be processed in a batch. ⓘ You must define the batch size based on the complexity of the cart and the number of incentives applied on the line items.
Code	Leave the field blank

4. Click **Save**.

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.

Configuring the Apply Promotion Window

The **Apply Promotion** window is displayed on the cart when the Sales Rep clicks **Apply Promotions** on the line items or the cart. You can configure the fields displayed on the window in the Config Settings.

To configure the Apply Promotion Window

1. Go to **All Tabs** > **Config Settings** > **Display Column Settings.**
2. From the **Display Type** drop-down, select **Apply Promotions**.
3. Select th flow you want to configure in **Flow**.
4. You can edit the existing fields or add new fields to the window. To add a new fields click add icon(➕).
5. The following fields are available to you:

Field	Description
Incentive Code	Displays the code of the incentive on the window
Incentive Name	Displays the name of the incentive on the window
Sequence	Displays the sequence of the incentive.

Field	Description
Expiration Date	Displays the date of the expiration of the incentive.
Description	Displays the description of the incentive that you added while creating the incentive.
Combine with Other Promotions	Indicates whether the promotion can be combined with other promotions.

The fields **Incentive Code**, **Incentive Name**, **Sequence**, and **Expiration Date** are displayed by default.

6. Click **Save**.

Use Case: Configuring a Promotion for the Total Price on the Cart

This use case describes how to configure a promotion on an order level. You can configure a promotion such that the user gets a discount of 10% if the total price of the cart is greater than 500$.

This section provides an overview of how to configure promotion for the use case and list high-level steps of configuring the promotion for an order. For detailed steps about configuring a promotion, refer to Configuring Promotions.

Before you begin

Understand Promotions. Refer to Managing Promotions.

To Configure a Promotion for an order

1. Go to the Salesforce App Menu and select **Conga Incentive Setup**.
2. Click **Incentives** > **New**.
3. On the Incentives information page, define values for the fields as described in Defining Promotions and **Save**. Ensure that you select **Summary Group** in the **Reference Data Source** Field.

4. In the **Criteria** tab, enter the line on which the promotion is applied. Select the following:
   - **Field**: Totals
   - **Operator:** Greater Than
   - **Value Type**: Constant
   - **Value:** 500

   > ⚠ You can also use Field as **Line Type** that has pre-populated values such as Total, Subtotal, Group Total, Category Total and Adjustment to configure the promotions for this use case.

5. In the **Benefits** tab, enter the following information:
   a. **Rate Type**: Single Rate
   b. **Rate Value Type**: % Discount
   c. **Adjustment Applies to**: Total price
   d. **Adjustment Amount**: 10

Result:

Click **Apply Promotions** on the Cart. The discount is visible on the Total Price.

Configuring Promotions on Benefit Products in Buy X Get Y

You can use the Admin Setting *APTS_EnableMultiBenefitItems* to control the ability of the Sales rep to apply promotions to multiple line items of the benefit product (Y). By default, this setting is disabled and CPQ applies the promotion only to the first line item of the benefit product (Y). When you enable this setting, CPQ applies the promotion to all line items of the benefit product (Y).

To configure promotions on benefit products in a Buy X Get Y scenario

You must follow the steps below to configure this setting:

1. Go to **All Tabs** > **Admin**
2. Click **New**, to create a new record. The New Admin page is displayed.
3. Fill the following details

Field	Value
Name	*APTS_EnableMultiBenefitItems*
Value	*true* or *false*
Code	Leave the field blank.

4. Click **Save**.

Applying Promotions for eCommerce Orders

You can apply the promotions that you define, for orders placed on Conga eCommerce, using APIs.

The following table lists 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 promotions, set the comma separated list of PULL promotion codes (Incentive codes) on the Product Configuration object in the field "Apttus_Config2__CouponCodes__c". After setting the values, reprice the cart using the "updatePriceForCart" API. CPQ 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 the order is accepted or confirmed by the customer. However, individual implementations can decide when to call this API.

Analyzing Promotions

For analysis purpose, 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.

Configuring Assets

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 Conga provides for each challenge.

Challenges with Asset Management	Conga-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 Conga CPQ with Conga 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.
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.

Challenges with Asset Management	Conga-provided solutions
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 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 Conga 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.
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 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 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.

About Asset and Asset Line Item

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 Conga 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 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-Based Ordering Packages Required

For implementing Asset-based Ordering, follow the instructions of installing a package from the Install Center explained in the Installing CPQ section. 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	Conga Quote Management	CPQ
2	Conga Configuration & Pricing	CPQ
3	Conga CPQ Api	CPQ
4	Conga Quote Configuration Integration	Integrations
5	Conga Quote Asset Integration	Integrations

For the Contracts Flow,

Order	Package	Install Center tab to access the package
1	Conga Contract Lifecycle Management	Contract Management
2	Conga Quote CLM Integration	Integrations
3	Conga CLM 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. Configuring Flow Management Settings
3. Configuring Installed Products Settings
4. Configuring Asset Management Flows
5. Customizing Display Actions for the Installed Products page
6. Defining the look and feel of the Installed Products page
7. Configuring Asset-based Pricing
8. Configuring Lookup Field Settings

Optional Configuration

The following configuration steps are optional. You may implement these steps based on your business requirements.

1. Defining the Actions you want to allow for each Asset Line Item
2. About Integration of Assets with Contracts
3. Creating an Approval Workflow for **Renew**, **Change**, **Swap**, and **Terminate**
4. Configuring Mass Update for Assets
5. Creating a Replacement Rule to enable the Swap feature
6. 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.

About Asset Management Flows

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.

Asset-based ordering is particularly useful when your business offers complex service products, such as phone services and equipment.

On the Installed Products page, you can Renew, Terminate, Swap, and Change an existing standalone, fixed, or a configurable bundle asset. For each transaction involving ABO actions, the system creates a new order while you are navigating on the Cart. The status of the Asset Line Item is not changed until you finalize the Cart and activate the Order containing that Asset Line Item.

Use case

For a sales-driven industry with products such as equipment (new purchase) and service (purchased for an equipment), you might want to achieve the following:

- As a manufacturing sales representative, for a service flow, you want to view the **Renew** and the **Terminate** button. For an equipment flow, you want to view the **Change** and the **Swap** button.
- As a subscription sales representative, you want to see the **Renew** button only when I make a purchase for a quote of type *renew*. For an add-on quote or an upgrade quote, you do not want to see the **Renew** button. For example, managing the renewal of a magazine subscription.

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 the 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 Lifecycle 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. 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 a 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.

Configuring Asset Management Flows

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** checkbox and select it. For more information, Configuring Proposal Settings.
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>&launchState=assets&auto
Finalize=true&activateOrder=true&retId=" & Id , IMAGE("", "Asset Manager"),"_se
lf")
```

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** checkbox and select it. For more information, see **Comply System Properties** section in Configuring Custom Settings.
4. Click **Save.**

⚠ We recommend you to refrain from enabling **Auto Activate Order** from **Comply System Properties** as well as **Proposal System Properties**, simultaneously.

For detailed information on the behavior of these two checkboxes, see About Integration of Assets with Contracts.

Configuring 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 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. You can turn off asset pricing from the custom setting **Config Asset Pricing Default** based on asset line status on the cart load.
- In the CPQ Admin UI, you can use PAV extension fields to create asset pricing criteria. Criteria fields can use attributes only from the out-of-the-box product attribute value (PAV) extension objects. CPQ does not support custom PAV extension objects.

✅ **Use Case 1**

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. Navigate to **Setup** > **Develop** > **Custom Settings** > **Config Asset Pricing Criteria Fields**. The Config Asset Pricing Criteria Fields page is displayed.

5. Click **New**. The Config Asset Pricing Criteria Fields Edit page is displayed.

6. In the Name field, enter a name for the asset pricing criteria.

7. In the **Criteria Field Names**, enter Apttus_Config2__Quantity__c (API name of the **Quantity** field).

8. Keep the **Line Status** field blank.

9. From the CSR flow, select this asset 'B' and perform the ABO operation of **Change**. You will see the asset price of $1000 on the cart.

10. Change the **Quantity** to '5' and you will see that asset price is now set to $1200 (from the PLI).

> ✅ **Use Case 2**
>
> 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. Navigate to **Setup** > **Develop** > **Custom Settings** > **Config Asset Pricing Criteria Fields**. The Config Asset Pricing Criteria Fields page is displayed.
>
> 5. Click **New**. The Config Asset Pricing Criteria Fields Edit page is displayed.
>
> 6. In the Name field, enter a name for the asset pricing criteria.
>
> 7. In the **Criteria Field Names**, enter Apttus_Config2__Quantity__c (API name of the **Quantity** field).
>
> 8. In the **Line Status** field, enter Renewed.
>
> 9. From the CSR flow, select this asset 'B' and perform the ABO operation of **Change**. You will see the asset price of $1000 on the cart.
>
> 10. Ensure that the line status changes to Renewed.
>
> 10. Change the **Quantity** to '5' and you will see that the asset price is still $1000. The asset pricing criteria works for change in quantity, but the asset pricing criteria does not work for Renewed. Though you change quantity, you will still see the asset pricing.

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 Net Unit Price (Net price / Quantity*term)
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	

Standalone Asset	Asset pricing Enabled	Base Price	Quantity	Term	Extended Price	Discount	Net Price	Asset Net Unit Price (Net price / Quantity*term)
On Change (asset)	True	85.416 (fetched from Asset Line Item)	4	12	4100	100	4000	83.333

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 Net Unit Price (Net price / Quantity*term)
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	200 (fetched from Price List Item)	2	12	4800	100	4700	195.8333

Overriding the Default Asset Pricing for Line Statuses Based on ABO Actions

CPQ enables you to override the default asset pricing for certain ABO actions and the corresponding line statuses on the cart. In that case, CPQ applies price from price list items.

Prerequisites

Ensure that the **Default Asset Pricing Indicator?** is turned on in **Config System Properties**.

To override default asset pricing for a line status

1. Navigate to **Setup** > **Develop** > **Custom Settings**.
2. Click **Manage** next to **Config Asset Pricing Default**. The Config Asset Pricing Default page is displayed.
3. Click **New**. The Config Asset Pricing Default Edit page is displayed.
4. In the **Name** field, enter the name of status for which you want to override the default asset pricing.
5. Clear the **Default Asset Pricing Indicator** checkbox to override the default asset pricing configured through the **Default Asset Pricing Indicator?** setting in Config System Properties. For more information, see Config System Properties.

   > ⓘ If you select this setting, the global setting **Default Asset Pricing Indicator?** in Config System Properties takes precedence. In that case, CPQ makes asset pricing as default for asset line items.

6. In the **ABO Action** field, enter the ABO action associated with the setting. Supported values are:
   - Renew
   - Change
   - Swap
   - Terminate
   - Change: Configuration
   - Change: Merge
   - Change: Split
   - Change: Quantity
   - Relate
   - Relate Component

When you come to the cart through these ABO actions, for the line statuses you configured above, you will see that the value is from price list item, not from the asset pricing because you have overridden the default setting.

> ⓘ CPQ applies Price Escalator on new ramp lines created during ABO operations regardless of the **Default Asset Pricing Indicator?** setting (true or false). A new ramp line always gets its value from the price list item (PLI) pricing. For more information, see Use Case: Applying Price Escalator on New Ramp Lines During ABO Operations.

Use Case: Switching Between Asset Price and PLI Price

This use case describes how you can switch between asset price and PLI price based on line status of cart line items and ABO actions.

Multiple ABO actions can result in the same line statuses for cart line items. For example:

ABO Action	Line Status
Renew and Change (for line items when **Price Type** = One Time and **Enable One Time Change** = False)	Existing
Renew and Change (for line items when **Price Type** = One Time and **Enable One Time Change** = True)	Amended

For the line status "Existing", you want to switch between asset price for "Renew" and PLI price for "Change".

Configure Asset Pricing for Renew

You must configure asset pricing for all ABO actions through the global setting **Default Asset Pricing Indicator?**.

1. Go to **Setup** > **Develop** > **Custom Settings** > **Config System Properties**.
2. Click **Manage** and click **Edit** next to System Properties.
3. Select the **Default Asset Pricing Indicator?** checkbox.
4. Click **Save**.

CPQ will apply asset price on cart line items in "Existing" status during the Renew operation because the global setting **Default Asset Pricing Indicator?** is set to True, which makes asset price as default for asset line items.

Configure PLI Price for Change

You must override the global setting **Default Asset Pricing Indicator?** for for the Change operation and the corresponding cart line item status "Existing".

1. Navigate to **Setup** > **Develop** > **Custom Settings**.
2. Click **Manage** next to **Config Asset Pricing Default**. The Config Asset Pricing Default page is displayed.
3. Click **New**. The Config Asset Pricing Default Edit page is displayed.
4. In the **Name** field, enter "Existing". This is the line status for which you want to override the default asset pricing.
5. Clear the **Default Asset Pricing Indicator** checkbox.
6. In the **ABO Action** field, enter "Change".

When you perform Change operation and come to the cart, for the line items in Existing status, you will see that the value is from the price list item, not from the asset pricing because you have overridden the default setting. However, when you perform the Renew operation for the same asset, you will see for the cart line item in "Existing" status that the value is from asset pricing.

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. Accessing the Installed Products Page
2. Configuring Column Settings for the Installed Products Page.
3. Defining the Number of Products Displayed on the Installed Products Page.
4. Configuring Asset Visibility on the Installed Products Page.
5. Creating Search Filters for 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.

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 Conga 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=ngflow&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&launchState=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.

Configuring 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, based on the user's 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.

Configuring 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) displayed on the Installed Products page. You can also see critical attribute values at the main bundle level when expanding a bundle product on this page.

> ⓘ Attributes for which you select the **Is Primary** check box on the Attribute Configuration page are called Critical Attributes.
> CPQ displays the critical or primary attributes associated with the main bundle product on the Installed Products page. However, CPQ does not display the critical or primary attributes associated with sub-bundles on the Installed Products 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. For more information, see Asset Line Item Callback Class.

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.

Creating Search Filters for 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 checkbox 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.

Displaying Assets from Multiple Accounts on the Installed Products Page

By default, the assets that CPQ displays on the Installed Products page are from a single account related to the current quote. As an administrator, you can configure CPQ to display assets from multiple accounts, which are related to an opportunity, on the Installed Products page. You must complete the following configurations to display assets from multiple accounts:

1. Enable the admin setting *APTS_EnableInstallBaseFilteringAcrossAccounts*.
2. Configure a custom callback class.

To enable asset filtering across accounts

1. Go to **All Tabs** > **Admin**.
2. Click **New** to create a new record. The New Admin page is displayed.
3. Fill in the following details:

Field	Value
Name	*APTS_EnableInstallBaseFilteringAcrossAccounts*
Value	Enter one of the following values: • true: CPQ displays assets from multiple accounts. • false: This is the default value. CPQ displays assets from a single account related to the quote. ⓘ It is recommended to keep the value as False. Set it to True only if there is a business requirement to fetch assets from multiple accounts.
Code	Leave the field blank.

4. Click **Save**.

To configure a custom callback class

1. Configure a custom callback class to fetch assets from the list of accounts. The sample code below enables you to fetch assets from multiple accounts.

```
/*
 * This class is used in Asset Line Item callback to show asset on Cart page.
 */
global with sharing class CurrentUserFilter_AssetLineItemCallback implements
 Apttus_Config2.CustomClass.IAssetLineItemCallback4
    {

    private List<String> assetSearchScope = null;
    private String assetSearchFilter = null;
    private ID userId = null;
```

```
    /**
     * 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(Apttus_Config2.ProductConfiguration cart, String
assetSearchFilter, List<String> assetSearchScope) {
        this.userId = cart.getConfigSO().CreatedById;
    }


    /**
     * 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) {
        // all Asset Lines created by the current user
        return 'CreatedById = \'' + userId + '\' AND CreatedDate > LAST_MONTH';
    }

    global Boolean validateAssetTermination(Set<ID> assetIds, Set<ID>
accountIds, Date eDate) {
        return true;
    }

    global Date getAssetTerminationDate() {
        return Date.newInstance(2018,01,01);
    }

    /**
     * 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(Apttus_Config2.CustomClass.ActionParams params)
   {
          return getQueryFilter(null);
      }


      /**
       * 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() {


      }
   }
```

2. Copy the custom callback class in the **Config Custom Classes** under **Asset Line Item Callback Class**. For more information, see Asset Line Item Callback Class.

This table shows how CPQ behaves with different combinations of the admin setting and the custom callback class.

APTS_EnableInstallBaseFilteringAcrossAccounts	Custom Callback Class	CPQ Behavior
True	Active	CPQ displays assets from multiple accounts.
False	Active	CPQ displays assets from the current account.
True	Inactive	CPQ displays assets from the current account.

APTS_EnableInstallBaseFilteringAcrossAccounts	Custom Callback Class	CPQ Behavior
False	Inactive	CPQ displays assets from the current account.

Configuring Lookup Field Settings for Asset-Based Ordering

Each new lookup field shown in the 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*.

Defining Allowed 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.

Defining Flow-based ABO Actions

Depending on your business units and type of purchase, the sales administrator can hide the action buttons on the Installed Products page. The ability to hide the action buttons is available with the flow settings defined for a Quote/Proposal. This is supported in conjunction with hiding the action buttons at the asset line item level (explained above) and global level (refer to Installed Product Settings).

Use case

For a sales-driven industry with products such as equipment (new purchase) and service (purchased for an equipment), you might want to achieve the following:

- As a manufacturing sales representative, for a service flow, you want to view the **Renew** and the **Terminate** button. For an equipment flow, you want to view the **Change** and the **Swap** button.
- As a subscription sales representative, you want to see the **Renew** button only when I make a purchase for a quote of type *renew*. For an add-on quote or an upgrade quote, you do not want to see the **Renew** button.

To define flow-based ABO action

You must have configured Installed Product settings and Flow settings.

1. Create a new flow using Flow Settings or choose an existing flow for your Quote/ Proposal.
2. Go to **Setup > App Setup > Develop > Custom Settings > Config System Properties > Manage**. You will see an existing record named *System Properties*.
3. Create a new record with the same name as your Flow name.
   For example, if you have created a flow named *ngCPQflow*, enter *ngCPQflow* in the **Name** field.
4. In the **Hide Asset Actions** field, enter the values of action buttons that you want to hide for your flow. In our example, for a service flow, enter *Change* and *Swap*. For an equipment flow, enter *Renew* and *Terminate*.
5. Click **Save**.

About Integration of Assets with Contracts

CPQ allows you to manage assets through contracts. A contract or an agreement is a legally binding arrangement between two or more entities.

Prerequisites

- For the Assets and Contracts integration to work, the following packages must be available in your org:
  - Conga Contract Lifecycle Management
  - Conga Quote CLM Integration
  - Conga CLM Configuration Integration
- The **Auto Create Order** setting in **Comply System Properties** must be enabled if you want to create an order and asset as soon as an agreement is activated.

How asset and contract flows work

There are two process flows you must consider:

- Contract + Proposal Flow: In this flow, both Contract and Quote/Proposal are involved.
- Contract Flow: In this flow, only Contract is involved (no Quote/Proposal). The assets are created through contracts/agreements using the normal contract life cycle flow. After you finalize and activate a contract (containing agreement line items), the corresponding asset line items are created.

The following table explains the CPQ behavior when the **Auto Create Order** setting is configured.

Flow	Auto Create Order (Proposal System Properties)	Auto Create Order (Comply System Properties)	CPQ behavior
Contract + Proposal Flow	False	True	A Quote/Proposal is created followed by a Contract. The Asset Line Items and Order Line Items are created when you activate a Contract.

Flow	Auto Create Order (Proposal System Properties)	Auto Create Order (Comply System Properties)	CPQ behavior
Contract Flow	False	True	Contract is created without a Quote/Proposal. The Asset and Order Line Items are created when you activate a Contract.

> ⓘ It is recommended to refrain from enabling **Auto Activate Order** from **Comply System Properties** and **Proposal System Properties** simultaneously.
> The Conga Contract Lifecycle Management package version must be 8.4.0325 (8.325.1) or higher to use the features mentioned above.

For more information, see Managing Assets through Contracts.

Creating Approval Workflows for Asset-Based Ordering

You can enable 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 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 in the guide.

Once enabled, you see **Submit for Approval** button on the Cart, every time you perform action on the asset.

Configuring Mass Update for Assets

From the **Allow Mass Change** setting on the **Installed Products Settings** under **Custom Settings** page, you can enable or disable mass changes for *must configure* assets on the Installed Products page.

Action	Allow Mass Change	Type of the Asset	Type of selection	Behavior
Change	Unchecked	Must Configure	Single Select	Launch configure product page
Change	Unchecked	Must Configure	Multi Select	Disable Change
Change	Unchecked	Combination of Must configure and not must configure	Multi Select	Disable Change
Change	Unchecked	Not must configure	Single Select	Load into cart directly
Change	Unchecked	Not must configure	Multi Select	Load into cart directly
Change	Checked	Must Configure	Multi Select	Load into cart directly
Change	Checked	not Must Configure	Multi Select	Load into cart directly
Change	Checked	Must Configure	Single Select	Load into cart directly
Change	Checked	not Must Configure	Single Select	Load into cart directly

Creating Replacement Rules 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.

> ⓘ CPQ enables replacement rules only on the ABO carts, for swap and split-swap functionalities.

For more information, see Managing Constraint Rules

Configuring the Custom Attributes Page

You can use a pre-defined custom attribute page while navigating from the Installed Products (Assets) page. By default, CPQ chooses the *Apttus_Config2__ProductAttributeDetail3* Visualforce page. This configuration is useful when you have designed a custom attribute page with the new look and design and you want to use it while configuring the attributes.

This capability allows customers to have their own UI implementation for the product configuration experience. You can enable Sales and Customer Service representatives to select options for assets and capture product attribute details for assets.

To customize your flow settings, perform the following steps.

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**.

Configuring Auto Renewal of Assets

CPQ allows Sales Representatives to auto-generate renewal quotes based on the fulfillment of order or the expiration of subscription. Through renewal quotes, they can forecast and analyze the sales pipeline. With renewal quotes, they get the finer visibility into the pricing and configuration of assets within the renewal pipeline. The benefits of auto renewals are:

- Accurate forecasting on the asset state and sales pipeline
- Faster and simpler processing with automated renewal
- Automatic closure of opportunity on asset expiry

Types of Asset Renewal Modes

There are two types of renewal modes, defined in **Renewal Execution Mode**:

- Auto
- OnDemand

> ⓘ It is recommended not to use these renewal modes interchangeably.

The following diagram explains how auto renewals work:

Configuring Renewal Settings

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. Currently, *Proposal* and *Agreement* are supported.
Renewal Default Price Book	Enter a Renewal Price Book name which you want to associate with renewals.

Field	Description
Renewal Execution Mode	Indicates if the Renewal of Asset Line Items must happen automatically or based on 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*.
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.

Field	Description
Create Renewal Opportunity	Indicates if you want to create renewal opportunities along with renewal quotes. By default, this checkbox is not selected. This is to prevent the system from running into locking issues when an account has a large number of assets and the batch process creates the renewal opportunities for all these assets.
	When set to *true*, the Create Renewal Opportunity scheduled job runs and creates the renewal opportunities along with renewal quotes.
	When set to *false*, the Create Renewal Opportunity scheduled job runs and does not create the renewal opportunities and creates only the renewal quote.

AutoSync Asset Changes to an Opportunity-Based Renewal Quote

When you select the **Sync Assets to Quote** flag on a proposal, whenever 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 in the renewal quote.
> - Opting out of a renewal opportunity does not pull the line out from the auto-renewal quote.
> - For Assets having **Auto Renewal Type** = *Evergreen*, no renewal opportunities and quotes are created and the **Renew** button is greyed out.

Adding Picklist Values to the Proposal Object

1. Go to the **Proposal** object
2. Click to open Proposal from the **Record Types**.
3. Edit the Picklist **ABO Type** from the list.
4. Add *New*, *Add-on*, and *Renewal* to the list of **Selected Values**.

Renewal Email Templates

To set up the email templates:

1. Navigate to **Setup** > **Communication Templates** > **Email Templates**.
2. Create templates for Asset Renewal Notification.
3. 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.

To schedule a 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.

1. Go to **Setup** > **Develop** > **Apex Classes**.
2. Click **Schedule Apex**. The Schedule Apex page is displayed.
3. In the **Job Name** field, enter a name for the job.
4. In the **Apex Class** lookup, search and select *AssetRenewalJobScheduler*.
5. Under **Schedule Apex Execution**, select the **Frequency**, **Start** date, **End** date, and **Preferred Start Time** for the job.
6. Click **Save**.

To monitor the job, go to **Setup** > **Monitoring** > **Apex Jobs**. You can see *AssetRenewalJobScheduler* in the list of jobs. You will get an email notification after this job is successfully completed.

To schedule a batch job for creating renewal opportunity

You must schedule the batch job *CreateRenewalOpportunityJobScheduler* for the first time. This job runs every 5 minutes after you have scheduled it.

1. Go to **Setup** > **Develop** > **Apex Classes**.
2. Click **Schedule Apex**. The Schedule Apex page is displayed.
3. In the **Job Name** field, enter a name for the job.
4. In the **Apex Class** lookup, search and select *CreateRenewalOpportunityJobScheduler*.
5. Under **Schedule Apex Execution**, select the **Frequency**, **Start** date, **End** date, and **Preferred Start Time** for the job.
6. Click **Save**.

To monitor the job, go to **Setup** > **Monitoring** > **Apex Jobs**. You can see *CreateRenewalOpportunityJobScheduler* in the list of jobs. You will get an email notification after this job is successfully completed.

Configuring Asset Renewal in Auto Renewal Mode

When **Renewal Execution Mode** is set to *Auto*, CPQ automatically renews an existing quote for all assets it has, as soon as the order is activated. You may want to renew an existing opportunity in Auto renewal mode when:

- You require renewal quotes for forecasting
- You must work on renewal quotes in advance

The following diagram describes the flow of **Auto** renewals.

Prerequisites

- Set the **Renewal Business Object Type** to Proposal.
- Set the **Renewal Execution Mode** to *Auto*.
- Define the **Renewal Group Fields**.
- Schedule the AssetRenewalJobScheduler batch job.

For more information, see Configuring Renewal Settings.

Configuring Asset Renewal in OnDemand Renewal Mode

When **Renewal Execution Mode** is set to *OnDemand*, CPQ automatically renews all assets on the account (possibly from different quotes) based on the defined **Renewal Lead Time** after you click a **custom button** on the account. You may want to renew an existing opportunity in OnDemand renewal mode:

- During one-time legacy asset data migrations
- During one time renewal quote creation for existing active asset base
- When minimal changes are required on the renewal quotes because of any ABO changes

The following diagram describes the flow of **OnDemand** renewals. In this diagram, **Renewal Lead Time** is 120 days.

Over a period of time, you sell some products/services through different quotes, which might have varying end dates. After the original quotes are converted into orders and the order are activated, assets are generated. All assets are visible on the account. When you want

to renew the assets that will expire in 120 days from today, you click the **custom button** on the account. CPQ compares the asset line items' end dates with the **Renewal Lead Time** defined and triggers the creation of renewal quotes for all those assets (from different quotes) that will expire in 120 days.

Prerequisites

- Set the **Renewal Business Object Type** to Proposal.
- Set the **Renewal Execution Mode** to OnDemand.
- Define the **Renewal Lead Time**.
- Create the **custom button** on the account if you want to renew all assets in the account based on the defined Renewal Lead Time.
- Define the **Renewal Group Fields**.
- Schedule the AssetRenewalJobScheduler batch job.
- 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**. If you want to renew assets from a quote based on the defined Renewal Lead Time, configure a custom link on the quote.

- 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();
```

For more information, see Configuring Renewal Settings.

Configuring Agreement-Based Renewals

CPQ supports Agreement-based renewals with a custom setting called **Renewal Business Object Type**, with value as *Agreement*. The entire functionality supported with Proposal object is supported with Agreement object. For information on how this feature works, Managing Assets in the Contract Flow.

For auto closure of renewal agreements, the **Status Category** of the agreement record should be one of the following:

- Request
- In Authoring
- In Signatures
- In Filing

The renewal agreement set for auto closure expires within 24 hours.
For manual submission of agreement-based renewals, you must create a new custom link for Agreement object, pointing to `Apttus_CMConfig_AgreementAssetRenewalSubmit` Visualforce page.

For more information on configuring data required to trigger renewal agreement creation, see Configuring Renewal Settings.

About Extending the Lead Time Functionality

A renewals administrator at a major healthcare company can be contractually mandated to create the renewal suite 90 days before the asset expiry for a large company and 30 days before asset expiry for a small company.

The CPQ Salesforce Installed products Settings such as Lead Time and Renewal Execution Setting is OnDemand a renewal quote is generated for all accounts globally. However, if you want to generate renewals for a list of accounts for a particular lead time, you can use the following construct in the developer console.

```
// specify a lead time to override setting value
Integer leadTime = 1000;
// gather account ids
List<ID> accountIds = new List<ID>();
accountIds.add('0014C00000FQSuz');
accountIds.add('0014C00000FQSuu');
// instantiate the OOTB controller and pass in the account ids
Apttus_Config2.AssetRenewalSubmitController baseController = new
 Apttus_Config2.AssetRenewalSubmitController(leadTime, accountIds);
// submit the job
ID jobId = baseController.doSubmitJob();
```

You can use the following construct to renew assets that belong to separate list of accounts for different lead times:

```
// create map to be used as the constructor argument
Map<Integer, List<ID>> accountIdsByLeadtime = new Map<Integer, List<ID>>();
// gather account ids to be renewed with a leadTime of 400
Integer leadTime = 1000;
List<ID> accountIds = new List<ID>();
accountIds.add('0014C00000FQSv9');
// add leadTime as key and accounts as value to map
accountIdsByLeadtime.put(leadTime, accountIds);
// gather account ids to be renewed with a leadTime of 1000
leadTime = 60;
accountIds = new List<ID>();
accountIds.add('0014C00000FQSv4');
// add leadTime as key and accounts as value to map
accountIdsByLeadtime.put(leadTime, accountIds);
// instantiate the OOTB controller and pass in the account ids
Apttus_Config2.AssetRenewalSubmitController baseController = new
 Apttus_Config2.AssetRenewalSubmitController(accountIdsByLeadtime);
// submit the job
ID jobId = baseController.doSubmitJob();
```

After executing the construct, view changes on the Temp Renew page. New records are created in the Temp Renew page.

Configuring Service CPQ

The Service CPQ feature caters to customer requirements related to services for assets (equipment purchased from Conga). A Sales Representative associates a service product with an asset. You can perform the following configurations to set up Service CPQ:

- Create service products by selecting **Product Type** as *Service*. For more information, see Creating Products.
- Create constraint rules to display relevant services on the Service Catalog for a selected asset. Service products and assets are associated based on the eligibility rules, which is the framework for service eligibility at large. The eligibility rules, which are client-side constraint rules, play a vital role in guiding the association between a service product and an asset.
  For example, when the user selects a specific asset, you want CPQ to hide some services on the Service Catalog. In the eligibility constraint rule criteria, you must define the Exclusion type constraint rule. The constraint rule action intent is scoped to other service products under a product family or product group to make them ineligible (block them) when the rule is triggered. For more information, see Creating Constraint Rule Conditions and Creating Constraint Rule Actions.
- The **Relate** button on the Installed Products page is displayed by default. You can change the label and alignment of the **Relate** button from **Display Action Settings > Display Type =** *Installed Products* for your flow name.
- Configure the **Submenu Actions** setting to display the **Relate Component** drop-down option on the **Relate** button. For more information, see Configuring Installed Products Settings.
- Set the **Relate Action Criteria Fields** to enable relate asset action. For more information, see Configuring Installed Products Settings.
- Create a price dimension for Service CPQ and use that price dimension in the required price matrix. For more information, see Creating Price Matrices and Dimensions.
- Define the **Service Price Distribution Method** setting to define the mode used for service pricing, at Setup> Develop> Custom Setting > Config System Properties. For more information, see Configuring Custom Settings.
- Define the **Service Line Split Criteria** setting to allow users to clone a service with some or all related line items using in the Service Cart, at Setup> Develop> Custom Setting > Config System Properties. For more information, see Defining Service Line Split Criteria.
- Configure the **Show Service Coverage** setting to display service coverage for a primary service, at Setup> Develop> Custom Setting > Installed Product Settings. For more information, see Configuring Custom Settings.

- Configure the columns to be displayed on the **Related Purchases** tab on the Installed Products page, using **Display Column Settings > Display Type** = *Related Purchases* for your flow name. For more information, see Configuring Display Column Settings.
- Configure the **Related Line Item Fields** setting in **Field Set Settings** to display the required fields on the Related Line Items pop-up. For more information, see Configuring Field Set Settings.
- Configure the **Asset Line Fields For Selected Assets** setting in **Field Set Settings** to display the required fields on the Selected Assets pop-up. For more information, see Configuring Field Set Settings.

> ⓘ It is mandatory to display the **Quantity** field on the Selected Assets pop-up for CPQ to calculate the correct number of assets selected on the Installed Products page.

- Configure how CPQ should display service product attributes on the Configuration page, while creating product attribute groups. For more information, see Creating Product Attribute Groups.

Configuring Service Pricing

The service pricing flow is as follows:

- Create as many dimensions in the matrix as the number of your asset dimensions. Set up the matrix pricing for the service bundle and the service component individually. CPQ relies on the first match for the service price. Specify entries with less ambiguity (wild card) at the beginning of the matrix. For more information, see Creating Price Matrices and Dimensions.
- Configure the **Service Price Distribution Method** setting at Setup > Develop > Events > Custom Settings > Config System Properties. For more information, see Configuring Custom Settings.
- For **Relate Component - Bundle to Bundle scenario**: Related line items created only at the bundle level for the equipment items selected. Service is priced individually at the service bundle level and at the service option level for each related line item Individual service prices are then averaged to obtain the base price of the service line item and are then further rolled up to arrive at the price for the service line.
- For **Relate Component - Bundle to Component scenario:** Related line items created only at the component level for the selected components of the equipment (Note the behavior change from the previous releases – no related line items are created for the equipment bundles – you can see bundle associated with the component as an associated column on the same related line item). Service is priced individually at the service bundle level and the at the service option level for each

related line item. Individual service prices are then averaged to obtain the base price of the service line item and are then further rolled to arrive at the price for the service line.

- If there is no matching entry in the price matrix, it will take price as zero.
- CPQ derives the weightage of the service line from its contribution.
- Define split criteria to enable users to clone a service with some or all related line items using in the Service Cart. For more information, see Defining Service Line Split Criteria.

Defining Service Line Split Criteria

The Split feature allows users to clone a service with some or all related line items using in the Service Cart. The Split feature is available only at the service bundle level. You can define split criteria that apply globally or create split criteria for the specific flow.

> ⓘ The **Service Line Split Criteria** setting only supports API names of the fields created on the Related Line Item object only. These fields can be formula fields and can fetch their values from Asset Line item and Product through formula expressions. CPQ supports the following types of custom fields on the Related Line Item object for split:
> - Text
> - Picklist
> - MultiPicklist
> - Formula (return type: Text)

To define global split criteria

1. Go to **Setup> Develop> Custom Setting > Config System Properties > Service Line Split Criteria.**
2. Specify the API names of the fields from the Related Line Item object on the basis of which you want to split the service.
3. Click **Save**.

The Split functionality is enabled on the Service Cart page.

To create a flow-based split criteria

1. Remove the values in the **Service Line Split Criteria** setting in Config System Properties.
2. Create a new config system property with the name that matches with the flow name of the **Configure Product** button.
3. Add the split criteria for the new Config System Property.
4. Click **Save**.

The **Split** button is displayed only for the cart grid that is created based on the flow name.

Synchronizing Related Line Items with Opportunity Line Items

As an administrator, you can enable synchronization of related line items with opportunity line items. The following are the methods of synchronization:

> ⓘ   This feature is used only in Service CPQ.

Auto-Synchronization

1. Enable the **Auto Sync With Opportunity** setting in the Proposal System Properties custom setting.
2. Create a lookup field with API name *LineItemCollId__c* on the *Opportunity Product* object with a lookup relationship to the *Related Opportunity Line Item Coll* object.

3. Display related lists **Related Opportunity Line Item Collection** and **Related Opportunity Line Items** on the opportunity layout.

4. Ensure that the **Primary** checkbox is selected on the quote and the opportunity has a price book associated with it.

5. When you finalize the quote, you observe that:

   a. Proposal line items synchronize on the *Opportunity Product* object.

   b. The lookup field with API name *LineItemCollId__c* is populated with a related item collection ID.

   c. Each record in the *Opportunity Product* object has a unique related Item collection ID.

   d. The *Related Opportunity Line Item Collection* object links the opportunity product to the related line items associated with it. For every record in the *Opportunity Product* object, which has related lines associated with it, a unique *Related Opportunity Line Item Collection* is created.

   e. In the *Related Opportunity Line Item Collection* related list, the *Related Item Coll Id* of all collection records is displayed.

   f. The *Proposal Related Line Items* are synchronized with the *Related Opportunity Line Items* object and are displayed in its related list.

Manual Synchronization

1. Disable the **Auto Sync With Opportunity** setting in the Proposal System Properties custom setting.
2. Create a lookup field with API name *LineItemCollId__c* on the *Opportunity Product* object with a lookup relationship to the *Related Opportunity Line Item Coll* object.

Opportunity Product Custom Fields & Relationships		New Field Dependencies		
Action	**Field Label**	**API Name**	**Installed Package**	**Data Type**
Edit | Replace ⬇	Approval Status	Apttus_Approval__Approval_Status__c	Apttus Approvals Management	Picklist
Edit | Del	Line Item Coll	LineItemCollId__c		Lookup(Related Opportunity Line Item Coll)

3. Display related lists **Related Opportunity Line Item Collection** and **Related Opportunity Line Items** on the opportunity layout.

Related Opportunity Line Item Collection	New Related Opportunity Line Item Coll
No records to display	

Related Opportunity Line Items	New Related Opportunity Line Item
No records to display	

4. Ensure that the **Primary** checkbox is selected on the quote and the opportunity has a price book associated with it.
5. Ensure that **Synchronize with Opportunity** formula button in displayed on the proposal layout.
6. Finalize quote and click **Synchronize with Opportunity**.
   Observations are same as step 5 in Auto-Synchronization.

ⓘ The options of the service bundle are synchronized to the opportunity only if the **Sync Option Products** setting is enabled in the Proposal System Properties custom setting.

Configuring the Execution of the RelatedLineItem Trigger

The out-of-the-box RelatedLineItem trigger, in it's update block, allocates the net price of the service line item to all associated related line item records based on the **Weightage Type** and **Weightage Amount** fields. However, some customers may have their own allocation model and want to customize it based on their requirements. For such customers, the out-of-the-box RelatedLineItem trigger is redundant and can impact performance. CPQ allows you to control the execution of the RelatedLineItem trigger. You can use the global flag isTriggerInitiatedUpdate to conditionally skip the trigger logic, which allocates the net price of the service line item, and to avoid the unnecessary code execution and potentially improve performance in such cases.

The **isTriggerInitiatedUpdate** flag controls the "before and after update" block only. However, CPQ executes the "before and after insert" block regardless of the value of this flag. This flag remains set for the whole transaction until you unset the flag or until the transaction ends. Currently, CPQ respects this flag only in the RelatedLineItem trigger.

To configure the global flag

Purpose	Value of the Flag	Description
To set the global flag	Apttus_Config2.RuntimeContext.getParameters().put('isTriggerInitiatedUpdate','true');	When the value of the flag is set to *True*, CPQ skips the execution of the RelatedLineItem trigger. CPQ does not use the out-of-the-box logic to allocate the net price of the service line item to all related line items on updating the related line items from the custom code.

Purpose	Value of the Flag	Description
To unset the global flag	Apttus_Config2.RuntimeContext.getParameters().put('isTriggerInitiatedUpdate', 'false');	When the value of the flag is set to *False*, CPQ executes of the RelatedLineItem trigger. CPQ uses the out-of-the-box logic to allocate the net price of the service line item to all related line items based on the Weightage Type and Weightage Amount on updating the related line items from the custom code.

The following is an example of custom code for reference:

```
List<Apttus_Config2__RelatedLineItem__c> RLIs = [SELECT Id,
Apttus_Config2__WeightageType__c, Apttus_Config2__WeightageAmount__c FROM
Apttus_Config2__RelatedLineItem__c WHERE Apttus_Config2__LineItemId__c =
'a132i0000012WS9'];
RLIs.get(0).Apttus_Config2__WeightageType__c = 'Percentage';
RLIs.get(1).Apttus_Config2__WeightageType__c = 'Percentage';
RLIs.get(2).Apttus_Config2__WeightageType__c = 'Percentage';
RLIs.get(0).Apttus_Config2__WeightageAmount__c = 30;
RLIs.get(1).Apttus_Config2__WeightageAmount__c = 30;
RLIs.get(2).Apttus_Config2__WeightageAmount__c = 40;

Apttus_Config2.RuntimeContext.getParameters().put('isTriggerInitiatedUpdate','true');
Database.update(RLIs);
```

Here the **isTriggerInitiatedUpdate** flag is set to *True*.

Configuring the Cart

You can configure the following feature available on the cart.

- Configuring Flow Settings
- Creating Custom Buttons for Different Flows
- Sorting Main Categories on the Catalog Page
- Configuring the Location-Based Cart
- Configuring Column Settings on the Cart and Installed Products Page
- Configuring the Number of Products Displayed on the Catalog Page
- Configuring Custom Attribute Pages
- Hiding the Add to Cart and Configure Buttons for Option Products
- Displaying Leaf Products on the Catalog Page
- Cloning Sub-Bundles, Options, and Attributes on the Options Page
- Enabling Multiple Adjustments at the Line-Item Level
- Configuring Option Net Adjustment Rollup to Bundle
- Enabling Price Breakup for Products
- Displaying Attributes and Options Inline on the Configuration page or a Visualforce Page
- Sequencing the Favorite Category on the Catalog Page
- Configuring the Number of Option Products Displayed on the Cart
- Configuring Save as Favorite
- Configuring the Smart Cart
- Configuring Product Sort
- Configuring Cart Views
- Configuring Cart Activity History
- Configuring CSS Override
- Configuring Mass Option Selection on the Configuration Page
- Configuring Express Proposal

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 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.

Feature	Description
Price Breakdown for cumulative tiers and related pricing	Price breakdown information icon is not displayed for the base price on the Cart page.
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

Flow settings are required so that appropriate pages are loaded for various sections (such as Catalog, Product Configuration, Cart, and Asset). 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. ⓘ Do not use special characters in the name.
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. If you want to view the cart with the new Grid UI, name the cart page as *Apttus_Config2__Cart#cartgrid.*
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 Custom Buttons for Different Flows

To use a different flow for CPQ based on the flow settings, you must create a custom button for the Quote/Proposal object. The custom buttons is named **Configure Product** with a custom label you define in the formula.

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**, you must paste one the following URL under the **Simple Formula** tab.

URL	Description

Regular	This is the regular URL available for **Configure Products** button.
	
	``
	IF (LEN(Apttus_QPConfig__PriceListId__c) > 0 ,
	HYPERLINK("/apex/
	Apttus_QPConfig__ProposalConfiguration?id="&Id&"&flow=
	ngFlow" , IMAGE("/resource/
	Apttus_QPConfig__Button_Configure", "Configure
	Products"),"_self"), NULL)
	``

URL	Description
Optimized	You can use the optimized URL to launch Catalog & Cart quickly compared to the above-mentioned URL. The reduction in time to launch Catalog and Cart compared using the optimized URL may vary and in some Orgs, the benefits may be minimal. The optimized URL primarily requires the following parameters: • Business Object ID • Flow All other parameters are supported. You can update the existing Configure Products formula with the optimized launch URL. If the org has existing customization, you must test and verify the optimized URL after replacing the existing URLs. The following mentioned URL is an example of an optimized URL for the **Configure Products** button. ``` IF (LEN(Apttus_QPConfig__PriceListId__c) > 0 , HYPERLINK("/apex/Apttus_Config2__Cart? businessObjectId="&Id&"&flow=ngFlow", IMAGE("/ resource/Apttus_QPConfig__Button_Configure", "Configure 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. Click **Next**
7. Select appropriate Field Level Security and click **Next.**
8. Choose the page layouts on which you want to display this field.
9. Click **Save**.

You can use to following parameters in the formula to achieve different tasks.

Parameter	Description
productOrderByClause	Use this parameter to sequence the products within a Category. Refer to Sequencing Products within a Category for more details.

Parameter	Description
asyncFinalize	Set the value of this parameter to *True* to finalize any cart asynchronously. When the Sales rep clicks **Finalize**, the cart closes and the Sales rep is redirected to the Quote Details page. The finalization process is completed in the background. The Sales Rep must refresh to see the changed product configuration status. If the parameter value is *False*, then the cart is finalized without closing.
isCartTotalingDisabled	Set the value of this parameter to *True* to selectively skip totaling of pricing of products on the Cart, at a transaction level. When you add products on the cart, the totals section is not updated. ⓘ When you see the value to *True*, CPQ does not execute Pricing Callback in ADJUSTMENT mode.

Parameter	Description
useAdvancedApproval	Set the value of this parameter to *False* to selectively skip the approval of cart, at a transaction level. CPQ invokes the approval process only if this flag is set to *True*. If any workflow processes are setup to trigger approvals based on criteria, they will occur when the criteria are matched. The Approval Stage inside the Quote/Proposal record is set to Approval Required. When the Approval Stage changes, the **Submit for Approval** button becomes available and you can submit the Quote for approval to management if any approval processes have been configured. If this parameter is set to *False* or if it is absent, CPQ skips the approval process even when the criteria are matched. ⓘ To support quote approvals, you must use the **isCartApprovalDisabled** parameter.

useAdvancedApproval	isCartApprovalDisabled	Approval Type	Approval Check
True	True	Quote Approval	No
True	False or absent	Cart Approval	Yes
False or absent	True	None	No
False or absent	False	None	No

When the **useAdvancedApproval** parameter is enabled, it enables cart approvals automatically unless cart approval is explicitly disabled. When cart approval is disabled, CPQ reverts to quote approvals.

Parameter	Description
useDealOptimizer	Set the value of this parameter to *False* to selectively skip deal guidance, at a transaction level. CPQ shows the deal guidance on the Cart page only if this flag is set to *True*. If you have created rules that define the various criteria of a good deal, the deal rating is available to the Sales Representatives at the time of negotiation. If this parameter is set to *False* or if it is absent, CPQ skips the deal guidance and does not show on the Cart page.
deferPricingUntilCart	Set the value of this parameter to *True* to set Defer Pricing on, at a transaction level. When Defer Pricing is on, CPQ performs pricing of products only after you click **Go to Pricing**. When Defer Pricing is off, CPQ performs pricing as and when you add or delete products.
useAdvancedCurrency	Set the value of this parameter to *True* to enable dated currency conversion on the Cart, at a transaction level.
styleSheetURL	Define the URL to the custom style sheet as the value of this parameter to override the default style sheet in CPQ.

Accessing the Cart in Read-Only Mode

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"&"&mode=readOnly",IMAG
E("/resource/Apttus_QPConfig__Button_Configure", "Configure Products"),"_self"),
NULL)
```

By applying this formula to your Configure button, the checkboxes of products and the action buttons are disabled on the Cart page when the sales rep accesses the cart in read-only mode. This prevents your sales rep from applying any mass updates to the products in

the Cart. In read-only mode, the Sales rep can still create views, search the products, and filter the line items on the cart.

Configuring Custom Button to Access Configuration Page Directly

You can define a formula action to configure products and service bundles directly from any page in the org without having to navigate through the Catalog or Cart page. The custom button allows the Sales rep to configure products or service bundles from any other page in the org by opening the Configuration or Service Configuration pages directly. The Sales rep can configure attributes, options, sub-bundles, and related line items on the configuration pages. For example, you can add the custom button on the Quote Detail page or the Line Item Detail page.

> ⓘ For Service CPQ flow, service products and related line items must already be configured before using this feature. You cannot add a new service bundle to the cart using this feature.

You can define the custom button to either add a new product to the cart or edit an existing product or service in the product configuration which is saved or finalized. For service bundles, you can only configure existing service bundles. The formula for the custom button is different based on the functionality you want to define.

You must modify URL structures to either add a new product to the cart or edit an existing product in the product configuration.

You must use the following structures of URL to navigate directly to the Configuration page. Select the URL structure based on how you are configuring the custom setting. You can configure Custom Settings differently for different flows or you can configure common custom settings for all flows. For more information on flow-based settings, see Configuring Custom Settings for Different Flows.

> ⓘ Conga recommends using the URL structure for flow-based custom settings for all buttons for direct access, regardless of the type of custom setting configuration.

Task	Custom Setting Configuration Type	URL
Add a new product	Common	*<instanceURL>/apex/ Apttus_Config2__Cart#!/flows/ <flow name>/businessObjects/ <businessObjectsId>/products/ <productId>/configure*
	Flow-based	<instanceURL>/apex/Cart? flow=<flow name>#!/ businessObjects/ <businessObjectsId>/products/ <productId>/configure
Edit an existing product or service bundle in the quote	Common	*<instanceURL>/apex/ Apttus_Config2__Cart#!/flows/ <flow name>/businessObjects/ <businessObjectsId>/lines/ <primaryLineNumbers>/configure*
	Flow-based	<instanceURL>/apex/Cart? flow=<flow name>#!/ businessObjects/ <businessObjectsId>/lines/ <primaryLineNumbers>/

To modify the URL you must gather the following values.

Value	Description
Instance URL	You can find the Instance URL from the URL of any page in your org. For example, **https://cpq--test.my.salesforce.com**/apex/Cart#!/flows/CartGrid/ businessObjects/a0Y7A000001kWce/products/01t3C000002Rao3/configure
Flow name	The name of the flow using which you want to configure your product. For example, https://cpq--test.my.salesforce.com/apex/Cart#!/flows/**CartGrid**/businessObjects/ a0Y7A000001kWce/products/01t3C000002Rao3/configure ⓘ Smart Cart flows and Partner Portal are not supported.

Value	Description
Business Object ID	The ID of your quote/proposal, order, or agreement which you can find in the URL of the Quote Detail page. For example, *https://cpq--test.my.salesforce.com/apex/Cart#!/flows/CartGrid/businessObjects/**a0Y7A000001kWce**/products/01t3C000002Rao3/configure*
Product ID	The ID of the product which can find in the URL of the Product Detail page. You need the ID of the product to add a new product to the cart. For example, *https://cpq--test.my.salesforce.com/apex/Cart#!/flows/CartGrid/businessObjects/a0Y7A000001kWce/products/**01t3C000002Rao3**/configure* ⓘ In the following scenarios, the Sales rep is directed to the Catalog page instead of the Configuration page and CPQ displays anerror message: • The Product ID provided is incorrect • The Product ID belongs to a different price list that is not associated with the Business Object • The Business Object is not associated with any price list.
Primary Line Numer	The Primary Line Number of a product is generated on the line item after the product is added to the cart. You can find the Primary Line Number of the product on the Line Item Detail page. You need the Primary Line Number of the product to edit an existing product in the configuration. For example, *https://cpq--test.my.salesforce.com/apex/Cart#!/flows/CartGrid/businessObjects/a0Y7A000001kWce/lines/**1**/configure*

The following use cases describe how to define both the functionalities. The custom button on the Quote Detail page allows the Sales rep to add a new product or service each time the button is clicked. However, the custom button on the Line Item Detail page allows the Sales rep to update and modify the existing product or service.

Custom Button on Quote Detail Page to Add a New Product

On the Quote Detail page, the Sales rep must provide the URL of the product in a custom text field to open the Configuration page for that product. When the Sales rep clicks the custom button on the Quote Detail page, the product corresponding to the URL provided in the custom text field is added to the cart and the Configuration page of that product is displayed. If the Sales rep clicks the button multiple times, the same product is added to

the cart again. You must create a custom text field on the **Quote/Proposal** object where the Sales rep can enter the product URL.

Pre-requisite

Create a custom text field in the **Quote/Proposal** object.

To create a custom button on Quote Detail page

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**:

Custom Setting Type	URL
Common	IF (LEN(Apttus_QPConfig__PriceListId__c) > 0 , HYPERLINK("/apex/Apttus_Config2__Cart#!/flows/ CartGrid/businessObjects/" &Id & "/ products/"&ProductId__c&"/configure" , IMAGE("/resource/ Apttus_Config2__Button_Configure", "Configure Products"),"_self"), NULL)
Flow-based	IF (LEN(Apttus_QPConfig__PriceListId__c) > 0 , HYPERLINK("/apex/Apttus_Config2__Cart? flow=CartGrid#!/businessObjects/" &Id & "/ products/"&ProductId__c&"/configure" , IMAGE("/resource/ Apttus_Config2__Button_Configure", "Configure 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 *CartGrid*.
> Ensure that you replace *ProductId__c* with the name of the custom text field that you created in the **Quote/Proposal** object.

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**.

Custom Button on Line Item Detail page to Edit an Existing Product or Service Bundle

You can add the custom button on the Line Item Detail page to allow the Sales rep to update or modify the existing product configuration of a product or a service bunble. The formula for the custom buttons refers to the Primary Line Number that is generated by CPQ upon adding a product to the cart, to open the Configuration page for the specific product or service bundle.

To create a custom button on the Line Item Detail page

1. Go to **Setup > Create > Objects** and search for **Line Item** 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**:

Custom Setting Type	URL
Common	IF (LEN(Apttus_Config2__PriceListId__c) > 0 && TEXT(Apttus_Config2__LineType__c) = "Product/Service" , HYPERLINK("/apex/Apttus_Config2__Cart#!/flows/CartGrid/businessObjects/" & Apttus_Config2__ConfigurationId__r.Apttus_QPConfig__Proposald__c & "/lines/" & TEXT(Apttus_Config2__PrimaryLineNumber__c)&"/configure", IMAGE("/resource/Apttus_Config2__Button_Configure", "Config"),"_self"), NULL)

Custom Setting Type	URL
Flow-based	IF (LEN(Apttus_Config2__PriceListId__c) > 0 && TEXT(Apttus_Config2__LineType__c) = "Product/Service" , HYPERLINK("/apex/Apttus_Config2__Cart?flow=CartGrid#!/businessObjects/" & Apttus_Config2__ConfigurationId__r.Apttus_QPConfig__Proposal__c & "/lines/" & TEXT(Apttus_Config2__PrimaryLineNumber__c)&"/configure", IMAGE("/resource/Apttus_Config2__Button_Configure", "Config"),"_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 *CartGrid*.

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**.

Configuring a Custom Button to Launch the Installed Products Page Directly for the Add/Remove Flow

You can launch the Installed Products page directly for the Add/Remove flow (Service CPQ), where you can modify the association of assets with services. CPQ enables you to define parameters on the URL to launch the Installed Products page directly from the custom cart page.

The following is the syntax of the URL:

```
<instanceURL>/apex/Cart?
businessObjectId=<businessObjectId>&flow=<flowName>&servicePLN=<servicePrimaryLineNumber>#!/assetsgrid
```

Here you can pass the value servicePLN as follows:

- servicePLN=1, which means the primary line number 1, representing the first service product
- servicePLN=2, which means the primary line number 2, representing the second service product

To create a custom button

Perform the following steps to create a custom button:

1. Go to **Setup** > **Create** > **Objects** and search for the **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," enter the following details and click Next:
   a. In the **Field Label** field, enter Configure Products <Flow>.
   b. In the **Field Name** field, enter Configure_Products_<Flow>.
   c. For Formula Return Type, choose **Text**.
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/Cart?
businessObjectId=<businessObjectId>&flow=<flowName>&servicePLN=<servicePrimaryL
ineNumber>#!/assetsgrid", IMAGE("/resource/Apttus_QPConfig__Button_Configure",
"Configure Products"),"_self"), NULL)
```

6. In **Step 4. Establish field-level security**, select the appropriate Field Level Security and click **Next**.
7. In **Step 5. Add to page layouts**, choose the page layouts on which you want to display this field.
8. Click **Save**.

Example URL:

```
IF ( LEN( Apttus_QPConfig__PriceListId__c ) > 0 , HYPERLINK("/apex/
Apttus_Config2__Cart?businessObjectId="&Id&"&flow=LAngflow&servicePLN=1#!/assetsgrid",
 IMAGE("/resource/Apttus_QPConfig__Button_Configure", "Configure Products"),"_self"),
NULL)
```

Here `servicePLN=1` means the primary line number 1, representing the first service product.

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.

Configuring the 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.

Configuring Column Settings on the Cart and Installed Products 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, based on your requirement.
5. Click **Save**.

Configuring the 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.

Configuring Custom Attribute Pages

You can now use your predefined 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 Conga 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.

Hiding the 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 **Custom Settings**> **Config Select Products Settings** and select **Disable Add To Cart for Option Products**. For more details, refer to Configuring Custom Settings.

Displaying 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 Sub-Bundles, Options, and 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.

Enabling Multiple Adjustments at the Line-Item Level

There are two types of adjustments that are applicable on a line item on the cart:

- Auto-applied adjustments: CPQ applies these automatic adjustments such as promotions, price matrices, and price rulesets.
- Manual adjustments: Sales reps can manually apply these adjustments on the multiple adjustments pop-up on the cart. For more information, see Applying Multiple Adjustments on Line Items in the Cart.

You can configure how many manual adjustments that sales reps can apply on a cart line item using the **Max Adjustment Lines** setting available in **Config System Properties**. For example, if Max Adjustment Lines = 3, the sales representative can apply a maximum of three manual adjustments on the multiple adjustments pop-up. After that, CPQ disables the **+ Add another adjustment** button and displays a message.

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

Max Adjustment Lines Value	Behavior
Blank or 0	If the value of this setting is 0, CPQ displays only auto-applied adjustments on the multiple adjustments pop-up. Sales reps cannot add any manual adjustments. They can hover the mouse on the line item, click the more icon ( ), and then select $ **Multiple Adjustments** view auto-applied adjustments (in read-only mode). CPQ does not display the **+ Add another adjustment** button, but displays a message.

Max Adjustment Lines Value	Behavior
Greater than or equal to 1	CPQ displays auto-applied adjustments (in read-only mode) on the multiple adjustments pop-up and allows sales representatives to create as many adjustments as the value defined in the **Max Adjustment Lines** setting. After reaching the limit, CPQ disables the **+ Add another adjustment** button and displays a message.

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.

You can enable users to apply an adjustment on the price of a bundle without applying that same adjustment on the options within the bundle that share the same charge type name.

You can enable users to apply an adjustment type on the bundle or total using % Discount or % Markup so that the Net Adjustment % on the Bundle or Total is set to the user requested value. CPQ allocates the adjustment to the group members (options in the case of a bundle and standalone products and options of bundles if the group is a total). The following requirements must be fulfilled to enable this feature:

- Group Adjustment Spread is set to True.
- Max Adjustment Line must be blank or 1.
- Bundle price must be zero.

The sales rep can select an adjustment from the following Adjustment Types and enter a value in **Adjustment Amount** column on the Cart page:

> ⓘ Some of the values might not be available by default. You must add the required value to the **Adjustment Type** field in the Line Item object.
> Perform the following steps to add a value to the **Adjustment Type** field in the Line Item object:
> 1. Go to **Setup> Create> Object> Line Item> Adjustment Type**.
> 2. Under the section **Values**, click **New**.
> 3. Enter *<Value_Name>* and click **Save**.

Option	Description
% Discount	Indicates that a percentage discount is applied on the line item based on Adjustment Amount.
Discount Amount	Indicates that a flat discount is applied on the line item based on Adjustment Amount.
% Markup	Indicates that a percentage markup is applied on the line item based on Adjustment Amount.
Markup Amount	Indicates that a flat markup is applied on the line item based on Adjustment Amount.
Price Override	Indicates that the Adjustment Amount entered overrides the net price of the product.
Base Price Override	Indicates that the Adjustment Amount overrides the base price of the product and all the other calculations on the product would happen based on the new base price. **Allow Manual Adjustment** must be selected on the product line item, in order to make **Base Price Override** editable.
% Discount Off List	This is similar to % Discount, except that it calculates the discount amount always on the List Price of a Bundle and applies to the Extended Price.
% Uplift	This gets applied every time to products that are added from the Asset.
Price Factor	Indicates that the Adjustment Amount is multiplied with the **Extended Price** to calculate the new **Net Price**.
Base Price Discount	Indicates that the adjustment amount is reduced from the Base Price.
Base Price Markup	Indicates that the adjustment amount is added to the Base Price.
% Discount (Bundle Only)	Indicates that a percentage discount is applied on the bundle without applying that same adjustment on the options within the bundle.

Option	Description
Discount Amount (Bundle Only)	Indicates that a flat discount is applied on the bundle without applying that same adjustment on the options within the bundle.
% Uplift (Bundle Only)	This is applied to the bundle every time a product is added from the Asset.
% Markup (Bundle Only)	Indicates that a percentage markup is applied on the bundle without applying that same adjustment on the options within the bundle.
Markup Amount (Bundle Only)	Indicates that a flat markup is applied on the bundle without applying that same adjustment on the options within the bundle.
Price Override (Bundle Only)	Indicates that the Adjustment Amount entered overrides the net price of the bundle without applying that same adjustment on the options within the bundle.
Price Factor (Bundle Only)	Indicates that the Adjustment Amount is multiplied with the **Extended Price** to calculate the new **Net Price** of the bundle.
% Discount Effective	Indicates that a percentage discount is applied on the bundle or total so that the Net Adjustment % on the Bundle or Total is set to the user requested value. This adjustment type is applicable when the following requirements are fulfilled: • Group Adjustment Spread is set to True. • Max Adjustment Line must be blank or 1. • Bundle price must be zero.

Option	Description
% Markup Effective	Indicates that a percentage markup is applied on the bundle or total so that the Net Adjustment % on the Bundle or Total is set to the user requested value. This adjustment type is applicable when the following requirements are fulfilled: • Group Adjustment Spread is set to True. • Max Adjustment Line must be blank or 1. • Bundle price must be zero.
Price Factor Effective	Indicates that the Adjustment Amount is multiplied with the **Extended Price** to calculate the new **Net Price** of the bundle or total so that the Net Adjustment % on the Bundle or Total is set to the user requested value. This adjustment type is applicable when the following requirements are fulfilled: • Group Adjustment Spread is set to True. • Max Adjustment Line must be blank or 1. • Bundle price must be zero.
% Discount Off Base	Indicates that a percentage discount is applied on the base price of standalone and bundle products. CPQ applies this adjustment on the unit price after applying adjustments from price matrices, price rules, and promotions. If the administrator has enabled currency rounding, CPQ rounds the adjustment applied and applies it on the unit price before calculating the new net price.

Configuring Option Net Adjustment Rollup to Bundle

You must set the Admin setting *APTS_RollUpOptionNetAdjustment* to *True* so that when a Sales Representative applies adjustments on options, CPQ displays the adjustments on options at the bundle and summary totals level.

1. Go to **All Tabs** > **Admin**
2. Click **New**, to create a new record. The New Admin page is displayed.
3. Fill the following details

Field	Value
Name	*APTS_RollUpOptionNetAdjustment*

Field	Value
Value	*true* or *false*
Code	Leave the field blank

4. Click **Save**.

Enabling Price Breakup for Products

The Price Breakup provides additional information if the line item has related pricing or a price matrix with a dimension having cumulative range. A small icon () next to the base price is displayed if the line item uses either of the two pricing features. You can click the icon to view the additional information.

You can add custom fields on the **Price Breakup** object and add the same in **Display Column Settings** for the Price Breakup table so that when sales representatives open the Price Breakdown pop-up on the Cart, they can see columns for the added custom fields with their corresponding values. If a product line is priced using related pricing, when the sale representative clicks the price breakup icon () next to the base price on the cart, the custom field values are displayed on the Price Breakdown pop-up.

For example, a product record has various fields such as Tier 1, Tier 2, and Tier 3. These are the parameters that help the sales representative identify the product's family and its line of business. You can display these fields on the Price Breakdown pop-up so that the sales representative can understand which parameters are contributing to the related price calculation.

Prerequisite

You must have already created a Price Dimension.

To enable Price Breakup for products

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 Inline on the Configuration page or a Visualforce 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.

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

Sequencing the Favorite Category on the Catalog Page

The new UI supports the custom setting **Show Favorite as First Category** to display your favorite category at the top.

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 **Show Favorite as First Category** check box.

Now, when you have multiple product offerings, you can quickly find your favorite category on the cart. For more information, see *About Favorite Configurations on the Cart*.

Configuring the Number of Option Products Displayed on the Cart

For the New UI, customize the following custom setting to display the number of option products on the cart 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 **Option Items Per Page** field, enter the number of option products that you want to display on the cart per page. Valid values 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 option 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.

Configuring Save as Favorite

The Save as Favorite feature allows the sales rep to save a cart configuration for reuse. The saved configuration can be used multiple times in the same quote or across different quotes. A saved configuration includes all the selected products on the cart with the

options and attributes defined within the product. The configuration can be public for all the sales reps or private to the sales rep who saved the configuration.

As an administrator, you can use various functionalities to configure the Save as Favorite feature to control the usage of the feature by the sales reps. Following functionalities are available to you to configure Save as Favorite feature.

To disable the Save as Favorite feature

By default, the feature is enabled. You can disable the feature globally or selectively apply permission to individual user or profile. Disabling the feature only disables the feature icon on the Cart page. To disable Favorite feature globally follow the steps below.

1. Go to **Setup** > **App Setup** > **Develop** > **Events** > **Custom Settings**.
2. Find **Config Select Product Setting** and click **Manage** next to the setting.
3. Click **Edit** next to **Primary Settings**.
4. Select **Disable Favorite**.
5. Click **Save**.

As a result of following the steps above the Save as Favorite icon(⭐) on the Cart page is disabled.

You can also control which user or profile can use Favorite Feature by defining access permissions for a user or profile.

To define permissions to use Save as Favorite feature

The administrator must define proper access permissions for user profiles to use Save as Favorite Feature. Otherwise, the user cannot see the Save as Favorite feature on the Cart page. Follow the steps below to define access permissions for the user.

1. Go to **Setup> Administrator Setup> Manage Users> Profiles.**
2. Find the profile for which you want to set permissions and click **Edit**.
3. Find **Favorite Configuration** under the section **Custom Object Permissions**.
4. Select one or more permission from the list below to define access for the profiles.

Name	Impact on the Cart Page
Read	A disabled **Save As Favorite** button or a star icon (⭐) is displayed on the Cart page.

Name	Impact on the Cart Page
Create	The **Save As Favorite** button is enabled on the Cart page.
Edit	The Sales rep can view and import Favorites configuration saved by others on the Catalog page when you define Edit permission
Delete	**Delete** option is enabled for Favorites on the Catalog page.

If you do not select any permission, the Favorite feature is disabled on the Cart page.

5. To enable **Publish** button on the Cart page, find **Published Favorite** under **Custom Object Permissions** and select **Create**.
6. Click **Save**.

Based on the permissions you define here, the sales rep can use different functionality available to them.

To hide scope field in Save as Favorite pop-up on the Cart

Follow the steps below to hide the Scope field.

1. Go to **Setup**. From the **Manage Users** drop-down, select **Profiles**.
2. Select a custom Profile.

> ⓘ You can only edit these permissions for a custom profile.

3. Under the **Field-level Security** section find **Favorite Configuration** and click **View**.
4. Click **Edit**
5. To hide the Scope field clear the checkboxes **Read Access** and **Edit Access**.
6. To make the Scope field visible to the profile select the checkboxes **Read Access** and **Edit Access**.
7. Click **Save**.

The above action hides or displays the **Scope** field for a specific profile. When the **Scope** field is visible, the users can save a configuration record as **Public** or **Private** by selecting from the picklist. Otherwise, the configuration record is saved as **Private**.

Categorization of Favorite Configuration

CPQ provides sales rep the ability to categorize the saved configuration using pre-defined filters. These fields are displayed on Save as Favorite pop-up on the Cart page while saving the configuration. The fields in the filter are of picklist and multi-select picklist type that you must create in the Favorite Configuration object to use as filters. Any existing picklist or multi-select picklist fields can also be used, except for **Scope**. If you define default values or make the fields required, they are honored on the pop-up as well. Only the first five fields that you define are displayed on the pop-up.

To define filters to save a configuration

Follow the steps below after creating picklist or multi-select picklist type fields in Favorite Configuration object.

1. Go to **Setup** > **App Setup** > **Develop** > **Events** > **Custom Settings**.
2. Click **Manage** next to **Config System Properties** setting.
3. Click **Edit** next to **System Properties**.
4. Find **FavoriteFilters** and enter the API names of picklists you created, separated by comma or a new line.
5. Click **Save**.

Configuring the Smart Cart

The Smart Cart feature allows the sales rep to manage the cart with a large number of line items. In Smart Cart flow, line items in the cart are divided into groups for pricing based on the threshold and split criteria you define. The threshold defines the number of line items in a group. The split criteria list the fields based on which the line items are grouped. You can define different threshold and criteria for different flow. You must configure the Smart Cart feature for the sales rep to use.

When you configure the product in the Smart Cart flow, you must take the following concepts into consideration.

- The option line items along with bundle line item are included in the threshold count. For example, if the threshold is 10, then in a bundle you can only select 9 options as that occupies an entire group. So, when the groups are divided on pricing, if your bundle and option line items exceed the threshold, pricing is not calculated. However, if the bundle doesn't fit in a half-full group a new group is created for that bundle as long as the line items are within the threshold limit.

- A bundle is not divided across multiple carts. If the options are auto-included or added by default and the split cart threshold is exceeded, the split function is not executed.
- Line items with multiple charge lines are divided across multiple carts if required.

Follow the steps below to configure the feature.

To add QTC profile field on Quote Detail page layout

You must add the field **QTC Profile** to the Quote Detail page layout. This field is used by the sales rep to enable the Smart Cart flow for their quote.

To define the threshold for the Smart Cart

The threshold defines the size of the group that the line items are divided into for pricing. Only the primary line items are considered in the threshold to split the cart. The primary line items consist of the line items for standalone products, bundle products and option products.

You must define the threshold considering the pricing complexity and the number of line items your pricing engine can process. If the threshold exceeds the limit of the number of line items processed in the price engine, Smart Cart does not split the cart properly and the price is not calculated. However, if you define a low threshold when you have a relatively large number of line items, Smart Cart creates a large number of groups, and pricing is not calculated properly.

Also, the bundles are not divided across multiple carts. If the number of bundles and option line items is greater than the threshold, the split function is not executed. You can increase the threshold keeping the pricing engine in consideration or restructure the bundle.

Split Cart Threshold is the primary field that Smart Cart uses to split the line items in the cart. Even if you don't define **Split Cart Criteria Fields**, the line items are grouped based on just the threshold.

Perform the following steps to set the threshold.

1. Go to **Setup > Develop > Custom Settings > Config System Properties > Manage**.
2. Find **System Properties** and click **Edit**.
3. Enter a number in the **Split Cart Threshold** field.
4. Click **Save**.

To define the split criteria for the Smart Cart

The split criteria define the criteria based on which the line items are grouped. Defining **Split Cart Criteria Fields** is optional as Smart Cart can split the cart based on just the threshold. If you define the **Split Cart Criteria Fields**, then Smart Cart considers the criteria first then the threshold. Hence, when the line items are divided, the groups are created based on the criteria first and then the number of line items in a group is controlled by Smart Cart.

Perform the following steps to set the criteria.

1. Go to **Setup > Develop > Custom Settings > Config System Properties > Manage**.
2. Find **System Properties** and click **Edit**.
3. in the **Split Cart Criteria Fields** field, enter the API names of the fields from the Line Item object, separated by commas. Only the fields with the following data types are supported as criteria:
   - Text
   - Picklist
   - Number
   - Formula fields evaluating to Text output
   - Relationship fields.

   > ⓘ Other data types such as Boolean, Text Area, and Rollups are not supported.

4. Click **Save**.

> ✅ If the fields you entered in **Split Cart Criteria Fields** are uncommon fields that are not used in any prominent custom settings or display columns, execute *Refresh Field Metadata* from Maintenance Console to update the data cache.

To define line item processing batch size

Smart Cart uses **Split Cart Threshold** to split the line items into groups to calculate pricing. This threshold is defined based on the pricing complexity and the number of line items your pricing engine can process. However, the threshold can exceed the governer heap limit if the size is large for a single pricing call. You can use the admin setting **APTS_CartLineItemsUpdateChunkSize** to define the number of line items in a batch to be processed in a single pricing call. You can use this admin setting to avoid hitting heap size errors.

1. Go to **All Tabs** > **Admin**
2. Click **New**, to create a new record. The New Admin page is displayed.
3. Fill in the following details

Field	Value
Name	*APTS_CartLineItemsUpdateChunkSize*
Value	Enter the number of line items to be processed in a batch for pricing.
Code	Leave the field blank

4. Click **Save**.

To enable actions on the cart page

In the Smart Cart flow, the action buttons are different than the actions available in a regular flow. You must enable the action buttons to display them in the action menu of the cart for Smart Cart flow. Follow the steps below.

1. Click the All Tabs icon ( ).
2. Go to **Config Settings**>**Display Action Settings**.
3. Select *Cart Page* in **Display Type** and the flow that you want to use in Smart Cart flow in **Flow**.
4. Find and select the checkbox **is Enabled** for the following actions.
   - Submit for Pricing (Async)
   - Pre-Price
   - Post-Price
5. Click **Save**.

Use Case for Configuring the Smart Cart

Description: This use case describes how the line items are grouped in Smart Cart.

Suppose you are a CPQ administrator of a company selling health care equipment to hospitals in bulk. You sell to hospitals with branches in different locations. Selling to such hospitals in bulk results in quotes with a large number of line items and configuration based on different criteria such as Location. The quotes also include the pricing details including the adjustments applied on the line items. Your Sales Rep regularly deals with

such a large number of line items in a single quote. A large number of line items are difficult to handle and navigate through.

Using the Smart Cart feature, the pricing of the line items calculated in chunks based on a threshold and criteria. The cart is divided into Config and Pricing configurations based on the threshold and criteria you define.

For example, if you want to divide the line items into groups of 500 line items and create the groups based on the location of different branches of the hospital. You must define the **Split Cart Threshold** as 500 and **Split Cart Criteria Fields** as the API name of the location field.

Result

If the total number of line items the Sales Rep adds to the cart is 2000 and clicks **Submit for Pricing(Async)**. The line items are divided into groups on the quote page represented by the configurations. Consider the below screenshot as the sample of the products added to the cart with location.

Location	Product
Loc1	Orbital Sandler
Loc2	Orbital Sandler
Loc1	Orbital Sandler
Loc4	Orbital Sandler
Loc5	Orbital Sandler
Loc6	Orbital Sandler
Loc7	Orbital Sandler
Loc8	Orbital Sandler
Loc9	Orbital Sandler
Loc10	Orbital Sandler
Loc1	Orbital Sandler
Loc2	Orbital Sandler
Loc1	Orbital Sandler
Loc4	Orbital Sandler
Loc5	Orbital Sandler
Loc6	Orbital Sandler
Loc7	Orbital Sandler
Loc8	Orbital Sandler
Loc9	Orbital Sandler
Loc10	Orbital Sandler
Loc1	Orbital Sandler
Loc2	Orbital Sandler
Loc1	Orbital Sandler
Loc4	Orbital Sandler

Based on the values of **Split Cart Threshold** and **Split Cart Criteria Fields**, the Child Carts are created as depicted in the table below, while the number in the parenthesis denotes the number of line items in the Child Cart.

Pricing Configuration	Line Item Division
1	Location 1 (200)
2	Location 2 (300)
3	Location 3 (500)
4	Location 4 (10)
5	Location 5 - 1 (500)
6	Location 5 - 2 (490)

Configuring Product Sort

Product Sorting enables the Sales rep to sort products on the Catalog page. The Catalog page is by default sorted in ascending order of Product Name, the Sales rep can change that in the **Sort** drop-down. The Sales rep can select to sort the product in either ascending or descending order of the field. In **Sort** drop-down on the Catalog page, two options are provided by default:

- *Product Name Ascending*
- *Product Name Descending*

You can add two more fields in the drop-down by defining the fields in **Product Sorting Fields** in Config System Properties. Each field you define in **Product Sorting Fields** has two entries in the **Sort** drop-down on the Catalog page, one for ascending order and other for descending order of the products.

Defining Product Sorting Fields

You can define a maximum of two additional fields in **Product Sorting Fields** to be displayed in the **Sort** drop-down apart from the field **Product Name. Product Name** is added by default. If you add more than two API names in **Product Sorting Fields** then only the first two values are displayed in the **Sort** drop-down.

To Define Product Sorting Fields

1. Go to **Setup** > **Develop** > **Custom Settings**.
2. Click **Manage** next to **Config System Properties**.
3. Click **Edit** next to **System Properties.**
4. In **Product Sorting Fields**, enter the API names of the fields from Product object to display the **Sort** drop-down. You can define a maximum of two fields at a time. You can add fields only from the Product object with the following data types:
   - Text
   - Number
   - Checkbox
   - Date
   - Time
   - Date/Time
   - Picklist
   - Currency
     An example of how the API name fields are added to the **Product Sorting Fields** is shown below:

5. Click **Save**.

Configuring Cart Views

This section provides information on configuring cart views.

- About Cart Views
- Enabling Cart View Creation in CPQ Admin Console
- Running the Migration Script for Cart Views

About Cart Views

CPQ provides admin cart views.

You can set a cart view as default or admin for a group of users in order to provide them a tailor-made experience per business requirements.

- You can set an exisiting or a new cart as default or admin.

- You can create a user group based on profiles, roles, and users to assign the default view.
- You can disassociate the user groups from the default view.
- You can format the admin view layout.

The cart views uses following two objects for storing the information related to views instead of only ConfigSetting object.

- Config Field Set object

  This object is used to store the field list for a particular Cart View that is created by a user.

  Go to **Custom Setting** > **Config Field Set** to see the view that a user has created from the Cart. The cart views created on the cart by the users are *Private* and are not visible to other users.

- User Views object

  This object is used to store the association between the Cart Views and the User Type (User Profiles or User Roles or Users). It also stores information such as Flow, Default setting, Group By field, and Type of profile.

  Go to **All Tabs** > **User Views** to see the view that an admin has created from the New Admin UI (CPQ Admin) > **Application Management** > **Cart Page**.

  On the New Admin UI, an admin can configure a view with a Flow, View Name, Group By field, default setting, Public or Private profile type, and the User Type. Ususally the view created by an admin is *Protected* by default and is visible to other users based on the eligibility. The admin can make the view public to make it visible to all users.

> ⓘ A user can delete a view which is created only by the user (Private view) while an admin can delete a view which is created only by admin (Protected and Public views).
>
> If a default view is deleted, then the next default view in the following order of precedence is displayed:
> - User view = Private
> - Admin views = Profile, Role, User, Public
>
> If a user does not set the default view, then the first view that is associated with the user profile is displayed as default.

Enabling Cart View Creation in CPQ Admin Console

You need to configure the settings for creating a cart view in the New Admin Console of CPQ.

To create a Cart View in New Admin Console

1. Go to **Custom Object Setting Group** and click **Display Type** picklist field.
2. Add the picklist value **Custom**.
3. Create a tab for **Application Features** object.
4. Go to **Application Features** tab and then click the **Cart Page** record.

   > ⓘ If you do not see any records under **Application Features** tab, then you need to click **Load Defaults** on CPQ Admin by navigating to **All Tabs > CPQ Admin > Application Management > Load Defaults**.

5. Edit the layout of the "Cart Page" record and add "Setting Groups" related list to it.
6. Click **New Setting Group** in the "Setting Groups" related list on the "Cart Page" layout. Fill the below details in the new setting group record for "Cart View" and save it.

7. After you create the "Cart View" setting group record, it will be available in the "Setting Groups" related list on the "Cart Page" layout as shown below:

Cart View page gets available on the CPQ Admin console. You can access this page from **All Tabs > CPQ Admin > Application Management > Cart Page > Cart Views**.

> ⓘ To create view for a flow, first you need to configure the flow for Cart Grid by adding "#cartgrid" after the flow name.
> Only those flows which are configured for Cart Grid are available in the "Flow" field on the "Cart View" page.

Running the Migration Script for Cart Views

As an existing customer, if you are upgrading to the current release from a release version older than Summer 2018, you must run a post-installation script manually through the developer console to update your existing Cart Views to the new framework.

> ⓘ For customers setting up org on release version later than Summer 2018, the post-install script runs automatically after you install the CPQ package. In such a case, you do not need to execute the migration script manually.

Migration Script

Apex Code: Create a new class in the org with the code given below.

```
public class APTS_CreateCartViews{
```

```java
/**
 * Checks if the given string value is null or empty.
 * @param strValue the string to check
 * @return <code>true</code> if the string value is null or empty, <code>false</code>
otherwise
 */
public static Boolean nullOrEmpty(String strValue) {
    // check if null or zero length string
    return (strValue == null || strValue.trim().length() == 0);
}


/**
 * Checks if the given list of objects is null or empty.
 * @param objList the list of objects to check
 * @return <code>true</code> if the list is null or empty, <code>false</code>
otherwise
 */
public static Boolean nullOrEmpty(List<Object> objList) {
    // check if null or empty
    return (objList == null || objList.isEmpty());
}


/**
 * returns comma separated value
 */
public static String arrayToCSV(String[] arr){
    String str = '';
    Boolean isFirst = true;
    for(String ele : arr){
        if(isFirst){ isFirst = false; }else{
            str += ', '; //adding space since to display rule action info properly in
the layout
        }
        str += ele;
    }
    return str;
}

public static void saveFieldList(String dataStore, List<String> fieldNames) {

    String names = arrayToCSV(fieldNames);
    Integer len = names.length();
    Integer recordCount = Integer.valueOf(Math.ceil(len * 1.0 / 255));
```

```
    //delete existing records
    Apttus_Config2__ConfigFieldSet__c dataCache =
Apttus_Config2__ConfigFieldSet__c.getInstance(dataStore + '_' + 0 );
    if (dataCache != null) {
        List<Apttus_Config2__ConfigFieldSet__c> queryFieldList = new
 List<Apttus_Config2__ConfigFieldSet__c>();
        Integer index = 1;
        while(dataCache != null){
            queryFieldList.add(dataCache);
            dataCache = Apttus_Config2__ConfigFieldSet__c.getInstance(dataStore + '_'
+ index );
            index += 1;

        }
        delete queryFieldList;

    }

    List<Apttus_Config2__ConfigFieldSet__c> queryFieldList = new
 List<Apttus_Config2__ConfigFieldSet__c>();

    for (Integer i = 0; i < recordCount; i++) {
        Integer startIndex = i * 255;
        Integer endIndex = (i + 1) * 255;
        Integer isLast = 0;
        if (endIndex >= len) {
            endIndex = len;
            isLast = 1;
        }
        queryFieldList.add(new Apttus_Config2__ConfigFieldSet__c(Name = dataStore +
'_' + i, Apttus_Config2__Data__c = names.substring(startIndex, endIndex),
Apttus_Config2__IsLast__c = isLast));

    }

    insert queryFieldList;
}

public class MigrateCartView {
    public String GroupByField;
    public String viewName;
    public List<columnDO> Columns;
    public Boolean isDefault = false;
```

```
}

//Data structure used to hold the deserialized JSON of cartviewsetting.
public class columnDO {
    public String FieldName;
    public Integer Sequence;
    public String Pinned;
}

//this method deserialize the cartview setting related data stored in JSON format.
private static List<MigrateCartView> parse(String json) {
    return (List<MigrateCartView>) System.JSON.deserialize(json,
List<MigrateCartView>.class);
}

/**
 * Create the new Cart Views Setting Group for Application Management feature of New
Admin
 */
public static void createCartViewsSettingGroup() {
    List<Apttus_Config2__ApplicationFeature__c> cartPageAppFeatures = [SELECT
Id,Name,Apttus_Config2__ApplicationName__c

                                                  FROM
Apttus_Config2__ApplicationFeature__c

                                                  WHERE
Apttus_Config2__ApplicationName__c = 'CPQ'

                                                      AND Name = 'Cart Page'
                                                      LIMIT 1];

    List<Apttus_Config2__SettingGroup__c> cartViewsSettingGroups = [SELECT Id,Name
                                                  FROM Apttus_Config2__SettingGroup__c
                                                  WHERE Name = 'Cart Views'
                                                  LIMIT 1];

    // Create the new Cart Views Setting Group only if the Cart Page Application
Feature exists and the Cart View Setting Group DOES NOT exist
    if (cartPageAppFeatures != null
        && cartPageAppFeatures[0] != null
        && nullOrEmpty(cartViewsSettingGroups)
        ) {

        Apttus_Config2__SettingGroup__c newCartViewsSettingGroup = new
 Apttus_Config2__SettingGroup__c();
        newCartViewsSettingGroup.Name = 'Cart Views';
```

```
        newCartViewsSettingGroup.Apttus_Config2__ApplicationFeatureId__c =
cartPageAppFeatures[0].Id;
        newCartViewsSettingGroup.Apttus_Config2__IsCustom__c = false;
        newCartViewsSettingGroup.Apttus_Config2__Sequence__c = 12;
        newCartViewsSettingGroup.Apttus_Config2__DisplayType__c = 'Custom';
        newCartViewsSettingGroup.Apttus_Config2__ConfigFlow__c = 'Default';

        insert newCartViewsSettingGroup;
    }
}


/**
 * Post installation script to migrate cartview setting related data into new object
model.
 */
public static void migrateCartViewSetting() {
    String nsPrefix = 'Apttus_Config2__';
    List<Apttus_Config2__UserView__c> newCartViewSettings = new
 List<Apttus_Config2__UserView__c>();
    Integer isProductIdCount = 0;
    List<String> allColumns;
    List<MigrateCartView> allOldData;
    String pinned;
    String viewName;

    //
    List<Apttus_Config2__ConfigSettings__c> configSOs = [SELECT Name,
                                            Apttus_Config2__Scope__c,
                                            Apttus_Config2__ViewSettings__c,
                                            Apttus_Config2__Flow__c
                                    FROM Apttus_Config2__ConfigSettings__c
                                    WHERE Apttus_Config2__Scope__c = 'User'
                                    LIMIT 100];
    for (Apttus_Config2__ConfigSettings__c configSO : configSOs) {
        isProductIdCount =0;

        // Deserialize the JSON text stored in viewSettings__c into MigrateCartView
data structure.
        allOldData = parse(configSO.Apttus_Config2__ViewSettings__c);

        // One record can hold setting data for more than one view. Loop thru all
views.
        for (MigrateCartView oldView : allOldData) {
```

```
        allColumns = new List<String>();

        // Current design allows max 36 chars for view name. Remaining chars to
be truncated below.
        viewName = oldView.viewName.length() > 36 ? oldView.viewName.left(36) :
oldView.viewName;
        newCartViewSettings.add(new Apttus_Config2__UserView__c (Name = viewName,
                                    Apttus_Config2__Type__c =
'Public',

                                    Apttus_Config2__Flow__c =
configSO.Apttus_Config2__Flow__c,

                                    Apttus_Config2__GroupBy__c =
oldView.GroupByField,

                                    Apttus_Config2__IsDefaultView__c
= oldView.isDefault,

Apttus_Config2__IsDefaultRecord__c = oldView.isDefault));

            // Loop through every column settings
            for(ColumnDO ColumnD : oldView.Columns) {

                //Column pinned value left null to be changed to middle
                if (ColumnD.Pinned == 'left') {
                    pinned = 'Left';
                } else    if (ColumnD.Pinned == 'right') {
                    pinned = 'Right';
                } else {
                    pinned = 'Middle';
                }

                allColumns.add(ColumnD.FieldName + '|' + ColumnD.Sequence + '|' +
pinned);

                // Check whether the current setup contains ProductID field or
not
                if (ColumnD.FieldName == nsPrefix + 'ProductId__r') {
                    isProductIdCount++;
                }

            }

            // Add productID if the view doesn't contain
            if (isProductIdCount == 0) {
                allColumns.add(nsPrefix + 'ProductId__r|1|Left');
```

```
            }

            // Create customsetting data for column values.
            saveFieldList(viewName, allColumns);
        }

        // Flag the processed data and not to migrate in next run.
        configSO.Apttus_Config2__Scope__c = 'Admin';
    }
    if (!nullOrEmpty(configSOs)) {
        update configSOs;
    }
    if (!nullOrEmpty(newCartViewSettings)) {
        insert newCartViewSettings;
    }
}


}
```

Run the following script manually using the developer console:

Run this script only after the abovementioned apex class is created successfully in the org.

```
Savepoint sp = null;
try {
    sp = Database.setSavepoint();
    // Method to create the new Setting Group in New Admin, since Load Defaults will
override existing user customizations
    APTS_CreateCartViews.createCartViewsSettingGroup();

    // Method to migrate old Cart Views to the New Cart Views data model
    APTS_CreateCartViews.migrateCartViewSetting();

} catch (Exception ex) {
    Database.rollback(sp);
    System.debug(ex.getMessage());
}
```

Significance of script

In Summer 2018 Release, Cart Views functionality has been enhanced to include Cart Views Admin setup. For more information, see About Cart Views.

The data model for Cart Views has been updated in order to provide better user experience and enable more features.

You have to migrate to new Cart Views for maintaining data that you created before Summer 2018 Release.

Mechanism of script

The script queries all the Cart Views that were created prior to upgrading to Summer 2018 Release and copies the related data over to the new Data Model.

It checks for Unique Name for the migration to ensure that the same Cart View is not migrated more than once.

It creates the new tab called **Cart Views** under **Cart Page** in New Admin UI. For more information, see Application Management Settings

> ⓘ Old Cart Views were created by individual users and were accessible to all users. Hence after migration such old Cart Views will be marked as Public, so that all users can access them. However, due to the nature of the new enhanced Cart Views, only admin users can edit the Public Cart Views.

Configuring Cart Activity History

Cart Activity History records the updates you make to product configuration. CPQ adds an entry to the **Activity History** when you create, finalize, and reconfigure the product configuration. It helps you track the changes made to the product configuration. You can choose to disable the generation of such history, if not needed. You can use the Admin Setting *APTS_DisableCartActivityHistory* to disable the creation of entries in the **Activity History**. If you create and enable this setting, CPQ does not add entries when you create, finalize, or reconfigure a product configuration.

> ⓘ This setting does not impact the Activity History of Proposal and Agreement object.

To disable cart activity history

You must follow the steps below to configure this setting:

1. Go to **All Tabs** > **Admin**
2. Click **New**, to create a new record. The New Admin page is displayed.
3. Fill the following details

Field	Value
Name	*APTS_DisableCartActivityHistory*
Value	*true* or *false*
Code	Leave the field blank

4. Click **Save**.

Configuring CSS Override

CPQ enables you to implement CSS adjustments on different CPQ pages based on your organization's needs using the **CSS Override** field in Config System Properties. You can use CSS Override to add custom styles to pages like Catalog, Configuration, and Cart, by overriding the default style. For example, you can add logos, images on CPQ pages. You can define CSS Override to specific flows or globally to all flows. When you configure CSS override for a specific flow, the CSS is applied only on that flow. However, when you apply CSS override globally, CPQ also overrides flow-specific CSS adjustments along with CPQ default style. For more information, see About Custom Setting Order of Precedence.

Pre-requisites

Follow the steps below before you configure CSS Override.

1. Create a CSS file with the adjustments you want to implement on CPQ pages.
2. Upload the file as a static resource in your org. For more information on static resources, see Using Static Resources in Salesforce.

To configure CSS Override globally

1. Go to **Setup > Develop > Custom Settings > Config System Properties > Manage**.

2. Find **System Properties** and click **Edit**.
3. Enter the name of the CSS file that you uploaded as static resources in the **CSS Override** field. This enables you to use the styles defined in the CSS file as opposed to default styles.
4. Click **Save**.

To configure CSS Override specifically for a flow

1. Create a new flow-specific Config System Properties data set or open an existing flow-specific data set. For more information, see Configuring Custom Settings for Different Flows.
2. Enter the name of the CSS file that you uploaded as static resources in the **CSS Override** field. This enables you to use the styles defined in the CSS file as opposed to default styles.
3. Click **Save**.

Example

You can use the **CSS Override** field to hide the Line Item Actions such as Price Ramp, available in the menu (⋮) next to each line item on the Cart page. You can define such CSS Override to hide line item action globally or based on a flow. The sample code below is an example of CSS code that you can use to hide the line item menu actions from the Cart page.

```
1  /*To hide price ramp*/
2  #lla-price-ramp {
3      display: none;
4  }
5
6  /*To hide related-lines*/
7  #lla-related-lines {
8      display: none;
9  }
10
11 /*To hide multiple adjustments*/
12 #lla-adjustments {
13     display: none;
14 }
15
16 /*To hide usage tier*/
17 #lla-usage-tier {
```

```
18        display: none;
19    }
20
21    /*To hide incentives*/
22    #lla-incentives {
23        display: none;
24    }
25
26    /*To hide water fall*/
27    #lla-price-waterfall {
28        display: none;
29    }
```

Configuring Mass Option Selection on the Configuration Page

The Mass Option Selection feature allows the Sales rep to select, clear the selection, and update options and attributes in one go before processing pricing and expression and invoking rules. After configuring all options and attributes, the Sales rep must click the **Confirm Option Selection** to initiate the process of pricing and executing rules. You can enable this feature by selecting the **Check Many Options** Custom Setting. When you enable this feature, **Confirm Option Selections** button is enabled on the Configuration and Service Bundle Configuration page. If there are any errors encountered during the process, CPQ displays them after completion of the process.

By default, CPQ calculates pricing, processes relevant expression, and executes associated rules each time you select, clear, or update an option or attribute.

To enable the Mass Option Selection feature

Follow the steps below:

1. Go to **Setup** > **App Setup** > **Develop** > **Custom Settings**.
2. Find **Config System Properties** and click **Manage** next to the setting.
3. Click **Edit** next to **System Properties**.
4. Select **Check Many Options**.
5. Click **Save**.

You can also define **Check Many Options** for specific flows. For more information, see Configuring Custom Settings for Different Flows.

To add the Confirm Option Selection button on the Configuration page

Follow the steps mentioned in the "*To configure action buttons on the Summary panel on the Configuration page*" section in Configuring Display Action Settings to add the **Confirm Option Selections** button on the Configuration page.

Configuring Express Proposal

Express Proposal provides a single place for a sales representative to finalize the cart, select a template and generate the quote, email the document to the customers, and mark the quote as presented, in one place. The feature is invoked upon the click of the **Express Quote** button on the Catalog or Cart page, which must be added by configuring Display Action Settings in Config Settings. CPQ executes the following tasks when the Sales rep clicks the **Express Quote** button:

- Finalize the proposal in asynchronous mode.
- Generate the proposal in the desired template in PDF format.
- Email the proposal to the quote owner.
- Mark the proposal as presented.

You can configure the feature in either:

- Single-click mode: to execute the abovementioned tasks automatically or
- Prompt mode: to prompt the Sales rep to modify the task execution on a popup.

You can use the **showDetails** parameter in the **Action Page** column to drive the mode in which feature is executed. CPQ executes the feature differently for a different value of the **showDetails** parameter, which is described in the table below:

Value of showDetails parameters	Mode	Description
false or blank	Single-click	CPQ invokes the Express Quote feature in single-click mode, where the tasks are executed automatically without displaying the popup on the page where the sales rep clicks the **Express Quote** button. In this case, CPQ uses the default template from **Default Template Name** in Proposal System Properties, or a template filtered by query template filters to generate the document in PDF format and sends the email to the quote owner. ⓘ If no default template is defined or the query template filters returns multiple templates in result, CPQ displays the **Express Quote** popup.
true	Prompt	CPQ invokes the Express Quote feature in Prompt mode and displays the **Express Quote** popup on the page where the Sales rep can select document generation templates, add email recipients, and mark the proposal as presented.

For more information on where to add the **Express Quote** button and define the showDetails parameter, see Configuring Display Action Settings.

For more information on how to use the express proposal, see Using Express Proposal.

Managing Translation of Fields and Values in Different Language

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

You must follow different sets of instructions to translate Salesforce standard fields and Conga CPQ field values.

This section describes the following topics:

- Translating CPQ Record Values
- Translation of Field Labels

Translating CPQ Record Values

Conga CPQ provides the ability to translate the records to other languages. You can use this functionality to display translated values of the following type of records:

- Product
- Category and Option Group

For Sales reps using the org in a locale other than English, you can translate the values as necessary and display them on the Catalog, Configuration, and Cart pages. To translate the values you must add the translated text in the corresponding translation-related list for each record. For example, you must add translated text of the product field details in the **Product Translation** related list for each product.

> ⓘ If you want to translate the custom fields you have created on the Product or Category object, you must create custom fields with the same API names on the Product Translation or Category Translation object as well. After you create the fields, you must add those fields in the Product Translation or the Category Translation edit pages layout.

Translating Product Details field value

You must translate the information in the product details field values in the desired language and add them to the Product Translation related list for each product.

To add translated information in the Product Translation related list

1. In Salesforce Classic UI, go to App Menu > **Conga Products Setup** > **Products**.
2. Open the product for which you want to add translated information.
3. Under **Product Translations** related list, click **New Product Translation**
4. Fill in the following details.

Field	Description
Translation Name	Enter the name of the language.
Currency	Select the currency in which the prices are displayed.
ProductCode	Enter the translated Product Code.
Description	Enter the translated product description.
Product	Product name is populated automatically if you navigated to this page through the Product Details page. Otherwise, search and select the name of the product.
Family	Enter the translated Product Family
Language	Enter the 2 letter ISO code for language. For example, *FR* for French and *JP* for Japanese.
Name	Enter the translated Product Name

5. Click **Save**.

Translating Category Details field value

You must translate the information in the category, sub-category, and option groups detail field values in the desired language and add them to the **Category Translation** related list for each record.

To add translated information in the Category Translation related list

1. In Salesforce Classic UI, go to App Menu > **Conga Products Setup** > **Category**.
2. Open the category, sub-category, or option group record for which you want to add translated information.
3. Go to the Category Hierarchy associated with the record you opened.
4. Under **Category Translations** related list, click **New Category Translation**
5. Fill in the following details.

Field	Description
CategoryTranslation Name	Enter the name of the translation record
Language	Enter the 2 letter ISO code for language. For example, *FR* for French and *JP* for Japanese.
Label	Enter the translated text for the record.
CategoryHierarchy	Enter the name of the Category Hierarchy associated with the record.

6. Click **Save**.

Translate Field Values on the Cart page

Unlike the Catalog and Configuration page, the Cart page retrieves values of fields from the Line Item object. To display the translated details on the Cart page, you must create formula fields on the Line Item object to enable CPQ to use the translated details from the Product Translation and Category Translation object.

To create a formula field in the Line Item object

1. Go to **Setup > App Setup > Create > Objects** and select **Line Item**
2. Under **Custom Fields & Relationships** section, click **New**.
3. Select **Formula** from the Field Type.
4. Click **Next**.
5. In **Field Label**, enter the name of the Product or Category field for which you want to translate the values.

6. Click the **Field Name** textbox, the value is populated automatically.
7. From **Formula Return Type**, select **Text**.
8. Click **Next**.
9. Enter the formula to return the text value of the product field value of which has to be translated. For example, if you want to display the field value of **Product Type** on the cart, enter

    TEXT(Apttus_Config2__ProductId__r.Apttus_Config2__ProductType__c).
10. Establish field-level security and click **Next**.
11. Select appropriate page layouts where you want to add this field.
12. Click **Save.**

Translation of Field Labels

You can use this feature to translate the field labels. There are threes types of fields in a CPQ org, namely standard, custom, and CPQ managed package fields. You must follow different steps to translate the different types of fields.

This section describes the following topics:

- Translating Standard and Custom Field Labels
- Translating and Overriding Managed Packaged Field Label
- Using Translated Picklist Values in Document Generation

Translating Standard and Custom Field Labels

You can configure translation for out-of-the-box field names, picklist values, button labels, and error messages directly in the **Translation Workbench**. For custom label translation, you must upload Salesforce Translation File. Follow the steps below to configure the translation of content.

To configure Translation Workbench for translating the label data

1. Go to **Setup** > **Administration Setup** > **Translation Workbench** > **Translation Language Settings**.
2. Click **Add** button.
3. Select language in the **Language** dropdown.
4. Select **Active** to activate the language.

5. Under **Identify Translators for this Language**, you must define User who can be the translators. Select a User from **Available List** and click **Add**.
6. Click **Save**.

To change the language at the profile level

1. Go to **Setup** > **Manage Users** > **Users**.
2. Click **Edit** next to the desired profile.
3. Under the **Locale Settings**, change the value in the **Language** field and click **Save**.
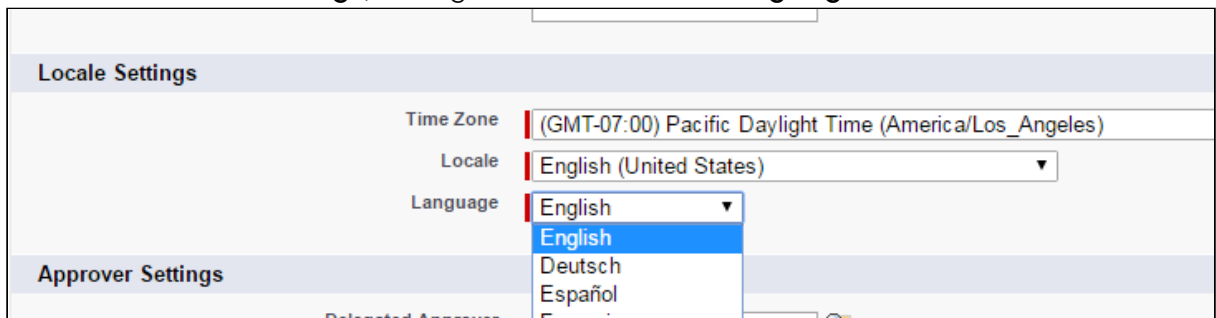
> ⓘ You must modify the **Language** field to match the **Language** field in the Product Translation and Category Translation records. CPQ only displays the translated details when the **Language** field matches in both the records. However, the Locale itself is not a criterion.

To mass upload translations for custom labels

You need to upload a new Salesforce Translation File (STF) through the Translation Workbench to translate customer labels

1. Go to **Setup** > **Administration Setup** > **Translation Workbench** > **Export**
2. Select **Bilingual**.
3. Click **Export as STF**
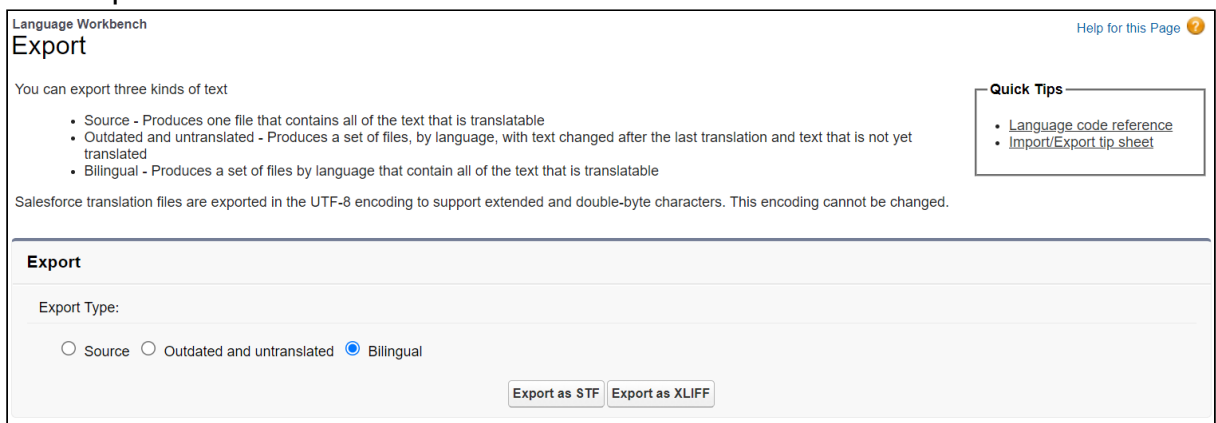
4. Translate the STF (non-translated items will show up in English).
5. Go to **Setup** > **Translation Workbench** > **Import**
6. Click **Choose File**.
7. Select the translated STF file.
8. Click **Import**.

You can translate the field labels to the desired language.

To translate an individual custom label

1. Go to **Setup > Create > Custom Labels**.
2. Search and open the label you want to translate. The system displays the Custom Label Detail page.
3. Click the button, **New Local Translations/Overrides** in the Local Translations/Overrides section. The system displays the New Translation page.
4. Select the appropriate language from the drop-down.
5. Enter the Translation Text as an Add-on in the text box.
6. Click **Save**.

Translating and Overriding Managed Packaged Field Label

By default, CPQ provides the labels in English and Japanese language. You must use the Override feature to customize the labels of components for CPQ managed package objects in English and Japanese. Whereas, you must use the Translate feature to translate the labels to any other language. Follow the steps below based on the task you want to acheive.

To override CPQ labels in English and Japanese languages

1. Go to **Setup** > **Administration Setup** > **Translation Workbench** > **Override**.
2. Define the fields based on the below description.

Field	Description
Package	Select the Conga package where the label you want to translate belongs.
Language	Select the desired language.
Setup Component	From the dropdown, select the type of component that you want to translate. For example, if you want to translate the label of the custom field, select *Custom Field* in **Setup Component.**
Object	Select the object where the component belongs to.
Aspect	Select the type of label you want to translate.

3. Find the label that you want to translate and add the translated text of the label in the **Field Label Override** column in the respective row.
4. Click **Save.**

To translate CPQ labels to languanges other than English and Japanese

1. Go to **Setup** > **Administration Setup** > **Translation Workbench** > **Translate**.
2. Define the fields based on the below description.

Field	Description
Language	Select the desired language.
Setup Component	From the dropdown, select the type of component that you want to translate. For example, if you want to translate the label of a button, select *Button and Link Label* in **Setup Component.**
Object	Select the object where the component belongs to.

3. Find the component label that you want to translate and enter the translated text of the label in the **Field Label Override** column in the respective row.
4. Click **Save**.

Use Case: Overriding Label of a Button

In the scenario where you want to change the label of a CPQ managed package components. For example, you need to override the label of the field **Hierarchy Manager** button to **Category Manager**. Follow the steps below.

1. Go to **Setup** > **Administration Setup** > **Translation Workbench** > **Override**.
2. Define the fields based on the below description.

Field	Description
Package	Select the **Conga Configuration & Pricing**.
Language	Select **English**.
Setup Component	From the dropdown, **Button and Link Label**.
Object	Select **Category**.
Aspect	When you select **Button and Link Label**, the field **Aspect** is not required.

3. Find the label **Hierarchy** in the **Master Button or Link Label**.
4. Enter **Category Manager** in the **Field Label Override** column in the respective row.

5. Click **Save**.

The overridden value is displayed on the Category Detail page.
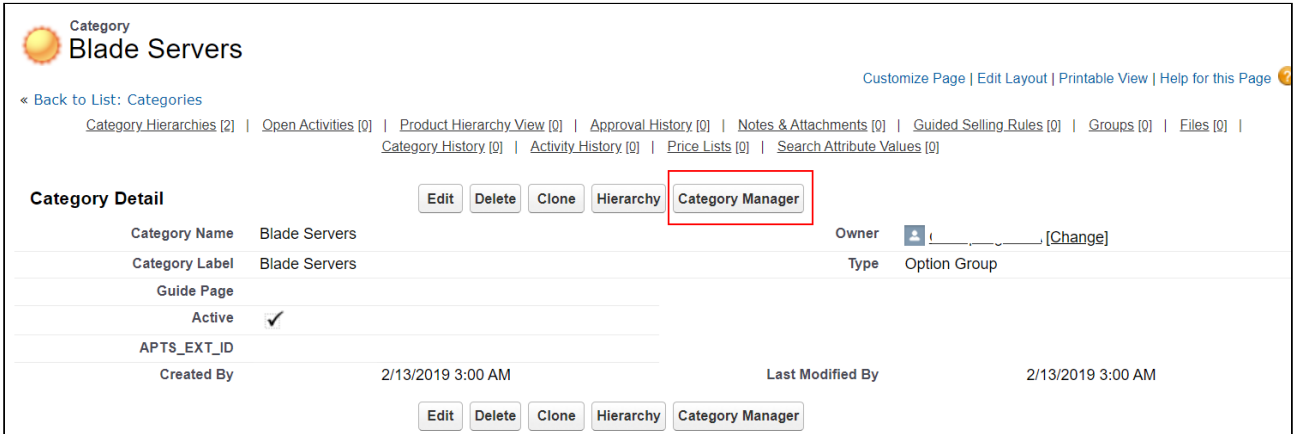
Use Case: Translating Label of a Button

In the scenario where your company has a base in the European regions and you need to translate the field labels in French for the Sales Rep working in a French locale. For example, you need to translate the label of the field **Hierarchy Manager** button to french. Follow the steps below.

1. Go to **Setup** > **Administration Setup** > **Translation Workbench** > **Override**.
2. Define the fields based on the below description.

Field	Description
Language	Select the **French**.
Setup Component	From the dropdown, **Button and Link Label**.
Object	Select **Category**.
Aspect	When you select **Button and Link Label**.

3. Find the label **Hierarchy** in the **Master Button or Link Label**.

4. Enter the translated text of the label in the **Field Label Override** column in the respective row.
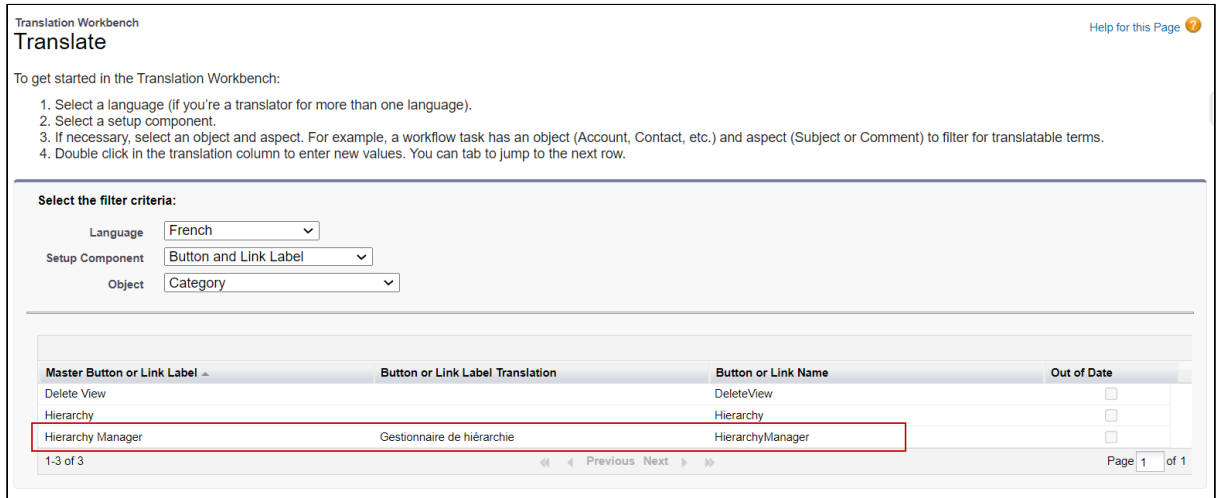
5. Click **Save**.

The translated value is displayed on the Category Details page.
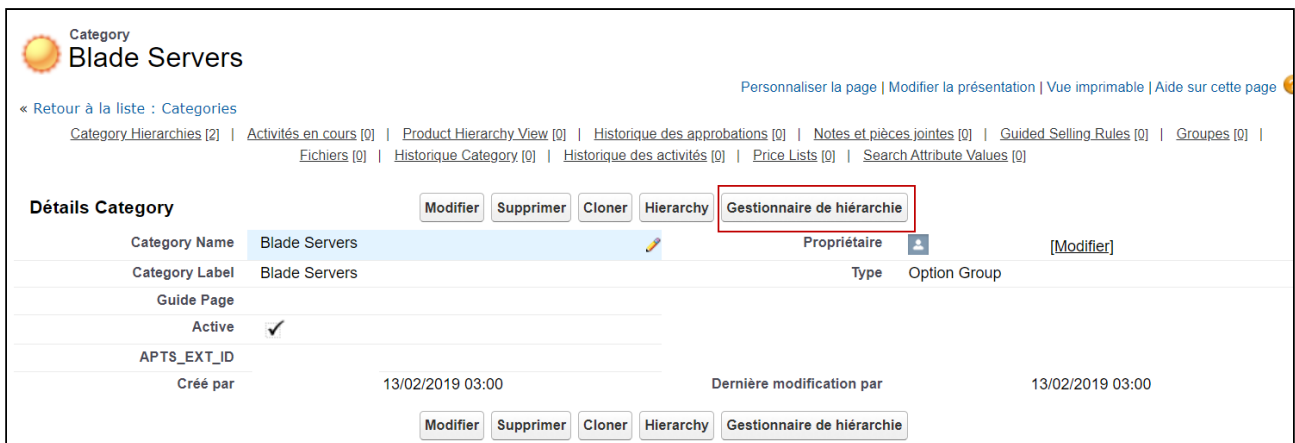
Using Translated Picklist Values in Document Generation

You can enable non-English-speaking users to get translated pick list values during document generation. To do this, set the APTS_TranslatePickValuesInDocs admin setting to true. If you do not configure this admin setting or set it to false, CPQ uses the values defined in English in the generated document instead of the translated values.

To enable use of translated picklist values in document generation

1. Go to **All Tabs** > **Admin.**
2. Click **New** to create a new record.
3. Fill in the these details:

Field	Value
Name	*APTS_TranslatePickValuesInDocs*
Value	*true*
Code	Leave the field blank

4. Click **Save**.

Customizing CPQ Using Callbacks

Callbacks provide you with a mechanism to apply custom logic on different components of CPQ like line items, Cart page, Catalog page at run-time. For example, you can apply custom pricing or validation on the line items in the cart using Pricing Callback Class and Validation Callback Class. Callbacks are implemented using interfaces that are specific to each callback. These interfaces have various methods you can use to achieve your task. You must implement the interface in an Apex class and within that Apex class, you must configure your custom logic using the methods of the interface. Once you implement the interface, register that Apex class in Custom settings to invoke the callback class on run-time.

To register Callback Class in Custom Settings

To use the Callback Class you created in Apex Classes you must register the Callback Class in the **Config Custom Classes** Custom Settings. Follow the steps below to register the Callback Class

1. Go to **Setup** > **Develop** > **Custom Settings**.
2. Click **Manage** next to **Config Custom Classes**.
3. Click **Edit** next to the **Config Custom Classes** dataset.

   > ⓘ   Avoid creating multiple datasets. Only create a new dataset if the **Config Custom Classes** dataset is missing. You must create a new entry with the name **Config Custom Classes.**

4. Enter the name of the custom Apex class next to the Callback you implemented.
5. Click **Save**.

In this section, the following Callbacks are described:

- Display Action Callback Class
- Product Filter Callback Class
- Option Filter Callback Class
- Adjustment Line Item Callback Class
- Validation Callback Class
- Pricing Callback Class
- Pricing Extension Callback Class

- Product Attribute Callback Class
- Related Pricing Callback Class
- Asset Line Item Callback Class
- Asset Renewal Custom Callback Class
- Revalidation Callback Class
- Action Params Callback Class
- Action Invoker Callback Class
- Action Callback Class
- Cart Approval Callback Class
- Deal Optimizer Callback Class
- Adjustment Spread Callback Class
- Lifecycle Callback Class
- Tax Callback Class
- Formula Callback Class
- Metadata Callback Class
- Recommendation Callback Class
- Advanced Approval Callback Class

Display Action Callback Class

Display Action Callback provides you a mechanism to disable any of the action buttons on the cart. Following are few examples of custom buttons that you can disable:

- Abandon
- Approvals
- Generate
- Quick Save

To use the Display Action Callback you must create a custom Apex class that implements the *Apttus_Config2.CustomClass.IDisplayActionCallback* interface and register the custom apex class with Display Action Callback Class. You must write your custom logic in the custom apex class.

The following methods are available in the
 Apttus_Config2.CustomClass.IDisplayActionCallback interface:

Method	Signature	Description
setActionProperties()	*void setActionProperties(Apttus_Config2.CustomClass.ActionParams)*	You can use this method to enable or disable an action button.

Method	Signature	Description
start()	*void start(Apttus_Config2.ProductConfiguration, List)*	This is the first method that is executed when the callback is invoked.

Example

This is just a sample callback custom class to hide the visibility of the **Abandon** and **Generate** action buttons on the cart page. You may change the custom class to fit your business requirements

```
global with sharing class SampleDisplayActionCallback 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 == 'Abandon'){
                action.isEnabled = false;
            }
            if (action.ActionSO.Apttus_Config2__ActionName__c == 'Generate'){
                action.isEnabled = false;
            }
        }
    }
}
```

Product Filter Callback Class

Product Filter Callback provides you a mechanism to control the visibility of products on the catalog page based on custom criteria that you define. You can define entry criteria to determine whether a product in a price list should be filtered to a certain subset of the full list. You can define the criteria to select standalone products and option products as well.

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

To use the Product Filter Callback you must create a custom Apex class that implements the *Apttus_Config2.CustomClass.IProductFilterCallback* interface and register the custom apex class with Product Filter Callback Class. You must write your custom logic in the custom apex class.

The following methods are available in the *Apttus_Config2.CustomClass.IProductFilterCallback* interface:

Method	Signature	Description
getOptionFilterExpr()	*String getOptionFilterExpr(Apttus_Config2.CustomClass.ActionParams)*	You can use this method to control the visibility of option on the Catalog page.
getProductFilterExpr()	*String getProductFilterExpr(Apttus_Config2.CustomClass.ActionParams)*	You can use this method to control the visibility of Stanalone and bundle products on the Catalog page.

Example

This is just a sample callback custom class to control the visibility of products on the Catalog page. The following only 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.

```
1   global with sharing class SampleProductFilterCallback implements
     Apttus_Config2.CustomClass.IProductFilterCallback {
2       /**
3        * Callback to return part of SOQL filter clause
4        * This filter is used in listing catalog products
5        * @param params is the Apttus_Config2.CustomClass.ActionParams that
6        * contains accountIds, locationIds when available
7        * @return The query filter is like the following.
8        * Name LIKE 'A%' AND Quantity__c > 100
9        * Id IN ('000123', '000124')
10       */
11      global String
     getProductFilterExpr(Apttus_Config2.CustomClass.ActionParams params) {
12          return 'Color__c = \'Red\'';
```

```
13          }
14
15          /**
16           * Callback to return part of SOQLfilter clause
17           * This filter is used in listing option products
18           * @param parsms is the CustomClass.ActionParams that contains
            accountIds, locationIds when available
19           * @return The query filter is like the following.
20           * Name LIKE 'A%' AND Quantity__c > 100
21           * Id IN ('000123', '000124')
22           */
23          global String getOptionFilterExpr(CustomClass.ActionParams params) {
24              return 'ComponentProductId__r.Color__c =\'Green\'';
25          }
26      }
```

Option Filter Callback Class

Option Filter Callback provides you with a mechanism to control the visibility of the option products on the Configuration page based on the custom logic that you define. The callback is executed in the following scenarios:

- when you click **Configure** on the Catalog page
- when the configuration of the product is pending and you click the wrench icon (🔧) on the Cart page.

When callback is invoked, CPQ excludes the option products on the Configuration page.

> ⓘ Default options and options included based on constraint rules are also hidden when the Option Filter Callback is used to hide options. In the scenario, where all the case all the options in an option group are hidden, the option group is also hidden on the Configuration page.

To use the Option Filter Callback you must create a custom Apex class that implements the *Apttus_Config2.CustomClass.IOptionFilterCallback* interface and register the custom apex class with Option Filter Callback Class. You must write your custom logic in the custom apex class.

The following methods are available in the
Apttus_Config2.CustomClass.IOptionFilterCallback interface:

Method	Signature	Description
getExcludedOptionIds()	*List getExcludedOptionIds(Apttus_Config2.CustomClass.ActionParams)*	You can use this method to define custom logic to hide options on the Configuration page.

Example

This is just a sample callback custom class to control the exclusion of option products on the catalog page. The following custom code only removes a few option products. You may change the custom class to list out the ID of option products to fit your business requirements.

```
global with sharing class SampleOptionFilterCallback implements
 Apttus_Config2.CustomClass.IOptionFilterCallback {

    /**
     * Callback to return option IDs which are to be excluded from the bundle
     * This filter is used when we configure a bundle.
     * @param params is the CustomClass.ActionParams that contains bundleId and
productIds
     * @return List of option product IDs which will be excluded
     */
    global List<ID> getExcludedOptionIds(Apttus_Config2.CustomClass.ActionParams
params) {
        //return new List<ID>{ <18 CHARACTER PRODUCT ID> };
        //Example for returning THREE products to be excluded:
            return new List<ID>{ '01t3C000000DVwh', '01t3C000000DVwm','01t3C000000DVw
w'};
        //Example for returning ONE product to be excluded:
        //  return new List<ID>{ 01t50000004nmL1YYX'};
    }
}
```

Adjustment Line Item Callback Class

Adjustment Line Item Callback allows you to write custom logic to auto-create or auto-delete adjustments for the line items. For example, once a product is added to cart, some adjustments need to be auto-included based on the account contract or other terms. In

such a case, you can use Adjustment Line Item callback to automate this process. When you apply an adjustment on a proposal, the callback will overwrite manual adjustments.

Agents can change the discounts anytime and that will not reflect on the cart. Agents cannot do this manually per line item, so we have given the custom page for them to manage the discounts.

To use the Adjustment Line Item Callback you must create a custom Apex class that implements the *Apttus_Config2.CustomClass.IAdjustmentLineItemCallback2* interface and register the custom apex class with Adjustment Line Item Callback Class. You must write your custom logic in the custom apex class.

The following methods are available in the
 Apttus_Config2.CustomClass.IAdjustmentLineItemCallback2 interface:

Method	Signature	Description
createAdjustmentLineIte ms()	*List createAdjustmentLineItems(Apt tus_Config2.LineItem)*	You can use this method to create adjustments for the given parent line item
finish()	*void finish()*	This the last method that is executed after the callback is invoked.
start()	*void start(Apttus_Config2.ProductCo nfiguration)*	This is the first method that is executed when the callback is invoked.
beforeSaveAdjustmentLin eItems()	*Boolean beforeSaveAdjustmentLineItems (Apttus_Config2.LineItem, List)*	You can use this method to apply custom logic after the user adjustments are finished and before the changes are saved.

Example

The given code example implements *createAdjustmentLineItems()* method to create new adjustment line items using the *IAdjustmentLineItemCallback2* interface.

```
1   global with sharing class SampleAdjustmentCallback
2                   implements
      Apttus_Config2.CustomClass.IAdjustmentLineItemCallback,
```

```
 3    Apttus_Config2.CustomClass.IAdjustmentLineItemCallback2 {

 4
 5        /**
 6         * Callback at the beginning of the adjustment call.
 7         * Use the start method to initialize state
 8         * @param cart the cart object
 9         */
10        global void start(Apttus_Config2.ProductConfiguration cart) {
11            // do nothing
12        }

13
14        /**
15         * Callback to create adjustments for the given parent line item
16         * @param parentItem the parent line item associated with the
      adjustments
17         * @return the list of adjustment line item sobjects
18         */
19        global List<Apttus_Config2.AdjustmentLineItem__c>
      createAdjustmentLineItems (

20
      Apttus_Config2.LineItem parentItem) {

21
22            List<Apttus_Config2.AdjustmentLineItem__c> adjItems = new
       List<Apttus_Config2.AdjustmentLineItem__c>();

23
24            // test data
25            if (Test.isRunningTest() == true &&
      SystemUtil.isMultiAdjustmentEnabled() == true) {
26                // get the parent item sobject
27                Apttus_Config2__LineItem__c parentItemSO =
      parentItem.getLineItemSO();

28
29                // create a line item adjustment
30                //AdjustmentLineItem__c adjItemSO = new
      AdjustmentLineItem__c(LineItemId__c = parentItemSO.Id);
31                Apttus_Config2__AdjustmentLineItem__c adjItemSO = new
       Apttus_Config2__AdjustmentLineItem__c();
32                // line number
33                adjItemSO.LineNumber__c = 1;
34                // line type
35                adjItemSO.LineType__c = 'Manual';
36                // modifiable
37                adjItemSO.IsModifiable__c = true;
38                // adjustment type
```

```
39              adjItemSO.AdjustmentType__c = '% Discount';
40              // adjustment amount
41              adjItemSO.AdjustmentAmount__c = 10;
42              // adjustment applies to
43              adjItemSO.AdjustmentAppliesTo__c ='Starting Price';
44
45              adjItems.add(adjItemSO);
46          }
47          return adjItems;
48      }
49      /**
50       * Callback before the adjustment line items are saved after user
    changes
51       * @param parentItem the parent line item associated with the
    adjustments
52       * @param adjItems the list of adjustment line items associated with
    the save operation
53       * @return <code>true</code> if adjustment line items were modified by
    the callback,
54       * <code>false</code> otherwise
55       */
56      global Boolean beforeSaveAdjustmentLineItems(Apttus_Config2__LineItem
    parentItem,
57                                          List<
    Apttus_Config2__AdjustmentLineItem__c> adjItems) {
58          return false;
59      }
60
61      /**
62       * Callback after all batches of adjustment line items are processed
63       * Use the finish method to release state
64       */
65      global void finish() {
66          // do nothing
67      }
68  }
```

Validation Callback Class

Validation Callback provides a mechanism to implement custom validations on the line items in the cart. Validations are any constraints you define on the line item to restrict the selection of that line item based on any condition or criteria. When you define validations,

CPQ checks if the condition or the criteria is satisfied and displays an error message accordingly. For example, you can validate if the **Quantity** field for any product is negative and prevent the sales rep from finalizing the cart.

Validation Callback applies validation on the following aspects in the configuration:

- Line items in the cart
- Price Ramps

To use the Validation Callback you must create a custom Apex class that implements the *Apttus_Config2.CustomClass.IValidationCallback3* interface and register the custom apex class with Validation Callback Class. You must write your custom logic in the custom apex class.

The following methods are available in the *Apttus_Config2.CustomClass.IValidationCallback3* interface:

Method	Signature	Description
validateCart()	*Apttus_Config2.CustomClass.ValidationResult validateCart (Apttus_Config2.CustomClass.ActionParams, Apttus_Config2.ProductConfiguration)*	You can use this method to apply validation on line items in the cart. This method is invoked when the Sales rep performs the following actions: • **View Cart** (i) To invoke Validation Callback when click **View Cart** on the Mini Cart, you must enable **Run Validation Callback On Add** in Config System Properties custom setting. For more information, see Config System Properties. • **Reprice** • **Save** • **Finalize** • **Analyze Deal** • **Validate** on the Configuration page. • Any action that is available on the Cart and Catalog page. • Custom actions

Method	Signature	Description
validateRampLineItems()	*Apttus_Config2.CustomClass.ValidationResult validateRampLineItems (Apttus_Config2.CustomClass.ActionParams, Apttus_Config2.ProductConfiguration, List)*	You can use this method to apply validation on the ramp line items in a Price Ramp. This method is invoked when the Sales rep performs the following actions: • Click **Save** on the Price Ramp pop-up • Any action that is available on the Cart and Catalog page when price ramps exist for any line item.

The following are examples of the custom logic you can implement using Validation example

Example

You can use *validateCart()* apply the following validations:

• Check if the **Duration** for any product is null.
• Check if the value in the **Adjustment Amount** field is negative.

The given code example implements two methods of *IValidationCallback3* interface. The code sample implements the abovementioned validation using the method **validateCart()**.

```
global with sharing class Test_ValidationCallBack_UC1 implements
 Apttus_Config2.CustomClass.IValidationCallback3 {

    global Apttus_Config2.CustomClass.ValidationResult
validateCart(Apttus_Config2.CustomClass.ActionParams params,
Apttus_Config2.ProductConfiguration cart) {
        system.debug('***ValidationCallBack > validateCart() > ' + cart);

        Apttus_Config2.CustomClass.ValidationResult result = new
 Apttus_Config2.CustomClass.ValidationResult(true);

        list <Apttus_Config2.LineItem> allLines = cart.getLineItems();
        list<Apttus_Config2__LineItem__c> allLineItems = getLineItems(allLines);


        Id configurationId;
        if(allLineItems != null) {
            configurationId = allLineItems[0].Apttus_Config2__ConfigurationId__c;
        }
```

```
        Integer i = 0;
        System.debug('***ValidationCallBack > allLineItems: '+allLineItems.size() + '
<> ' + allLineItems);
        for(Apttus_Config2__LineItem__c lineItem : allLineItems) {
            system.debug('***ValidationCallBack > ' + (++i) + ' > lineItem : ' +
lineItem);

            // Check if the Duration for any product is null.
            if(lineItem.Apttus_Config2__SellingTerm__c == null) {
                result.Messages.add(new ApexPages.Message(ApexPages.Severity.ERROR,
'Value of Duration for product : \'' + lineItem.Apttus_Config2__Description__c + '\'
is None.'));
                result.isSuccess =  false;
            }
            // Check if the value in the Adjustment Amount field is negative.
            if(lineItem.apttus_config2__adjustmentamount__c < 0) {
                result.Messages.add(new ApexPages.Message(ApexPages.Severity.ERROR,
'Adjustment Amount for product : \'' + lineItem.Apttus_Config2__Description__c + '\'
is Negative. It should be >= 0'));
                result.isSuccess =  false;
            }
        }
        return result;
    }


    global Apttus_Config2.CustomClass.ValidationResult
validateRampLineItems(Apttus_Config2.CustomClass.ActionParams params,
Apttus_Config2.ProductConfiguration cart,List<Apttus_Config2.LineItem> lstLI) {
        system.debug('***ValidationCallBack > validateRampLineItems() > Cart : ' +
cart + ' <> lstTemp : ' + lstLI);
        Apttus_Config2.CustomClass.ValidationResult result;
        return result;
    }


    /* Gets the list of product line items associated with the Battery line
     * @param cart the cart object
     * @return the list of line item objects
    */
    private static List<Apttus_Config2__LineItem__c>
getLineItems(List<Apttus_Config2.LineItem> allLines) {
        list<Apttus_Config2__LineItem__c> lineItems = new
 list<Apttus_Config2__LineItem__c>();
```

```
        // iterate through the cart and get the line items matching the battery code1
        for (Apttus_Config2.LineItem lineItemMO : allLines) {
            lineItems.add(lineItemMO.getLineItemSO());
        }

        return lineItems;
    }
}
```

Pricing Callback Class

The 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. The pricing callback is invoked during pricing action and executes the various methods in the BASEPRICE and ADJUSTMENT modes.

To use the Pricing Callback you must create a custom Apex class that implements the _Apttus_Config2.CustomClass.IPricingCallback3_ interface and register the custom apex class with Pricing Callback Class. You must write your custom logic in the custom apex class.

The following methods are available in the _Apttus_Config2.CustomClass.IPricingCallback3_ interface:

Method	Signature	Description
afterPricing()	_void afterPricing(Apttus_Config2.ProductConfiguration.LineItemColl)_	You can use this method to define the logic to be executed after Net Price calculation. The collection of line items currently in the cart is passed as an input parameter to this method.
afterPricingLineItem()	_void afterPricingLineItem(Apttus_Config2.ProductConfiguration.LineItemColl, Apttus_Config2.LineItem)_	You can use this method to define custom logic that has to be applied after the Net Price has been calculated. This method called for each line item in the cart.

Method	Signature	Description
beforePricing()	*void beforePricing(Apttus_Config2.ProductConfiguration.LineItemColl)*	You can use this method to define the logic for net price calculation. In this method, no pricelist item is associated to the cart line items. Price Matrix and Price Ruleset calculation occurs when the mode of the callback is BASEPRICE Mode.
beforePricingLineItem()	*void beforePricingLineItem(Apttus_Config2.ProductConfiguration.LineItemColl, Apttus_Config2.LineItem)*	You can use this method to define custom logic that has to be applied after the **Base Price** has been calculated but before the downstream prices like **Base Extended Price**, and **Extended Price** is calculated. This method called for each line item in the cart.
finish()	*void finish()*	This is the last method that is executed after the callback is invoked.
setMode()	*void setMode(Apttus_Config2.CustomClass.PricingMode)*	You can use this method to specify 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**. • **BASEPRICE**: The mode is base price when you click **Go to Price**.
start()	*void start(Apttus_Config2.ProductConfiguration)*	This is the first method that is executed when the callback is invoked.
onPriceItemSet()	*void onPriceItemSet(Apttus_Config2__PriceListItem__c, Apttus_Config2.LineItem)*	You can use this method to associate custom fields with line items and apply logic related to price list item. This method is called on each line item.

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 needs 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()*

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 all or some of the methods in the class will be called during each mode.
- The most widely used modes are either BASEPRICE or ADJUSTMENT.
- It is recommended to have logic in the BASEPRICE mode only, in the *beforePricing()*, *beforePricingLineItem()*, *afterPricing()*, and *afterPricingLineItem()* methods.

ADJUSTMENT mode is an 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.
- ADJUSTMENT mode is not executed when the parameter **isCartTotalingDisabled** is set to *true* in the **Configure Products** button URL.
- Any DML statement is not required. All changes related to line items are covered regardless of DML. However, if there are other object record changes then make sure that the query is not called multiple times in a single transaction of pricing. 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 the line item.
- If your logic is dependent on **Base Price** calculation, then make sure to use **Base Price Override** field of Line Item. You can put the newly computed **Base Price** in this field and then in the next mode, the system will take the value of **Base Price** from the override field to compute **Extended Price** and **Net Price**.
- When you do a **Reprice** on the Cart page and change any field on the line item, which is used in Price Matrix or Price Ruleset, then BASEPRICE mode will get called along with ADJUSTMENT mode. If you change fields on the line item which are not impacting Base Price Calculations (e.g. Giving 10% discount on a bundle line item), then only ADJUSTMENT mode is 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 the 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.
- The custom fields on Price List Item are not retrieved in the **onPriceItemSet** method automatically. To auto-retrieve the custom fields in this method, the Administrator must create a new entry named *APTS_PriceListItemCustomFields* in **Admin**. Enter the API names of the custom field of the Price List Item in the **Code** field separated by a comma or on a new line.

```
/**
 * 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() {
 }

}
```

Pricing Extension Callback Class

You can use the Pricing Extension Callback Class to define any logic or computation you need to execute before or after the pricing is calculated for the line items. You must define the pre-pricing or post-pricing hooks in the Pricing Extension Callback. The pre-pricing and post-pricing logic are executed separately from the pricing that is calculated when the Sales rep clicks **Submit for Pricing(Async)** button. This allows CPQ to run the pricing without executing custom logic every time.

The pre-pricing and post-pricing logic are executed when the Sales rep clicks **Pre-Price** and **Post-Price** buttons respectively, on the Cart page. If you define **Auto Execute Post-Pricing Step** and **Auto Execute Pre-Pricing Step** settings, the pre-pricing and post-pricing logics are executed automatically when the Sales rep clicks **Submit for Pricing(Async)**. For more information, see Config System Properties.

The following methods are available to you in Pricing Extension Callback Class.

Method	Signature	Description
afterPricing()	*Apttus_Config2.CustomClass.PricingExtensionResult afterPricing(Apttus_Config2.ProductConfiguration, Apttus_Config2.CustomClass.CartHelper)*	You can use this method to execute logic that is required post-processing after line items are priced
beforePricing()	*Apttus_Config2.CustomClass.PricingExtensionResult beforePricing(Apttus_Config2.ProductConfiguration, Apttus_Config2.CustomClass.CartHelper)*	You can use this method to execute logic that is required pre-processing to prepare the cart line items for pricing.
isAfterPricingEnabled()	*Boolean isAfterPricingEnabled()*	You can use this method to determine whether after-pricing should be invoked for the callback.
isBeforePricingEnabled()	*Boolean isBeforePricingEnabled()*	You can use this method to determine whether before-pricing should be invoked for the callback.

Pre-Pricing Logic

You can define logic in beforePricing() method of Pricing Extension callback to execute any computation that drives the pricing of the line item. For example, you can calculate the Total Quantity of reagents across all the locations and use that quantity to determine the tiered price for a line item. You must specify the fields you apply pre-price to in **Custom Pre-Pricing Fields**. For more information, see Config System Properties.

Post-Pricing Logic

You can define logic in afterPricing() method of Pricing Extension Callback to execute any computations that are dependent on the pricing fields like **Net Price** and **Net Extended Price** which are calculated after pricing. For example, if you want to calculate the annual value for a given line item with 3 years of Selling Term and after applying adjustments the Net Price is 3000, then the annual value is 3000/3 = 1000.

Example Code

```
/**
 *  Apttus Config & Pricing
 *  DefaultPricingExtensionCallback
 *
 *  @2019-2020 Apttus Inc. All rights reserved.
 */
global with sharing class SamplePricingExtensionCallback3 implements
 Apttus_Config2.CustomClass.IPricingExtensionCallback
{

    /**
     * Indicates whether the before pricing callback is enabled
     * Use this method to determine whether before pricing should be invoked for the
callback
     * @return <code>true</code> if before pricing callback is enabled, <code>false</
code> otherwise
     */
    global Boolean isBeforePricingEnabled()
    {
        return true;
    }

    /**
```

```
     * Callback before pricing the given cart
     * Use this method to do all required pre-processing to prepare the cart line
items for pricing.
     * @param cart the cart object to pre-process
     * @param helper the cart helper class
     * @return the pricing extension result object
     */
    global Apttus_Config2.CustomClass.PricingExtensionResult
beforePricing(Apttus_Config2.ProductConfiguration cart,
Apttus_Config2.CustomClass.CartHelper helper)
    {
        Id cartId = cart.getConfigSO().Id;
        List<Apttus_Config2__LineItem__c> lineItemsToUpdate = new
 List<Apttus_Config2__LineItem__c>();
        List<Apttus_Config2__LineItem__c> lineItems = [SELECT Id, Name ,

Total_reagent_qty_for_the_site__c ,

                                                      PrePricingCriteriaField__c,
                                                      Apttus_Config2__Quantity__c,
                                                      Apttus_Config2__SyncStatus__c,

Apttus_Config2__PricingStatus__c

                                                      FROM
Apttus_Config2__LineItem__c

                                                      WHERE
Apttus_Config2__ConfigurationId__c = :cartId];

        system.debug ('******************** lineItems:' + lineItems);

        Decimal totalQty = 0;
        for (Apttus_Config2__LineItem__c lineItemSO : lineItems)
        {
            //lineItemSO.Apttus_Config2__Quantity__c = 10;
            totalQty += lineItemSO.Apttus_Config2__Quantity__c +
(lineItemSO.PrePricingCriteriaField__c != null
                                                        ?
lineItemSO.PrePricingCriteriaField__c
                                                        : 0);
            lineItemSO.Apttus_Config2__PricingStatus__c = 'Pending';
            lineItemSO.Apttus_Config2__SyncStatus__c = 'Pre-Pricing Complete';
            lineItemsToUpdate.add(lineItemSO);
        }
```

```
        for (Apttus_Config2__LineItem__c lineItemSO : lineItemsToUpdate)
        {
            lineItemSO.Total_reagent_qty_for_the_site__c = totalQty;
        }

        Database.update (lineItemsToUpdate);

        lineItemsToUpdate = new List<Apttus_Config2__LineItem__c>();

        List<Id> pricingCartIds = helper.getPricingCartsFor(cartId);
        for (Apttus_Config2__LineItem__c pricingItemSO : [SELECT Id, Name
                                        FROM Apttus_Config2__LineItem__c
                                        WHERE Apttus_Config2__ConfigurationId__c
IN :pricingCartIds])
        {

            //pricingItemSO.Apttus_Config2__Quantity__c = 10;
            pricingItemSO.Total_reagent_qty_for_the_site__c = totalQty;
            pricingItemSO.Apttus_Config2__SyncStatus__c = 'Pre-Pricing Complete';
            lineItemsToUpdate.add(pricingItemSO);
        }


        Database.update (lineItemsToUpdate);
        return new Apttus_Config2.CustomClass.PricingExtensionResult();

    }

    /**
     * Indicates whether the after pricing callback is enabled
     * Use this method to determine whether after pricing should be invoked for the
callback
     * @return <code>true</code> if after pricing callback is enabled, <code>false</
code> otherwise
     */
    global Boolean isAfterPricingEnabled() {
        return true;

    }

    /**
     * Callback after pricing the given cart
     * Use this method to do all required post-processing after line items are
priced.
     * @param cart the cart object to post-process
```

```
    * @param helper the cart helper class
    * @return the pricing extension result object
    */
    global Apttus_Config2.CustomClass.PricingExtensionResult
afterPricing(Apttus_Config2.ProductConfiguration cart,
Apttus_Config2.CustomClass.CartHelper helper) {
        Id cartId = cart.getConfigSO().Id;
        List<Apttus_Config2__LineItem__c> lineItemsToUpdate = new
 List<Apttus_Config2__LineItem__c>();

        List<Id> pricingCartIds = helper.getPricingCartsFor(cartId);
        for (Apttus_Config2__LineItem__c lineItemSO : [SELECT Id, Name,
Apttus_Config2__SyncStatus__c

                                    FROM Apttus_Config2__LineItem__c
                                    WHERE Apttus_Config2__ConfigurationId__c
                                        IN :pricingCartIds]) {

            lineItemSO.Apttus_Config2__SyncStatus__c = 'Post-Pricing Complete';
            lineItemSO.Apttus_Config2__Description__c = 'Net Price should be updated
as per the new quantity 10';
            //lineItemSO.Custom_Field__c = 'Post-price';
            lineItemsToUpdate.add(lineItemSO);
        }

        Database.update (lineItemsToUpdate);
        return new Apttus_Config2.CustomClass.PricingExtensionResult();
    }

}
```

Product Attribute Callback Class

Product Attribute Callback allows you to set the default value of the product attributes based on custom logic at run-time. The callback is invoked when you open the Configuration page to select attributes and options for the products.

To use the Product Attribute Callback you must create a custom Apex class that implements the *Apttus_Config2.CustomClass.IProductAttributeCallback* interface and register the custom apex class with Product Attribute Callback Class. You must write your custom logic in the custom apex class.

The following methods are available in the
Apttus_Config2.CustomClass.IProductAttributeCallback interface:

Method	Signature	Description
finish()	*void finish()*	This the last method that is executed after the callback is invoked.
setDefaults()	*void setDefaults(Apttus_Config2.LineItem, Apttus_Config2__ProductAttributeValue__c)*	You can use this method to set the default value of product attributes. This method is invoked only when you enable **Has Attribute** field for the product.
start()	*void start(Apttus_Config2.ProductConfiguration)*	This is the first method that is executed when the callback is invoked.

⚠ You must enable **Has Attribute** field in product details. Otherwise, when the callback is invoked, only the **start()** and **finish()** methods are called and the callback does not set the default values for product attributes.

Example

In the following example, the Product Attribute Callback sets the default attributes of the Car object.

```
/*
 * Product Attribute Call Back example.
 *
 * Configure a proposal in test org, which has PriceList = 'CPQ 202'
 * Configure Product = 'Car',
 * and verify the default attributes
 * prodAttr.Exterior_Color__c = 'Silver'
 * prodAttr.Leather_Color__c = 'Brown'
 * prodAttr.Inlay__c = 'Walnut'
 * prodAttr.Fuel_Type__c = 'Diesel'
 * prodAttr.Fuel_Badge__c = 'No'
 */
global with sharing class SampleProductAttributeCallback implements
  Apttus_Config2.CustomClass.IProductAttributeCallback {
```

```
    Id accountId;

    global void start(Apttus_Config2.ProductConfiguration config) {
        system.debug('***ProdAttrCallBack > start() > ' + config);


    }

    global void setDefaults(Apttus_Config2.LineItem lineItem,
Apttus_Config2__ProductAttributeValue__c prodAttr) {
        system.debug('***ProdAttrCallBack > setDefaults() > prodAttr:' + prodAttr +
';<>; lineItem:' + lineItem);

        // lineItem is a wrapper class to the lineItem class which we use in product
configuration.
        // Take line Item from this wrapper class.
        Apttus_Config2__LineItem__c prodLineItem = lineItem.getLineItemSO();

        system.debug('***ProdAttrCallBack > Product Selected: ' +
prodLineItem.Apttus_Config2__Description__c);

        if(prodLineItem.Apttus_Config2__Description__c == '(Demo) Car') {
            prodAttr.Exterior_Color__c = 'Silver';
            prodAttr.Leather_Color__c = 'Brown';
            prodAttr.Inlay__c = 'Walnut';
            prodAttr.Fuel_Type__c = 'Diesel';
            prodAttr.Fuel_Badge__c = 'No';
        }

        if(prodLineItem.Apttus_Config2__Description__c == 'Internet Service') {
            prodAttr.Internet_Speed__c = '25 Mbps';
        }
    }

    global void finish() {
        system.debug('***ProdAttrCallBack > finish()');
    }
}
```

Related Pricing Callback Class

Related Pricing Callback allows you to apply custom logic to calculate the related pricing. This callback is invoked while CPQ is calculating related pricing after the main pricing is calculated.

To use the Related Pricing Callback you must create a custom Apex class that implements the *Apttus_Config2.CustomClass.IRelatedPricingCallback* interface and register the custom apex class with Related Pricing Callback Class. You must write your custom logic in the custom apex class.

The following methods are available in the
 Apttus_Config2.CustomClass.IRelatedPricingCallback interface:

Method	Signature	Description
computeBasePrice()	*Apttus_Config2.CustomClass.RelatedPriceResult computeBasePrice(Apttus_Config2.ProductConfiguration, Apttus_Config2.LineItem, List)*	You can use this method to apply custom logic for related pricing calculation and override the default CPQ related pricing logic.

If you have defined Pricing Callback class, Related Pricing Callback is always invoked after the Pricing Callback is executed.

Example

In the following sample code, Related Pricing Callback to calculates the price on related line items.

```
global with sharing class APTS_RelatedPricingCallBack implements
 Apttus_Config2.CustomClass.IRelatedPricingCallback {
    /**
        * Callback to compute the related price for the related line item
        * @param cart the cart object associated with the related line item
        * @param relLineItemMO the related line item to compute the base price for
        * @param breakups the list of price breakup sobjects associated with the
related line item
        * @return the related price result
    */
    public Apttus_Config2.CustomClass.RelatedPriceResult
computeBasePrice(Apttus_Config2.ProductConfiguration cart, Apttus_Config2.LineItem
 relLineItemMO, List<Apttus_Config2__PriceBreakup__c> breakups)
```

```
    {
        Apttus_Config2.CustomClass.RelatedPriceResult relPriceResult = new
  Apttus_Config2.CustomClass.RelatedPriceResult();

        //Apttus_Config2__ProductConfiguration__c cartSO =  cart.getConfigSO();
        Apttus_Config2__LineItem__c liSO = relLineItemMO.getLineItemSO();
        system.debug('Net Price-->' + liSO.Apttus_Config2__NetPrice__c + '
ChargeType-->' + liSO.Apttus_Config2__ChargeType__c + ' Base Price-->' +
liSO.Apttus_Config2__BasePrice__c);
        system.debug('breakups-->' + breakups);

        Decimal softwareFeesNetprice = 0;
        if(breakups != null)
        {
            for(Apttus_Config2__PriceBreakup__c b : breakups)
            {
                if(b.Apttus_Config2__BreakupType__c == 'Total' &&
liSO.Apttus_Config2__ChargeType__c == 'Maintenance Fee')
                {
                    softwareFeesNetprice  += b.Apttus_Config2__RelatedNetPrice__c;
                }
            }
        }

        system.debug('softwareFeesNetprice : ' +softwareFeesNetprice+ '
liSO.Maintenance_Level_Value__c ' + liSO.Maintenance_Level_Value__c);
        if(liSO.Apttus_Config2__ChargeType__c == 'Maintenance Fee' &&
liSO.Maintenance_Level_Value__c != null)
        {
            relPriceResult.BasePrice = (softwareFeesNetprice *
liSO.Maintenance_Level_Value__c)/100;
        }
        system.debug('relPriceResult.BasePrice : ' +relPriceResult.BasePrice);
        system.debug('##-- EXIT RELATED PRICING CALLBACK');
        return relPriceResult;
    }
}
```

Asset Line Item Callback Class

Asset Line Item Callback provides you a mechanism to define the criteria based on which you want to filter assets on the Installed Product page. The asset line items on the Installed product page are shown according to the business logic defined in the Asset Line Item Callback class. You can further filter the asset line items on the Installed Products page using the filters on the page. You can also use this callback to validate the termination of an asset on the Asset Termination page.

To use the Asset Line Item Callback you must create a custom Apex class that implements the *Apttus_Config2.CustomClass.IAssetLineItemCallback4* interface and register the custom apex class with Asset Line Callback Class. You must write your custom logic in the custom apex class.

The following methods are available in the
Apttus_Config2.CustomClass.IAssetLineItemCallback4 interface:

Method	Signature	Description
finish()	*void finish()*	This the last method that is executed after the callback is invoked.
getAssetSearchScope()	*List getAssetSearchScope()*	You can use this method to return the asset search scope that was set in the start method.
getFilterExpr()	*String getFilterExpr(Apttus_Config2.CustomClass.ActionParams)*	You can use this method to set the search scope.
getQueryFilter()	*String getQueryFilter(Id)*	You can use this method to create and return the query filter.
start()	*void start(Apttus_Config2.ProductConfiguration, String, List)*	This is the first method that is executed when the callback is invoked.
validateAssetTermination()	*Boolean validateAssetTermination(Set, Set, Date)*	You can use this method to validate the termination of asset.

Method	Signature	Description
getAssetTerminationDate()	*Date getAssetTerminationDate()*	You can use this method to fetch the termination date of asset.

> ⚠ You must update CPQ to the latest version in the Apex class that you create to implement the *Apttus_Config2.CustomClass.IAssetLineItemCallback4* interface. Otherwise, any logic to default date is not executed. Follow the steps below to update the version setting.
> 1. Go to **Setup** > **Develop** > **Apex Classes**.
> 2. Open the Apex class you created to implement the Asset Line Item Callback.
> 3. Click **Edit**. Click **Version Settings**.
> 4. Find the **Conga Configuration & Pricing** package. In the **Version** dropdown next to the package select the latest version of CPQ. For details about latest packages, refer to *CPQ on Salesforce Release Notes.*
> 5. Click **Save**.

Examples

You can use Asset Filter Callback in the following scenarios. You can modify the code based on your requirements.

- Defaulting date and custom validation.
- Retrieving assets from multiple accounts.

The following sample code enables you to set a default date and define custom validation.

```
private Apttus_Config2.CustomClass.IAssetLineItemCallback4 callback = null;
private transient Apttus_Config2.ProductConfiguration cart = null;
private List<String> assetSearchScope = null;
private String assetSearchFilter = null;


/**
 * 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(Apttus_Config2.ProductConfiguration cart, String assetSearchFilter,
List<String> assetSearchScope)
{
    // delegate to the asset callback
    this.cart = cart;
    this.assetSearchFilter = assetSearchFilter;
    this.assetSearchScope = assetSearchScope;
}

global Date getAssetTerminationDate()
{
    return Date.newInstance(2018,01,01);
}

/**
 * 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)
{
    // delegate to the asset callback
    system.debug ('*************** AssetLineItemCallbackAdapter:' + accountId);
    return null;

}

global Boolean validateAssetTermination(Set<ID> assetIds, Set<ID> accountIds, Date
eDate)
{
    String nsPrefix = 'Apttus_Config2__';
    Apttus_Config2.CPQStruct.QueryAssetsRequestDO request = new
 Apttus_Config2.CPQStruct.QueryAssetsRequestDO();
    request.AccountIds = new List<Id>(accountIds);
    system.debug ('accountids' + accountIds);
    request.FieldNames = new List<String>();
    request.FieldNames.add(nsPrefix + 'StartDate__c');
    request.FieldNames.add(nsPrefix + 'EndDate__c');
    request.FieldNames.add( 'Name');
```

```
    //added new fields
    request.FieldNames.add(nsPrefix + 'BusinessObjectId__c');
    request.FieldNames.add(nsPrefix + 'ProductId__c');
    request.FieldNames.add(nsPrefix + 'LineNumber__c');
    request.FieldNames.add(nsPrefix + 'PrimaryLineNumber__c');
    request.FieldNames.add(nsPrefix + 'IsPrimaryRampLine__c');
    request.FieldNames.add(nsPrefix + 'IsPrimaryLine__c');
    request.FieldNames.add(nsPrefix +  'PriceGroup__c');
    request.FieldNames.add(nsPrefix +  'LineType__c');
    request.FieldNames.add(nsPrefix +  'ChargeType__c');
    request.FieldNames.add(nsPrefix + 'CurrentContractStartDate__c');

    if (assetIds != null && assetIds.size() >0)
    {
        request.customFilter =  'Id IN  (';
        for (Id assetId : assetIds)
        {
            request.customFilter +=  '\'' + assetId + '\',';
            system.debug ('assetids: ' + assetId);
        }
        request.customFilter = request.customFilter.removeEnd(',');
        request.customFilter += ')';
        system.debug ('request.customFilter ' + request.customFilter);
    }

    Apttus_Config2.CPQStruct.QueryAssetsResponseDO assetResponseDO =
Apttus_Config2.AssetService.getAssetLineItems(request);

    for (Apttus_Config2__AssetLineItem__c asset: assetResponseDO.AssetLineItems)
    {
        if (asset.Apttus_Config2__IsPrimaryRampLine__c &&
asset.Apttus_Config2__IsPrimaryLine__c)
        {
            Apttus_Config2__AssetLineItem__c farthestLineItem =
findFarthestRampLine(asset, assetResponseDO.AssetLineItems);
            if (eDate > farthestLineItem.Apttus_Config2__EndDate__c)
            {
                ApexPages.Message message = new
 ApexPages.Message(ApexPages.Severity.ERROR, 'Termination date : '+eDate + 'should
not be greater than Ramp Farthest Asset End date: ' +
farthestLineItem.Apttus_Config2__EndDate__c );
                ApexPages.addMessage(message);
```

```
                    return false;

                }
            }
        else if (asset.Apttus_Config2__IsPrimaryLine__c &&
asset.Apttus_Config2__LineType__c == 'Product/Service')
            {
                if (eDate >= asset.Apttus_Config2__EndDate__c)
                {
                    ApexPages.Message message = new
 ApexPages.Message(ApexPages.Severity.ERROR, 'Termination date should not greater
than Asset End Date :Effective Date is ' + eDate );
                    ApexPages.addMessage(message);
                    return false;

                }
                else if (eDate < asset.Apttus_Config2__StartDate__c)
                {
                    ApexPages.Message message = new
 ApexPages.Message(ApexPages.Severity.ERROR, 'Invalid Termination date');
                    ApexPages.addMessage(message);
                    return false;
                }
            }

        /* if (eDate < asset.Apttus_Config2__CurrentContractStartDate__c) {
                ApexPages.Message message = new
ApexPages.Message(ApexPages.Severity.ERROR, 'Asset cannot be terminated before
Contract start date');
                ApexPages.addMessage(message);
                return false;
        } */
    }
    return true;
}


/**
 * 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(Apttus_Config2.CustomClass.ActionParams params)
```

```
{
    // dispatch based on the location id parameter
    String clause = '';
    Boolean locationFound = false;
    Boolean accountFound = false;
    Boolean customParamsExist = false;
    String ns = 'Apttus_Config2__';
    List<String> fieldNames = new List<String>();
    fieldNames.add(ns + 'AccountId__c');

    if(params.CustomParams != null && params.CustomParams.size() > 0)
    {
        customParamsExist = true;
        Boolean isFirst = true;
        for(String fieldName: params.CustomParams.keySet())
        {
            if (isFirst)
            {
                isFirst = false;
            }
            else
            {
                clause += ' AND ';
            }

            if (fieldName.toLowerCase().contains('fromdate'))
            {
                clause += fieldName.substring(0, fieldName.toLowerCase().indexOf('fro
mdate')) + ' >= ' + params.CustomParams.get(fieldName);

            }
            else if (fieldName.toLowerCase().contains('todate'))
            {
                clause += fieldName.substring(0, fieldName.toLowerCase().indexOf('tod
ate')) + ' <= ' + params.CustomParams.get(fieldName);
            }
            else
            {
                clause += fieldName + ' IN (' + params.CustomParams.get(fieldName) +
')';
            }
        }
    }
```

```
if(params.LocationIds != null && !params.LocationIds.isEmpty())
{
    if (customParamsExist)
    {
        clause += ' AND ';
    }
    Boolean isFirst = true;
    for (ID Id : params.LocationIds)
    {
        if(isFirst)
        {
            isFirst = false;
            locationFound = true;
            clause += ns + 'LocationId__c IN (\'' + Id;
        }
        else
        {
            clause += '\', \'' + Id;
        }
    }
    if (locationFound)
    {
        clause += '\')';
    }
}

if(params.AccountIds != null && !params.AccountIds.isEmpty())
{
    Boolean isFirstSource = true;
    for (String accountAPIName : fieldNames)
    {
        if (isFirstSource)
        {
            isFirstSource = false;
        }
        else
        {
            clause += ' OR ';
        }
        Boolean isFirst = true;
        for (ID Id : params.AccountIds)
        {
            if(isFirst)
            {
                if (locationFound || customParamsExist)
```

```
                  {
                      clause += ' AND ';
                  }
                  isFirst = false;
                  accountFound = true;
                  clause += accountAPIName + ' IN (\'' + Id;
              }
              else
              {
                  clause += '\', \'' + Id;
              }
          }
          if (accountFound)
          {
              clause += '\')';
          }
      }
  }
  boolean isShowMultipleAccountsEnabled = false;
  //when account filter is not displayed and static asset filter is not present
user cart account id

  if (!accountFound && !isShowMultipleAccountsEnabled && (assetSearchFilter == null
|| assetSearchFilter.trim().length() == 0) )
  {
      if (locationFound || customParamsExist)
      {
          clause += ' AND ';
      }
      accountFound = true;
      Boolean isFirstSource = true;
      for (String accountAPIName : fieldNames)
      {
          if (isFirstSource)
          {
              isFirstSource = false;
              clause += accountAPIName + ' = \'' + params.AccountId + '\' ';
          }
          else
          {
              clause += ' OR ' + accountAPIName + ' = \'' + params.AccountId + '\'
';
          }
```

```
            }
        }
    //append static asset filter
    if (locationFound == true || accountFound == true || customParamsExist == true)
    {
        if (!(assetSearchFilter == null || assetSearchFilter.trim().length() == 0))
        {
            return clause  +  ' AND '  + assetSearchFilter;
        }
        return clause ;
    }
    // account id is also available in the cart
    if (!(assetSearchFilter == null || assetSearchFilter.trim().length() == 0))
    {
        return assetSearchFilter;
    }

    // the sample query filter
    String qryStr = '';
    Boolean isFirstSource = true;
    for (String accountAPIName : fieldNames)
    {
        if (isFirstSource)
        {
            isFirstSource = false;
            qryStr += accountAPIName + ' = \'' + params.accountId + '\' ';
        }
        else
        {
            qryStr += ' OR ' + accountAPIName + ' = \'' + params.accountId + '\' ';
        }
    }
    system.debug ('*************** qryStr'+ qryStr);
    return qryStr;
}

/**
 * Gets the asset search scope
 * @return thr asset search scope or null to use the default asset search scope
 */
global List<String> getAssetSearchScope()
{
    // delegate to the asset callback
    return null;
}
```

```
/**
 * Callback after the filter is used
 * Use the finish method to release state
 */
global void finish()
{
    // delegate to the asset callback
}

private static Apttus_Config2__AssetLineItem__c
findFarthestRampLine(Apttus_Config2__AssetLineItem__c primaryRampLine,
List<Apttus_Config2__AssetLineItem__c> assetLineItems)
{
    Apttus_Config2__AssetLineItem__c farthestAssetLineItem = primaryRampLine;
    if (assetLineItems != null && !assetLineItems.isEmpty()) {
        for (Apttus_Config2__AssetLineItem__c assetLineItem : assetLineItems)
        {
        if (assetLineItem.Apttus_Config2__PriceGroup__c == 'Price Ramp'
            && assetLineItem.Apttus_Config2__BusinessObjectId__c ==
primaryRampLine.Apttus_Config2__BusinessObjectId__c
            && assetLineItem.Apttus_Config2__ProductId__c ==
primaryRampLine.Apttus_Config2__ProductId__c
            && assetLineItem.Apttus_Config2__LineNumber__c.intValue() ==
primaryRampLine.Apttus_Config2__LineNumber__c.intValue()
            && (primaryRampLine.Apttus_Config2__PrimaryLineNumber__c == null
                || (assetLineItem.Apttus_Config2__PrimaryLineNumber__c != null
                    && assetLineItem.Apttus_Config2__PrimaryLineNumber__c.intValue()
== primaryRampLine.Apttus_Config2__PrimaryLineNumber__c.intValue()))
            && assetLineItem.Apttus_Config2__IsPrimaryRampLine__c == false
            && assetLineItem.Apttus_Config2__EndDate__c >
farthestAssetLineItem.Apttus_Config2__EndDate__c)
            {
                farthestAssetLineItem = assetLineItem;
            }
        }
    }
    return farthestAssetLineItem;
}
```

The following sample code enables you to fetch assets from multiple accounts.

```
/*
 * This class is used in Asset Line Item callback to show asset on Cart page.
 */
global with sharing class CurrentUserFilter_AssetLineItemCallback implements
 Apttus_Config2.CustomClass.IAssetLineItemCallback4
    {

    private List<String> assetSearchScope = null;
    private String assetSearchFilter = null;
    private ID userId = null;


    /**
     * 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(Apttus_Config2.ProductConfiguration cart, String
assetSearchFilter, List<String> assetSearchScope) {
        this.userId = cart.getConfigSO().CreatedById;
    }


    /**
     * 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) {
        // all Asset Lines created by the current user
        return 'CreatedById = \'' + userId + '\' AND CreatedDate > LAST_MONTH';
    }


    global Boolean validateAssetTermination(Set<ID> assetIds, Set<ID> accountIds,
Date eDate) {
        return true;
    }


    global Date getAssetTerminationDate() {
```

```
        return Date.newInstance(2018,01,01);
    }

    /**
     * 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(Apttus_Config2.CustomClass.ActionParams params) {
        return getQueryFilter(null);
    }

    /**
     * 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() {

    }
}
```

Asset Renewal Custom Callback Class

Asset Renewal Custom Callback allows you to filter the renewal quote assets that are created using the On-Demand job. For example, you can filter assets by **Product Name** as a filter criteria. You can also use this callback to populate a selective set of parameters with corresponding values on a renewal quote (with the same data as the original quote) during renewal automation. If the user has data on the quote header that must be propagated to line items and then be used in pricing, this callback can copy those fields to the renewal quote and pricing will be accurate on the renewal quote generated.

To use the Asset Renewal Custom Callback, you must create a custom Apex class that implements the *Apttus_Config2.CustomClass.IAssetRenewalCustomCallback* interface and register the custom apex class with Asset Renewal Custom Callback Class. You must write your custom logic in the custom apex class.

The following methods are available in the *Apttus_Config2.CustomClass.IAssetRenewalCustomCallback* interface:

Method	Signature	Description
getAssetFilterExpr()	*String getAssetFilterExpr()*	You must use this method to append custom filters to the on-demand asset query when the asset line items are fetched from the on-demand renewal quote. If there are no additional filters (other than the standard filters), return null.
getConfigProperties()	*List getConfigProperties(SObject)*	You must use this method to fetch the configuration request parameters when the renewal cart is created. You can also use this method to update the proposal or agreement with default values by passing a proposal or agreement subject.

Example

In the following sample, the assets on the renewal quote are filtered based on the product name.

```
global with sharing class SampleAssetRenewalCustomCallback 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 = 'Standalone2'];
        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', 'LAngFlow'));
        configProps.add(new Apttus_Config2.Property('useAdvancedApproval', 'false'));
        configProps.add(new Apttus_Config2.Property('useDealOptimizer', 'false'));
        return configProps;
    }
}
```

Revalidation Callback Class

Revalidation Callback provides you a mechanism to implement custom revalidations for products based on custom criteria. You can define custom criteria based on which the products are identified by CPQ for revalidation. The callback allows you to define a custom logic to be applied on the line item along with revalidation. You can also define a custom product version when the line item is created.

To use the Revalidation Callback you must create a custom Apex class that implements the *Apttus_Config2.CustomClass.IRevalidationCallback* interface and register the custom apex class with Revalidation Callback Class. You must write your custom logic in the custom apex class.

The following methods are available in the *Apttus_Config2.CustomClass.IRevalidationCallback* interface:

Method	Signature	Description
getRevalidations()	*Apttus_Config2.CustomClass.RevalidationResponse getRevalidations(Apttus_Config2. CustomClass.RevalidationRequest)*	You can use this method to determine whether the line item requires revalidation based on a custom logic that you can define. This method is invoked when you reconfigure a save or finalized cart.
setRevalidations()	*void setRevalidations(Apttus_Config2.C ustomClass.RevalidationRequest)*	You can use this method write custom logic to apply on line items. This method is invoked when you click **Apply Revalidation** on the Revalidation pop-up.
setVersionNumber()	*void setVersionNumber(Id, List)*	You can use this method to set the version number field of the line item with custom value based on custom logic you define. This method is invoked when you add line items to the cart.

Example

You can use this callback to a scenario like:

- You can have multiple versions for a single product record. This enables you to revalidate a specific version of the product as necessary.
  For example, for product record, you can set the version as X, Y, and any such value for the newly created line item of that product. You can determine which line item qualifies for revalidation and apply custom logic on the line items accordingly.
- You can apply validation based custom fields other than the **Product Version**.
- You can validate changes in pricing fields like **List price** of a given price list item.

The below-given code example checks if the quantity of line items is greater then 10 then triggers the revalidation on the cart, revalidates the line item and resets the quantity of the line items.

```
/**
 *  Apttus Config & Pricing
 *  SampleRevalidationCallback
 *
 *  @2019-2020 Apttus Inc. All rights reserved.
 */
```

```
global with sharing class SampleRevalidationCallback implements
 Apttus_Config2.CustomClass.IRevalidationCallback {
    private static String QUANTITY_TOO_HIGH = 'Quantity Too High';
    /**
     * Callback at the beginning of revalidation check.
     * use this method to get custom revalidations
     * @param request contains lineItemSO and validation info
     * @return map of revalidation category against list of line numbers
     */
    global Apttus_Config2.CustomClass.RevalidationResponse
getRevalidations(Apttus_Config2.CustomClass.RevalidationRequest request){
        Apttus_Config2.CustomClass.RevalidationResponse response = new
 Apttus_Config2.CustomClass.RevalidationResponse();
        Map<String, List<Apttus_Config2__LineItem__c>> lineItemSOsByRevalCategory =
new Map<String, List<Apttus_Config2__LineItem__c>>();

        for (String revalCat : request.lineItemSOsByRevalCategory.keySet()) {
            lineItemSOsByRevalCategory.put(revalCat,
request.lineItemSOsByRevalCategory.get(revalCat));
            for (Apttus_Config2__LineItem__c lineItemSO :
request.lineItemSOsByRevalCategory.get(revalCat)) {
                response.isRevalidationNeeded = true;
            }
        }

        response.lineItemSOsByRevalCategory = lineItemSOsByRevalCategory;
        response.isHardRevalidationNeeded = false;
        List<Apttus_Config2__LineItem__c> customQuantityLines = new
 List<Apttus_Config2__LineItem__c>();

        // Implementing list price change use case
        for (Apttus_Config2__LineItem__c lineItemSO : [select Id,
Apttus_Config2__Quantity__c,
Apttus_Config2__PriceListItemId__r.Apttus_Config2__ListPrice__c,
Apttus_Config2__PrimaryLineNumber__c from Apttus_Config2__LineItem__c where
Apttus_Config2__ConfigurationId__c = :request.cartId]) {
            if(lineItemSO.Apttus_Config2__Quantity__c > 10) {
                customQuantityLines.add(lineItemSO);
            }
        }
        if (customQuantityLines.size() > 0) {
            response.lineItemSOsByRevalCategory.put(QUANTITY_TOO_HIGH,
customQuantityLines);
```

```
                    response.isRevalidationNeeded = true;
        }
        return response;
    }


    /**
     * Callback for applying revalidation
     * @param  cartId Id of the cart to revalidate
     * @param  lineNumbers the list of line numbers to be validated
     */
    global void setRevalidations(Apttus_Config2.CustomClass.RevalidationRequest
request) {
        List<Apttus_Config2__LineItem__c> lineItemsSOsToUpdate = new
 List<Apttus_Config2__LineItem__c>();
        for (String revalCategory : request.lineItemSOsByRevalCategory.keySet()) {
            for (Apttus_Config2__LineItem__c lineItemSO :
request.lineItemSOsByRevalCategory.get(revalCategory)) {

                if (revalCategory == QUANTITY_TOO_HIGH) {
                    lineItemSO.Apttus_Config2__Quantity__c = 1.0;
                    lineItemSO.Apttus_Config2__PricingStatus__c = 'Pending';
                    lineItemsSOsToUpdate.add(lineItemSO);
                }
            }
        }
        System.debug('**'+lineItemsSOsToUpdate);
        if (lineItemsSOsToUpdate.size() > 0) {
            System.debug('>>'+lineItemsSOsToUpdate);
            update lineItemsSOsToUpdate;
        }
    }


    /**
     * Callback to set custom version number
     * @param  cartId Id of the cart
     * @param  lineIntemSOs the list of line Item sobjects to set custom version
     */
    global void setVersionNumber(ID cartId, List<Apttus_Config2__LineItem__c>
lineItemSOs){
        /*for (Apttus_Config2__LineItem__c lineItemSO : lineItemSOs) {
            lineItemSO.Apttus_Config2__ProductVersion__c = 33.0;
        }*/
    }
}
```

Action Params Callback Class

Action Params Callback provides you a mechanism to create action parameters for a cart object. You can use the action parameters created using the callback in the following actions in CPQ:

- Submitting the cart for approval
- Generating document
- Analyzing the deal
- Retrieving Advanced Approval URL for selected line item

To use the Action Params Callback you must create a custom Apex class that implements the *Apttus_Config2.CustomClass.IActionParamsCallback* interface and register the custom apex class with Action Params Callback Class. You must write your custom logic in the custom apex class.

The following methods are available in the *Apttus_Config2.CustomClass.IActionParamsCallback* interface.

Method	Signature	Description
createActionParams	*Apttus_Config2.CustomClass.Action Params createActionParams(Apttus_Config 2.ProductConfiguration)*	You can use this method to create action parameters for the cart object. This method is invoked when you: • Submit the cart for approval • Retrieve Advanced Approval URL for selected line item • Perform the analyze deal action • Perform the generate doc action

The following table lists the response parameters of the *Apttus_Config2.CustomClass.IActionParamsCallback.*

Response Parameters:Apttus_Config2.CustomClass.ActionParams		
Name	**Type**	**Description**
AccountId	Id	The ID of the Account.

Response Parameters:Apttus_Config2.CustomClass.ActionParams		
Name	**Type**	**Description**
AccountIds	Set<Id>	The set of Account ID you want to use to filter the result. CPQ ignores the *AccountId* field if this field is used.
ActionName	String	The name of the action.
ActivateOrder	Boolean	Indicates whether to activate the order or not.
ApprovalCtxType	String	The Context Type of Approval.
ApprovalMode	String	The Mode of Approval
ApprovalReason	String	The reason of Approval.
ApprovalType	String	The type of Approval.
AssetLineItems	List<Apttus_Config2 __AssetLineItem__c>	The list of asset line items
BundleId	Id	The IDs of the bundle product.
CartIds	List<IDs>	The list of cart IDs.
CartSyncMode	Apttus_Config2.Cus tomClass.SyncMode	The Sync Mode of the cart. For example, *enum SyncMode {SYNC, FINALIZE}.*
ConfigurationId	Id	The ID of the configuration in the proposal.
CurrentState	String	The current state
CustomParams	Map<String, String>	The custom parameter that you have defined.
FinalizeClass	String	The callback class that is executed when you finalize the cart.

Response Parameters:Apttus_Config2.CustomClass.ActionParams		
Name	Type	Description
Flow	String	The flow used in the configuration
IsAngular	Boolean	Indicates whether the user interface is Angular.
IsDraft	Boolean	Indicated whether the proposal is in draft stage.
LaunchState	String	The launch state.
LineItemIds	Set<Id>	The set of line item IDs.
LineNumber	Decimal	The line number of the products.
LocationIds	Set<Id>	The set of location IDs.
Method	String	The method of configuration.
Mode	String	The mode of the configuration.
OrderLineItems	List<Apttus_Config2__OrderLineItem__c>	The list of order line items.
OriginalOrderSO	Apttus_Config2__Order__c	The Original Order SObject
OutputFormat	String	The output format of the generated document. You can use the following values: • _PDF_ • _DOC_ • _RTF_
ProductIDs	List<IDs>	The list of product IDs.

Response Parameters:Apttus_Config2.CustomClass.ActionParams		
Name	Type	Description
ProtectionLevel	String	The protection level for the parameters you created.
RequestId	Id	The request ID of the configuration.
ReturnId	Id	The record ID of the business object.
ReturnPage	String	The return page of the business object.
SessionId	String	The ID of the session.
SessionUrl	String	The URL of the session.
TemplateName	String	The name of the template in the proposal.

Example

The following sample code enables you to create parameters.

```
global with sharing class DefaultActionParamsCallback implements
 Apttus_Config2.CustomClass.IActionParamsCallback
{
    /**
     * Callback to create action parameters for the given cart object
     * @param cart the cart object
     * @return the action parameters object
     */
    global Apttus_Config2.CustomClass.ActionParams
createActionParams(Apttus_Config2.ProductConfiguration cart)
    {

        // create a test configuration request
        Apttus_Config2.TempObject__c requestSO = new
 ConfigRequestTest().createTempObject('Quote');

        // action parameters
        Apttus_Config2.CustomClass.ActionParams params = new
 Apttus_Config2.CustomClass.ActionParams();
```

```
        params.ActionName = 'CustomAction';
        params.ApprovalType = Apttus_Config2.CustomClass.APPROVALTYPE_STANDARD;
        params.ApprovalCtxType = Apttus_Config2.CustomClass.APPROVALCTX_HEADER;
        params.ApprovalMode = Apttus_Config2.CustomClass.APPROVALMODE_PREVIEW;
        params.ApprovalReason = 'Request for Approval';
        params.RequestId = requestSO.Id;
        params.Method = null;
        params.Mode = null;
        params.Flow = 'Default';
        params.LineNumber = 1;
        params.TemplateName = 'Test Template';
        params.ProtectionLevel = 'Read only';
        params.OutputFormat = 'PDF';
        params.IsDraft = true;
        params.SessionId = UserInfo.getSessionId();
        params.SessionUrl = 'https://login.salesforce.com'; // Use https://
  test.salesforce.com for sandbox.
        params.ReturnId = '01tA0000001qEcJ'; // The id of the bussiness object record
        params.ReturnPage = 'MyCustomPage';
        params.CustomParams.put('CartId', cart.getId());

        return params;


    }
}
```

Action Invoker Callback Class

Action Invoker Callback provides you a mechanism to implement custom navigation for custom actions. You can define logic for a custom page to redirect the Sales rep to.

To use the Action Invoker Callback you must create a custom Apex class that implements the *Apttus_Config2.CustomClass.IActionInvokerCallback* interface and register the custom apex class with Action Invoker Callback Class. You must write your custom logic in the custom apex class.

The following methods are available in the *Apttus_Config2.CustomClass.IActionInvokerCallback* interface.

Method	Signature	Description
invokeAction	*System.PageReference invokeAction(Apttus_Config2.Product Configuration, Apttus_Config2.CustomClass.ActionParams)*	You can use this method to execute the custom navigation upon the Sales rep's click of the custom action.

Example

The following sample code is a skeletal structure of custom callback class that implements the *IActionInvokerCallback* interface. You can write your own logic and return navigation for the custom action defined in Display Action Setting for the specific flow.

```
global with sharing class DefaultActionInvokerCallback implements
 Apttus_Config2.CustomClass.IActionInvokerCallback {
   /**
    * Callback to invoke the custom action
    * @param cart the cart object
    * @param params the parameters for the action
    * @return the result page reference or null if there is no page
    */
    global ApexPages.PageReference invokeAction(ProductConfiguration cart,
Apttus_Config2.CustomClass.ActionParams params) {
        // do nothing
        return null;
      }
 }
```

Action Callback Class

Action Callback provides you a mechanism to apply custom actions on different actions available on the cart. For example, you can apply custom action before deleting or copying a bundle, finalizing a cart, submitting a cart for approval. You can directly use methods of the callback to apply the custom action or use Web Services to implement the callback.

To use the Action Callback you must create a custom Apex class that implements one of the following interfaces and you must register the custom class with Action Callback Class. You must write your custom logic in the custom apex class.

Interface	Description
IActionCallback2	This interface provides a mechanism to implement custom actions on cart finalization, submission for approval using Web Services.
IActionCallback3	This interface provides a mechanism to implement custom actions before delete and copy actions on bundles.

This section describes the interfaces of Action Callback:

- IActionCallback2 Interface
- IActionCallback3 Interface

IActionCallback2 Interface

IActionCallback2 class provides a mechanism to apply custom actions on cart finalization, submission for approval using the below Web Services.

- *ActionInvokerWebService.invokeAfterFinalizeCart*
- *ActionInvokerWebService.invokeSyncCart*
- *ActionInvokerWebService.invokeSyncCartAsync*
- *ActionInvokerWebService.invokeFinalizeCart*
- *ActionInvokerWebService.invokeFinalizeCartAsync*
- *ActionInvokerWebService.invokeAfterSyncCart*
- *ActionInvokerWebService.invokeAfterSyncCartAsync*
- *ActionInvokerWebService.invokeAfterFinalizeCartAsync*
- *ActionInvokerWebService.invokeSubmitApproval*
- *ActionInvokerWebService.invokeGenerateDoc*

Action callback executes the custom logic from Web Services on the following aspects of CPQ:

- Finalizing the cart
- Synchronizing the cart
- Submitting the cart for approval
- Generating documents

The following methods are available in the *Apttus_Config2.CustomClass.IActionCallback2* interface:

Method	Signature	Description
afterFinalizeCart()	*Boolean afterFinalizeCart(Id)*	You can use this method to perform post-finalization tasks on the cart. This method is invoked when you execute the following Web Services: • *ActionInvokerWebService.invokeAfter FinalizeCart()* • *ActionInvokerWebService.invokeAfter FinalizeCartAsync()*
finalizeCart()	*Boolean finalizeCart(Id)*	You can use this method to finalize the cart. This method is invoked when you execute the following Web Services: • *ActionInvokerWebService.invokeFinal izeCart()*
generateDoc()	*Id generateDoc(Id, Apttus_Config2.CustomClass.A ctionParams)*	You can use this method to generate documents for the given cart. This method is invoked when you generate documented the following Web Service: • *ActionInvokerWebService.invokeGene rateDoc()*
submitApproval()	*Boolean submitApproval(Id, Apttus_Config2.CustomClass.A ctionParams)*	You can use this method to submit the cart for approval. This method is invoked when you submit the cart for approval using following the Web Service: • *ActionInvokerWebService.invokeSub mitApproval()*

Method	Signature	Description
afterSyncCart()	*Boolean afterSyncCart(Id)*	You can use this method to perform post-synchronization tasks on the cart. This method is invoked when the cart is synchronized using the below Web Services: • *ActionInvokerWebService.invokeSyncCart()* • *ActionInvokerWebService.invokeAfterSyncCartAsync()*
syncCart()	*Boolean syncCart(Id)*	You can use this method to perform to synchronize the cart. This method is invoked when the cart is synchronized synchronously or asynchronously using the below Web Services. • *ActionInvokerWebService.invokeSyncCart()* • *ActionInvokerWebService.invokeSyncCartAsync()*

The following table lists the response parameters of the *Apttus_Config2.CustomClass.IActionParamsCallback.*

Response Parameters:Apttus_Config2.CustomClass.ActionParams		
Name	**Type**	**Description**
AccountId	Id	The ID of the Account.
AccountIds	Set<Id>	The set of Account ID you want to use to filter the result. CPQ ignores the *AccountId* field if this field is used.
ActionName	String	The name of the action.
ActivateOrder	Boolean	Indicates whether to activate the order or not.

Response Parameters:Apttus_Config2.CustomClass.ActionParams		
Name	Type	Description
ApprovalCtxType	String	The Context Type of Approval.
ApprovalMode	String	The Mode of Approval
ApprovalReason	String	The reason of Approval.
ApprovalType	String	The type of Approval.
AssetLineItems	List<Apttus_Config2__AssetLineItem__c>	The list of asset line items
BundleId	Id	The IDs of the bundle product.
CartIds	List<IDs>	The list of cart IDs.
CartSyncMode	Apttus_Config2.CustomClass.SyncMode	The Sync Mode of the cart. For example, *enum SyncMode {SYNC, FINALIZE}*.
ConfigurationId	Id	The ID of the configuration in the proposal.
CurrentState	String	The current state
CustomParams	Map<String, String>	The custom parameter that you have defined.
FinalizeClass	String	The callback class that is executed when you finalize the cart.
Flow	String	The flow used in the configuration
IsAngular	Boolean	Indicates whether the user interface is Angular.
IsDraft	Boolean	Indicated whether the proposal is in draft stage.
LaunchState	String	The launch state.

Response Parameters:Apttus_Config2.CustomClass.ActionParams		
Name	Type	Description
LineItemIds	Set<Id>	The set of line item IDs.
LineNumber	Decimal	The line number of the products.
LocationIds	Set<Id>	The set of location IDs.
Method	String	The method of configuration.
Mode	String	The mode of the configuration.
OrderLineItems	List<Apttus_Config2 __OrderLineItem__c>	The list of order line items.
OriginalOrderSO	Apttus_Config2__Or der__c	The Original Order SObject
OutputFormat	String	The output format of the generated document. You can use the following values: • *PDF* • *DOC* • *RTF*
ProductIDs	List<IDs>	The list of product IDs.
ProtectionLevel	String	The protection level for the parameters you created.
RequestId	Id	The request ID of the configuration.
ReturnId	Id	The record ID of the business object.
ReturnPage	String	The return page of the business object.
SessionId	String	The ID of the session.

Response Parameters:Apttus_Config2.CustomClass.ActionParams		
Name	Type	Description
SessionUrl	String	The URL of the session.
TemplateName	String	The name of the template in the proposal.

Example

The sample code below enables you to send out the status of the cart to the proposal owner as callback action after finalizing the cart. You can perform post-finalization by calling the Web Service *Apttus_Config2. ActionInvokerWebService.invokeAfterFinalizeCart()* passing the custom callback class name and the cart ID as parameters.

```
1   global with sharing class Apttus_ActionCallback implements
     Apttus_Config2.CustomClass.IActionCallback2
2   {
3       global static Boolean finalizeCart(ID cartId)
4       {
5           // callback logic to finalize the cart.  This logic will be
    executed when we call the webservice
    Apttus_Config2.ActionInvokerWebService.invokeFinalizeCart
6           return true;
7       }
8
9       global static Boolean afterFinalizeCart(ID cartId)
10      {
11          // get the owner of the proposal
12          Apttus_Config2__ProductConfiguration__c proposalSO = [SELECT
    Name,Apttus_Config2__Proposald__r.owner.id
13                                                  FROM
    Apttus_Config2__ProductConfiguration__c
14                                                  WHERE Id
    =:cartId LIMIT 1];
15          // get the email ID of the proposal owner
16          Id ownerID = proposalSO.Apttus_Config2__Proposald__r.owner.id;
17          User userSO = [SELECT email FROM User WHERE Id =: ownerID LIMIT 1]
    ;
18
19          // send finalize action
20          Messaging.SingleEmailMessage mail = new
      Messaging.SingleEmailMessage();
```

```
21          mail.setSenderDisplayName('Salesforce Support'); //Sender's
        Display name
22          mail.setSaveAsActivity(true);
23          mail.setSubject('Cart - ' + proposalSO.Name + ' Finalized');
24          mail.setPlainTextBody('Please be informed that the cart is
        finalized.');
25          mail.setToAddresses(new List <String> {userSO.email});
26          Messaging.sendEmail(new Messaging.SingleEmailmessage[] {mail});
27
28          return true;
29
30      }
31
32      global static Boolean submitApproval(ID cartId,
        Apttus_Config2.CustomClass.ActionParams params)
33      {
34          // callback logic to submit the cart for approval. This method
        will be executed when we call the WebService
        Apttus_Config2.ActionInvokerWebService.invokeSubmitApproval
35          return true;
36      }
37
38      global static ID generateDoc(ID cartId,
        Apttus_Config2.CustomClass.ActionParams params)
39      {
40          // callback logic to generate the document for the given the cart.
        This method will be executed when we call the webserice
        Apttus_Config2.ActionInvokerWebService.invokeGenerateDoc()
41          return null;
42      }
43
44      global static Boolean syncCart(ID cartId)
45      {
46          // callback logic to synchronize the cart for the given cartId.
        This method will be executed when we call the webserice
        Apttus_Config2.ActionInvokerWebService.invokeSyncCart or
        Apttus_Config2.ActionInvokerWebService.invokeSyncCartAsync
47          return true;
48      }
49      global static Boolean afterSyncCart(ID cartId)
50      {
51          // callback logic to perform post-synchronization tasks for the
        given cartId. This method will be executed when we call the webserice
```

```
        Apttus_Config2.ActionInvokerWebService.invokeAfterSyncCart or
        Apttus_Config2.ActionInvokerWebService.invokeAfterSyncCartAsync
52          return true;
53      }
54  }
```

IActionCallback3 Interface

IActionCallback3 class provides a mechanism to apply custom actions before delete and copy actions on bundles.

The following methods are available in the *Apttus_Config2.CustomClass.IActionCallback3* interface:

Method	Signature	Description
beforeCopyBundleLineItems()	*Apttus_Config2.CustomClass.ActionCallbackResponse beforeCopyBundleLineItems(Apttus_Config2.CustomClass.ActionCallbackRequest)*	You can use this method to perform tasks to be executed before a bundle is copied. This method is invoked when the Sales Rep Clones bundle or standalone product.
beforeDeleteBundleLineItems()	*Apttus_Config2.CustomClass.ActionCallbackResponse beforeDeleteBundleLineItems(Apttus_Config2.CustomClass.ActionCallbackRequest)*	You can use this method to perform tasks to be executed before performing the following actions: • Abandoning the configuration • creating, updating, and deleting line items

Request Parameters: Apttus_Config2.CustomClass.ActionCallbackRequest

Name	Type	Description
BundleLineItemIds	List	The list IDs of line items.
CartId	Id	The ID of the configuration

Response Parameters: Apttus_Config2.CustomClass.ActionCallbackResponse		
Name	Type	Description
errorMessages	List<String>	List of error messages if they occur.
IsSuccess	Boolean	Indicates whether the action was successful.

Example

The sample code below allows you to use **beforeDeleteBundleLineItems()** to validate the line items based on custom logic written in the method. You can allow the deletion of a line only when the following conditions are fulfilled:

- The number of line items to be deleted is not more than the limit defined in the code.
- The line items are not read-only and any approval request is not pending.
- The line items are not in collaboration.

```
global with sharing class APTS_ActionCallback3 implements
 Apttus_Config2.CustomClass.IActionCallback3
{
    private static final String strEMPTY = '';
    public static final Integer MAX_LI_TO_DELETE = 2;
    public static final String BUSINESS_OBJECT_TYPE_PROPOSAL = 'Proposal';
    public static final String PRODCONFIG_STATUS_PENDING_APPROVAL =
'Pending Approval';
    public static final String COLLAB_STATUS_COMPLETED = 'Completed';
    public static final String COLLAB_STATUS_ACCEPTED = 'Accepted';
    public static final String DEL_LIMIT_EXCEED_ERROR_MESSAGE = 'Deletion
of more than \''+

MAX_LI_TO_DELETE +
                                                          '\' line
items not supported. Please select fewer line items to delete';

    public static final String PARENT_UC_ERROR_MESSAGE = 'Parent cart
under collaboration, can not delete';

    global static Apttus_Config2.CustomClass.ActionCallbackResponse
beforeDeleteBundleLineItems(Apttus_Config2.CustomClass.ActionCallbackReque
st CallBackReq)
    {
```

```
17          Apttus_Config2.CustomClass.ActionCallbackResponse result = new
    Apttus_Config2.CustomClass.ActionCallbackResponse();
18          List<Id> lstLIToDel = CallBackReq.BundleLineItemIds;
19
20          // check the deletion limit.
21          if(lstLIToDel != null && lstLIToDel.size() > MAX_LI_TO_DELETE)
22          {
23              result.isSuccess= false;
24              result.errorMessages.add(DEL_LIMIT_EXCEED_ERROR_MESSAGE);
25              System.debug(LoggingLevel.ERROR,
    DEL_LIMIT_EXCEED_ERROR_MESSAGE);
26              return result;
27          }
28
29          // Check for collaboration status.
30          String collabRequestStatus;
31          Boolean isCollabComplete;
32
33          for (Apttus_Config2__LineItem__c oLI : [
34              SELECT
    Apttus_Config2__CollaborationRequestId__r.Apttus_Config2__Status__c,
35                  Apttus_Config2__CollaborationRequestId__c,
36
    Apttus_Config2__ConfigurationId__r.Apttus_Config2__Status__c,
37
    Apttus_Config2__ConfigurationId__r.Apttus_Config2__BusinessObjectType__c,
38                  Apttus_Config2__IsReadOnly__c
39              FROM Apttus_Config2__LineItem__c
40              WHERE Id IN :lstLIToDel]) {
41
42          collabRequestStatus =
    (oLI.Apttus_Config2__CollaborationRequestId__c != NULL) ?
43
    oLI.Apttus_Config2__CollaborationRequestId__r.Apttus_Config2__Status__c :
44                              strEMPTY ;
45          isCollabComplete = (String.isNotBlank(collabRequestStatus) &&
46                          !(collabRequestStatus ==
    COLLAB_STATUS_COMPLETED || COLLAB_STATUS_ACCEPTED ==
    collabRequestStatus)) ?
47                              true :
48                              false;
49
50          if(BUSINESS_OBJECT_TYPE_PROPOSAL ==
    oLI.Apttus_Config2__ConfigurationId__r.Apttus_Config2__BusinessObjectType_
    _c
```

```
51                && (oLI.Apttus_Config2__IsReadOnly__c ||
52                    isCollabComplete ||
53                    PRODCONFIG_STATUS_PENDING_APPROVAL ==
     oLI.Apttus_Config2__ConfigurationId__r.Apttus_Config2__Status__c)){
54
55                    result.isSuccess= false;
56                    result.errorMessages.add(PARENT_UC_ERROR_MESSAGE);
57                    System.debug(LoggingLevel.ERROR, PARENT_UC_ERROR_MESSAGE);
58                    break;
59                }
60            }
61            return result;
62        }
63
64        global static Apttus_Config2.CustomClass.ActionCallbackResponse
     beforeCopyBundleLineItems(Apttus_Config2.CustomClass.ActionCallbackRequest
     callbackReq)
65        {
66            return new
      Apttus_Config2.CustomClass.ActionCallbackResponse();
67        }
68
69    }
```

Cart Approval Callback Class

Cart Approval Callback drives the Approvals functionality on the Cart page. You cannot use this callback to implement customization on CPQ. You can only use this callback to enable or disable the Approvals functionality. For more information, see Setting the Config Custom Classes.

Deal Optimizer Callback Class

Deal Optimizer Callback drives the Deal Maximizer functionality on the Cart page. You cannot use this callback to implement customization on CPQ. You can only use this callback to enable or disable the Deal Maximizer functionality. For more information, see Initiate and Integrate Deal Guidance.

Adjustment Spread Callback Class

Adjustment Spread Callback provides you a mechanism to restrict adjustment spread amount on option line items.

To use the Adjustment Spread Callback you must create a custom Apex class that implements the *Apttus_Config2.CustomClass.IAdjustmentSpreadCallback* interface and register the custom apex class with Adjustment Spread Callback Class. You must write your custom logic in the custom apex class.

The following methods are available in the *Apttus_Config2.CustomClass.IAdjustmentSpreadCallback* interface:

Method	Signature	Description
finish()	*void finish()*	This the last method that is executed after the callback is invoked.
spreadBundleAdjustment()	*List spreadBundleAdjustment(Apttus_Config2.LineItem, List)*	You can use this method to restrict the adjustment spread amount on options from the Bundle line adjustment.
spreadGroupAdjustment()	*List spreadGroupAdjustment(Apttus_Config2.SummaryGroup, List, Map>)*	You can use this method to restrict the adjustment spread amount on options from the summary group adjustment.
start()	*void start(Apttus_Config2.ProductConfiguration)*	This is the first method that is executed when the callback is invoked.

Example

The given code example implements both methods to restrict the adjustment spread values of option line items.

```
1   global with sharing class RG_AdjustmentSpreadCallback implements
     Apttus_Config2.CustomClass.IAdjustmentSpreadCallback
2   {
3
```

```
  4        private static Apttus_Config2__ProductConfiguration__c config;
  5        global static void start(Apttus_config2.ProductConfiguration cart)
  6        {
  7           config = cart.getConfigSO();
  8        }
  9
 10         global static List<Apttus_config2.LineItem>
           spreadGroupAdjustment(Apttus_config2.SummaryGroup parentGroup,
           List<Apttus_config2.LineItem> childLineItems, Map<ID,
           List<Apttus_config2.LineItem>> optionItemsByBundleId)
 11          {
 12
 13              List<Apttus_config2.LineItem> lines;
 14              if(config.Price_list_name__c== 'Auto Pricing Call Back Pricelist2')
 15              {
 16                     ID lineId = childLineItems.get(0).getLineItemSO().ID;
 17                     lines = optionItemsByBundleId.get(lineId);
 18                     for (Apttus_config2.LineItem optionLineItem : lines)
 19                     {
 20
 21                         //Below line use to restrict adjustment spread amount
           on option line items.
 22
           optionLineItem.getLineItemSO().Apttus_Config2__AdjustmentAmount__c =
           7.99999;
 23                     }
 24              }
 25              return lines;
 26          }
 27
 28
 29    global static List<Apttus_config2.LineItem>
       spreadBundleAdjustment(Apttus_config2.LineItem parentItem,
       List<Apttus_config2.LineItem> optionItems)
 30        {
 31          for(Apttus_Config2.LineItem OI: optionItems)
 32          {
 33            if(config.Price_list_name__c== 'Auto Pricing Call Back Pricelist')
 34            {
 35              Apttus_Config2__LineItem__c item = OI.getLineItemSO();
 36
 37              //Below line use to restrict adjustment spread amount on option
       line items.
```

```
38              item.Apttus_Config2__AdjustmentAmount__c = 8.99999;
39            }
40         }
41      return optionItems;
42    }
43  }
```

Lifecycle Callback Class

Lifecycle Callback provides you a mechanism to execute custom actions at the following stages in the quote process. class

- after a proposal is cloned
- after an agreement is created from a proposal
- creates a new proposal from the given account
- creates a new proposal from the given opportunity
- clones the given proposal
- creates a new agreement from the given proposal
- after a proposal is created from an account
- after a proposal is created from an opportunity

To use the Lifecycle Callback you must create a custom Apex class that implements the *Apttus_Proposal.CustomClass.IQuoteLifecycleCallback2* interface and register the custom apex class with Lifecycle Callback Class. You must write your custom logic in the custom apex class.

> ⓘ You must register the custom apex class for this callback in **Lifecycle Callback Class** in **Proposal Custom Classes** instead of **Config Custom Classes**.

The following methods are available in the *Apttus_Proposal.CustomClass.IQuoteLifecycleCallback2* interface:

Method	Signature	Description
afterClone()	*void afterClone(Apttus_Proposal__Proposal__c, Apttus_Proposal__Proposal__c)*	You can use this method to execute custom logic after the proposal is cloned. ⓘ CPQ does not invoke this method when you clone any proposal using the standard Salesforce **Clone** button. You must create a custom button or a formula field on the Proposal object to invoke this method after cloning a proposal.
afterCreateAgreement()	*void afterCreateAgreement(Apttus_Proposal__Proposal__c, Apttus__APTS_Agreement__c)*	You can use this method to execute custom logic after the agreement is created from proposal.
afterCreateFromAccount()	*void afterCreateFromAccount(Apttus_Proposal__Proposal__c)*	You can use this method to execute custom logic after the proposal is created from account. ⓘ CPQ only invokes this method when you execute the **Apttus_Proposal.ProposalWebService.afterCreateFromAccount** API.

Method	Signature	Description
afterCreateFromOpportunity()	*void afterCreateFromOpportunity(Apttus_Proposal__Proposal__c)*	You can use this method to execute custom logic after the proposal is created from opportunity. ⓘ CPQ only invokes this method when you execute the **Apttus_Proposal.Proposal WebService.afterCreateFromOpportunity** API.
afterComplete()	*void afterComplete(Apttus_Proposal__Proposal__c, Apttus_Proposal.CustomClass.ActionType)*	You can use this method to execute custom logic after proposal lifecycle is completed.
afterComplete()	*void afterComplete(Apttus__APTS_Agreement__c, Apttus_Proposal.CustomClass.ActionType)*	You can use this method to execute custom logic after agreement lifecycle is completed.

Example

You can use this callback to a scenario like:

- After you create a new proposal from Opportunity, values of **Bill To Account** and **Ship To Account** fields should be automatically set from the same fields that are present in the associated account.
- When you clone a proposal, you should have information about the proposal from which the new proposal was cloned from.

The below code implements the first example mentioned above.

```
/*
* Quote Life Cycle Call Back - Exmaple 1
* This is use case of executing Quote Life Cycle Call Back.
* Method - afterCreateFromOpportunity() of this class will be called whenever we go
to opportunity and click on Create Quote/Proposal button.
* Here, We will set BillToAccount and ShipToAccount fields of newly created proposal
by taking info of account from opporunity.
```

```
*/
global with sharing class APTS_PS_QuoteLifecycleCallback_UC1
    implements Apttus_Proposal.CustomClass.IQuoteLifecycleCallback2
{

    /**
     * Callback invoked after a quote/proposal is created from an account
     * @param quoteSO the new quote sobject
     */
    global void afterCreateFromAccount(Apttus_Proposal__Proposal__c quoteSO) {
    }

    /**
     * Callback invoked after a quote/proposal is created from an opportunity
     * @param quoteSO the new quote sobject
     */
    global void afterCreateFromOpportunity(Apttus_Proposal__Proposal__c quoteSO) {
        /* USECASE# 1 */

        String billTOAcc = quoteSO.Apttus_QPConfig__BillToAccountId__c;
        String shipTOAcc = quoteSO.Apttus_QPConfig__ShipToAccountId__c;
        String accId = quoteSO.Apttus_Proposal__Account__c;


        if(billTOAcc == null && shipTOAcc == null) {
            billTOAcc = accId;
            shipTOAcc = accId;
        } else if(billTOAcc == null) {
            shipTOAcc = billTOAcc;
        } else if(shipTOAcc == null) {
            billTOAcc = shipTOAcc;
        }


        quoteSO.Apttus_QPConfig__BillToAccountId__c = billTOAcc;
        quoteSO.Apttus_QPConfig__ShipToAccountId__c = shipTOAcc;

    }

    /**
     * Callback invoked after an agreement is created from a quote/proposal
     * @param quoteSO the quote sobject
     * @param agreementSO the new agreement sobject
```

```
    */
    global void afterCreateAgreement(Apttus_Proposal__Proposal__c quoteSO,
Apttus__APTS_Agreement__c agreementSO) {
    }


    /**
     * Callback invoked after a quote/proposal is cloned
     * @param originalSO the original quote sobject
     * @param cloneSO the clone quote sobject
     */
    global void afterClone(Apttus_Proposal__Proposal__c originalSO,
Apttus_Proposal__Proposal__c cloneSO) {
    }


    /**
     * Callback invoked before a quote/proposal is cloned
     * @param originalSO the original quote sobject
     * @param cloneSO the clone quote sobject
     */
    global void beforeClone(Apttus_Proposal__Proposal__c originalSO,
Apttus_Proposal__Proposal__c cloneSO) {
    }


    /**
     * Callback invoked after the completion of a lifecycle action
     * @param quoteSO the quote sobject
     * @PARAM actionType the lifecycle action type
     */
    global void afterComplete(Apttus_Proposal__Proposal__c quoteSO,
Apttus_Proposal.CustomClass.ActionType actionType) {
    }


    /**
     * Callback invoked after the completion of a lifecycle action
     * @param agreementSO the agreement sobject
     * @PARAM actionType the lifecycle action type
     */
    global void afterComplete(Apttus__APTS_Agreement__c agreementSO,
Apttus_Proposal.CustomClass.ActionType actionType) {
    }
}
```

Tax Callback Class

Tax Callback Class provides you the mechanism to communicate with a Tax Engine of your choice for tax calculation on invoice generation. You need to provide details like Invoice Line Item or a Credit Memo Line Item, Handback, Tax Address, Tax Code, and Taxable Amount to the callback. You must register the Tax Callback Class in Config Custom Classes in Custom Setting.

For more information, see Tax Callback Class. You can find information about how to setup the callback, method description, the working of the callback, and sample code in that topic.

Formula Callback Class

This callback has been deprecated. You must use the expression builder to create and use formulas.

Metadata Callback Class

This callback has been deprecated. Salesforce platform has been optimized with a better cache handle. This callback is no longer needed to maintain the metadata.

Recommendation Callback Class

Recommendation Callback provides you a mechanism to define products in the recommended products lists in addition to the products defined by the recommendation type constraint rules. This callback is invoked when constraint rules are executed on the Catalog page if the setting **Show Recommended Products Cart View** is enabled in Config Select Products Settings. After execution, Sales Rep sees the recommended products in the Recommendation popup.

To use the Recommendation Callback you must create a custom Apex class that implements the *Apttus_Config2.CustomClass.IRecommendationCallback* interface and register the custom apex class with Recommendation Callback Class. You must write your custom logic in the custom apex class.

The following methods are available in the
Apttus_Config2.CustomClass.IRecommendationCallback interface:

Method	Signature	Description
getRecommendedProducts()	*List getRecommendedProducts(Apttus_Config2.ProductConfiguration)*	You can use this method to define and recommend products. You must take into consideration the effective price list of the Quote. If the recommended product is not part of the price list in effecting for the quote, Sales Rep is not allowed to use those products in any operation.

Example

The given code example defines a product for recommendation.

```
global with sharing class SampleRecommendationCallback implements
 Apttus_Config2.CustomClass.IRecommendationCallback
{

    global List<Apttus_Config2.CustomClass.RecommendationInfo>
getRecommendedProducts(Apttus_Config2.ProductConfiguration cart)
    {
        Apttus_Config2.CustomClass.RecommendationInfo recommendedProduct =
new Apttus_Config2.CustomClass.RecommendationInfo('a0S0B000004JrYF',
'Relevance');

        // product ID, relevance

        return new
 List<Apttus_Config2.CustomClass.RecommendationInfo>{recommendedProduct};
    }
}
```

Advanced Approval Callback Class

This callback has been deprecated. Conga recommends not define any customization for approvals.

Configuring Admin Settings

Admin Settings are system properties that have different values and you can use them for different purposes. You must configure admin settings to meet the business objectives of your implementation along with Custom Settings.

To create admin settings

1. Click ✚ and click **Admin**. All the standard Admin Settings are displayed on the Admin home page. Based on your organization's requirements, you can add or create new Admin settings.
2. To create a new Admin Setting, click **New**.
3. Enter **Name**, **Value**, and **Code** for the Admin Setting and click **Save**. Your new Admin Setting is saved and added.

Admin Settings in CPQ

The following content provides information about various admin settings available in CPQ along with their values and purposes.

APTS_CriteriaExpressionsUpdateBatchSize

During Criteria Maintenance, this setting ensures optimal performance by managing expression fields, preventing Apex heap size limits from being exceeded.

Name	*APTS_CriteriaExpressionsUpdateBatchSize*
Value	Enter a number.
Code	Leave the field blank

APTS_ABOActionCallbackClass

You can use this admin setting to manage the sorting order of asset records in the installed products page. The records are displayed in descending order by default. This allows sorting based on specified fields in the apex class, enabling better visibility and

organization of bundled assets.

If we take a use case: There are equipment line items converted into Asset Line item records. These assets are then listed on the installed products page and further used for quoting service products. Frequently, Equipment bundles also have sometimes child bundles, and if a parent bundle has a child bundle configured inside it, then we have to show the parent bundle asset and child bundle asset separately. In this case, sorting can be done based on the asset name.

This sample code shows the sorting based on the quantity.

> ⓘ If you enable this setting, CPQ disables regular sorting on the installed products page column headers.

Name	APTS_ABOActionCallbackClass
Value	Custom class name: customABOcallback
Code	Leave the field blank

APTS_DisableRollupOptionStartDate

You can use this admin setting to manage the alignment between option and bundle start dates. Set it to TRUE for preserving bundle start dates and delinking them from option start dates.

Name	APTS_DisableRollupOptionStartDate
Value	• *True* • *False*
Default Value	*False*
Code	Leave the field blank.

APTS_EnableDirtySaveOnExit

Use this admin setting to save the cart when the sales representative navigates to the proposal summary page using a View link. When the **APTS_EnableDirtySaveOnExit** admin setting is set to *True,* it enables CPQ to automatically save the cart, including all

configurations, when a sales representative navigates to the proposal summary page by clicking the 'View' link in the menu under the proposal name on the Catalog page.

> ⚠ Enabling **APTS_EnableDirtySaveOnExit** with a value of *True* will override the behaviour of the **APTS_OpenLinkInNewTab** setting when it is also set to *True.*

Name	APTS_EnableDirtySaveOnExit
Value	• *True* • *False*
Default Value	*False*
Code	Leave the field blank.

APTS_OpenLinkInNewTab

You can use this admin setting to control the display of quotes with a click of the **View** button. When this admin setting is *True,* if the sales representative clicks the **View** button (in the menu under the proposal name) on the Catalog page, CPQ displays the quote on a new tab.

Name	APTS_OpenLinkInNewTab
Value	• *True* • *False*
Default Value	*False*
Code	Leave the field blank

APTS_OpenNewTabOnCartAction

You can use this setting to determine where the cart actions are to be performed.

Name	APTS_OpenNewTabOnCartAction

Value	• True • False
Code	Leave the field blank

> ⓘ • If *APTS_OpenNewTabOnCartAction = True*, the system will open cart actions in a new tab for Sales Console App.
> • if *APTS_OpenNewTabOnCartAction = False* or not defined, the system continues to open cart actions in the same tab for the sales console app.
> • This Admin entry does not have any impact on the cart from the proposal mgmt app or normal flow.

APTS_DisableNumberingForCRMessages

You can use this setting to hide or show the numbering next to the constraint rule messages on CPQ UI.

Name	*APTS_DisableNumberingForCRMessages*
Value	• *True : CPQ removes numbering to the CR messages and warnings.* • *False*
Code	Leave the field blank.

APTS_AsyncMergeCall

You must use the custom setting **Enable Submit Merge Call** in Proposal System Properties instead of the admin setting APTS_AsyncMergeCall. For more information, see Proposal System Properties.

APTS_AsyncMergeEmail

You can use this setting to send a notification email to the Sales rep initiating the asynchronous document generation after generation is complete.

> ⓘ Conga recommends to enable both **APTS_AsyncMergeCall** and **APTS_AsyncMergeEmail**.

Name	APTS_AsyncMergeEmail
Value	• *True* • *False*
Code	Leave the field blank.

APTS_CartLineItemsUpdateChunkSize

You can use this setting to determine the number of line items to be updated in each updateCartlineitem call.

Name	APTS_CartLineItemsUpdateChunkSize
Value	Enter a number.
Code	Leave the field blank.

APTS_CRMaxFetch

You can use this setting to determine the maximum number of constraint rules to be fetched.

Name	APTS_CRMaxFetch
Value	Enter a number. If you do not enter a number, CPQ fetches 1000 constraint rules by default.
Code	Leave the field blank.

APTS_CustomRevalidationLinesThreshold

You can use this setting to control lines items sent in the custom revalidation request object.

Name	APTS_CustomRevalidationLinesThreshold

Value	Enter a number.
	• If the value is less than the number of revalidation lines, the admin setting sends the whole lineItemSOs in the lineItemSOsByRevalCategory map.
	• If the value is more than the number of revalidation lines, the admin setting sends only lineItemIds in the lineItemIdsByRevalCategory map to control the head size. In this case, lineItemSOsByRevalCategory is empty and you must query lineItems into the revalidation callback with required fields.
Code	Leave the field blank.

APTS_DefaultEmailContactName

You can use this setting to define the default email contact name.

Name	*APTS_DefaultEmailContactName*
Value	Enter a value to define the default email contact name.
Code	Leave the field blank.

APTS_DisableCartActivityHistory

You can use this setting to disable the generation of activity history product configuration. When enabled, CPQ does not add entries to the **Activity History** related list, when you create, finalize, or reconfigure a product configuration.

Name	*APTS_DisableCartActivityHistory*
Value	Enter one of the following valid values:
	• *True*
	• *False*
Code	Leave the field blank.

APTS_DisableCartAsyncNotification

You can use the Admin Setting **APTS_DisableCartAsyncNotification** to turn off the email notifications you receive after completion of any async operation such as cart finalization and pricing in every type of quote or agreement. In the Split quote or agreement, the notification for cart splitting is also disabled along with finalization and pricing. If you enabled the Progress Tracker on the quote or agreement page to monitor the progress of all the async operations, email notifications are redundant, and you can use this feature to avoid them.

Name	*APTS_DisableCartAsyncNotification*
Value	*true*
Code	Leave the field blank.

APTS_DisableConfigurationLoading

You can use this setting to, when the value is set to True, enable attributes or options. Enabling this setting does not make attributes or options unavailable.

Name	*APTS_DisableConfigurationLoading*
Value	Enter one of the following valid values: • *True* • *False*
Code	Leave the field blank.

APTS_DisableConstraintRulesOnFinalize

You can use this setting to disable the execution of server-side constraint rules to optimize performance upon cart finalization.

> ⓘ This feature is not supported with the Replacement type Constraint Rule.

Name	APTS_DisableConstraintRulesOnFinalize
Value	Enter one of the following valid values: • *True* • *False*
Code	Leave the field blank.

APTS_DisableCumulativeQuantityAcrossCart

You can use this setting to disable CPQ from applying the cumulative range matrix on multiple line items of the same product.

Name	APTS_DisableCumulativeQuantityAcrossCart
Value	Enter one of the following valid values: • *True* • *False*
Code	Leave the field blank.

APTS_DisableOpptyForRenewalAgreement

You must use the custom setting **Create Renewal Opportunity** in Installed Product Settings instead of the admin setting APTS_DisableOpptyForRenewalAgreement. For more information, see Installed Products Settings.

APTS_DisableOpptyForRenewalQuote

You must use the custom setting **Create Renewal Opportunity** in Installed Product Settings instead of the admin setting APTS_DisableOpptyForRenewalQuote. For more information, see Installed Products Settings.

APTS_DisablePLITimeZoneAdj

You can disable the time zone adjustment on the **Effective Date** and Expiration **Date** fields in the Price List Item. By default, the time on the **Effective Date** and **Expiration Date** fields is

adjusted based on the Time Zone of the Locale in the org. If you disable the adjustment, CPQ does not change the time you manually configured in the **Effective Date** and Expiration **Date** fields. Refer to Adding Additional Details to a Price List Item for information about the Price List Item fields.

Name	APTS_DisablePLITimeZoneAdj
Value	The following are the valid values: • *true*: disables time zone adjustment. • *false*: enables time zone adjustment.
Code	Leave the field blank.

APTS_DisableRolldownOptimization

You can use this setting to optimize the system behavior while using custom logic in the methods invoked in the Roll Down mode.

Name	APTS_DisableRolldownOptimization
Value	The following are the valid values: • *True*: the method in Rolldown mode is invoked when the Sales rep clicks **Go To Pricing** or **Reprice** after changing the quantity or applying summary level discount. • *False*: the method in Rolldown mode is only invoked when the Sales rep clicks **Reprice** after applying summary level discount.
Code	Leave the field blank.

APTS_DisableSOSLTextSearch

You must use the custom setting **Disable SOSL Text Search** in Config System Properties instead of the admin setting APTS_DisableSOSLTextSearch. For more information, see Config System Properties.

APTS_EnableBlockingCallsOnServiceCatalog

Use this admin setting to display a spinner on the Service Catalog upon the click of the **Add to Cart** or **Update** buttons. When this admin setting is set to true, CPQ displays the spinner to ensure that the creation of related line items is completed for service products before you navigate away to a different page or click another button.

> ⚠ When this admin setting is set to false, if you quickly navigate away from the Service Catalog after clicking **Add to Cart**, CPQ does not create related line items for service products.

Name	APTS_EnableBlockingCallsOnServiceCatalog
Value	Enter one of the following valid values: • *True* • *False*
Code	Leave the field blank.

APTS_EnableABOChangeOptimization

The performance team uses this admin setting only for testing.

APTS_EnableCollabForAutoIncludedLines

You can use this setting to enable collaboration for auto-included standalone or bundle products. If you initiate a collaboration request for the parent product, CPQ automatically adds auto-included products as part of the collaboration request (except Prompt). You can also create a collaboration request for auto-included products independent of the parent product.

This admin setting is supported only for **Peer-To-Peer** type of collaboration.

Name	APTS_EnableCollabForAutoIncludedLines

Value	Enter one of the following valid values: • *True*: CPQ enables collaboration for auto-included standalone or bundle products. Also, CPQ displays the collaboration icon ( ) next to an auto-included line items. • *False*: This is the default value. CPQ disables collaboration for auto-included standalone or bundle products. CPQ hides the collaboration icon ( ) next to an auto-included line items.
Code	Leave the field blank.

APTS_EnableDefaultQuantityForABOItem

You must use the custom setting **Enable Default Quantity For ABO Item** in Config System Properties instead of the admin setting APTS_EnableDefaultQuantityForABOItem. For more information, see Config System Properties.

APTS_EnableInstallBaseFilteringAcrossAccounts

You can use this setting along with Asset Line Item Callback to display assets from multiple accounts, which are related to an opportunity, on the Installed Products page. For more information to configure the callback, see Displaying Assets from Multiple Accounts on the Installed Products Page.

Name	*APTS_EnableInstallBaseFilteringAcrossAccounts*
Value	Enter one of the following valid values: • *True*: CPQ displays assets from multiple accounts. • *False*: This is the default value. CPQ displays assets from a single account related to the quote.
Code	Leave the field blank.

APTS_EnableMultiBenefitItems

You can use this setting to control the ability of the Sales rep to apply promotions to multiple line items of the benefit product (Y). By default, this setting is disabled and CPQ

applies the promotion only to the first line item of the benefit product (Y). When you enable this setting, CPQ applies the promotion to all line items of the benefit product (Y).

Name	APTS_EnableMultiBenefitItems
Value	Enter one of the following valid values: • *True* • *False*
Code	Leave the field blank.

APTS_EnableMultiCurrencyPriceFormula

You can use this setting to indicate pricing formulas using a field on the line item or a PAV extension field in price matrix or price rule where pricing is already adjusted for a foreign currency.

Name	APTS_EnableMultiCurrencyPriceFormula
Value	Enter one of the following valid values: • *True* • *False*
Code	Leave the field blank.

APTS_EnablePriceRuleForABOItem

You must use the custom setting **Enable Price Rule For ABO Item** in Config System Properties instead of the admin setting APTS_EnablePriceRuleForABOItem. For more information, see Config System Properties.

APTS_EnableSessionReplay

This admin setting is no longer available.

APTS_EnableStaticConfigMessages

You can use this setting to display constraint rule messages when you scroll up and down to select options on the Configuration page. The messages are displayed at the top of the screen even when you have scrolled down to select the options. CPQ updates the messages as you update the selection of the options.

Name	APTS_EnableStaticConfigMessages
Value	Enter one of the following valid values: • *True* • *False*
Code	Leave the field blank.

APTS_ExcludeOptionalProducts

You must use the custom setting **Exclude Optional Products** in Config System Properties instead of the admin setting APTS_ExcludeOptionalProducts. For more information, see Config System Properties.

APTS_ExecuteMinMaxRulesOnInactiveOptions

You can use this setting to, when the value is set to True, execute prevent Sales Representatives from finalizing the cart if the bundle configuration does not honor the **Min Options** configured for an option group of that bundle. This feature is useful when:

- **Min Options** is defined for an option group.
- The option group is hidden because the options under it do not have any active price list items.
- Sales Representative finalizes the cart when there are no active products in an option group to fulfil the minimum number of options.

Name	APTS_ExecuteMinMaxRulesOnInactiveOptions

Valu e	Enter one of the following valid values: • *True*: CPQ displays an error message on the cart that configuration is pending for the bundle that has the option group (where options have inactive PLIs). • *False*: This is the default value. CPQ does not display any error message on the cart.
Cod e	Leave the field blank.

APTS_ExecuteRulesOnValidate

You can use this setting to, when the value is set to True, execute all rules on click of the **Validate** button.

Name	*APTS_ExecuteRulesOnValidate*
Value	Enter one of the following valid values: • *True* • *False*
Code	Leave the field blank.

APTS_IncentivePricingBatchSize

You can use this setting to process the promotions in batches for a cart with a large number of line items. The batch processing is only supported for Buy X Get X and Buy X Get Y promotions with Auto Apply? enabled. You can use this feature for both regular and async pricing.

Name	*APTS_IncentivePricingBatchSize*
Value	Enter the number of line items to be processed in a batch. ⓘ You must define the batch size based on the complexity of the cart and the number of incentives applied on the line items.
Code	Leave the field blank.

APTS_LargeCartThreshold

You must use the custom setting **Split Cart Threshold** in Config System Properties instead of the admin setting APTS_LargeCartThreshold. For more information, see Config System Properties.

APTS_LargeDocSObjects

You can use this setting to define objects as Large Objects for document generation. You can also define a batch size for each of these objects. By default, CPQ considers Proposal Line Item and Agreement Line Item as Large Objects for document generation.

Name	APTS_LargeDocSObjects
Value	Leave the field blank
Code	Use the format Auto Applied *<Object_API_name=<Batch Size>>* to define the objects. You can add more objects separated by a new line. Decide the batch size based on the complexity of the cart and template. For example, *Apttus_QPConfig__ProposalLocation__c=5* *QTC_Proposal_Related_Line_Item_c=1000*

APTS_LargeDocThreshold

You must use the custom setting **Large Doc Threshold** in Proposal System Properties instead of the admin setting APTS_LargeDocThreshold. For more information, see Proposal System Properties.

APTS_LoadProductRelatedAttributesOnly

You can use this setting to enable the optimized retrieval of attributes on the cart page. When this setting is enabled, CPQ only retrieves the attributes in the cart that are associated with the product added to the cart. By default, all the attributes from Product Attribute Value, Product Attribute Value Ext, Product Attribute Value Ext 2, Product Attribute Value Ext 3, and custom Product Attribute Value Ext objects are loaded on the Cart page.

Name	APTS_LoadProductRelatedAttributesOnly
Value	Enter one of the following valid values: • *True* • *False*
Code	Leave the field blank.

APTS_MergeCallTimeout

You can use this setting to display a timeout message whenever document generation exceeds 60 seconds.

Name	APTS_MergeCallTimeout
Value	Enter one of the following valid values: • *True* • *False*
Code	Leave the field blank.

APTS_MS_EnableNameSort

You can use this setting to specify whether the related list should be sorted by the name field in the generated document. When enabled the related lists will be printed in the merge documents in the order of the name field of the related list.

Name	APTS_MS_EnableNameSort
Value	Enter one of the following valid values: • *True* • *False*
Code	Leave the field blank.

APTS_NGUIOverride

You must use this setting only for troubleshooting errors. This setting overrides the ngCPQ UI with a static resource file.

APTS_NoISOCurrencyFieldObjects

You can use this setting to add support for custom object generation for objects that do not have a CurrencyIsoCode field (e.g., "EventRelation," "Task," "TaskRelation"). Separate Object names using a comma or new line. The setting is only applicable to templates in multi-currency org.

Name	APTS_NoISOCurrencyFieldObjects
Value	XML
Code	<Object API name>

APTS_NoNameFieldObjects

You can use this setting to add support for custom object generation for Objects that do not have a Name field. (for example, "EventRelation," "Task," "TaskRelation") Separate Object names using a comma or new line.

Name	APTS_NoNameFieldObjects
Value	XML

Name	APTS_NoNameFieldObjects
Code	<Object API name>

APTS_OrderItemActivateBatchSize

You must use the custom setting **Order Item Activate Batch Size** in Order System Properties instead of the admin setting APTS_OrderItemActivateBatchSize. For more information, see Configuring Order Management Custom Settings.

APTS_OrderItemCreateBatchSize

You must use the custom setting **Order Item Create Batch Size** in Order System Properties instead of the admin setting APTS_OrderItemCreateBatchSize. For more information, see Configuring Order Management Custom Settings.

APTS_Password

You can use this setting to create a password for protecting Microsoft Word agreements that are generated by the application. It also allows the application to password protect generated Microsoft Word documents.

Name	APTS_Password
Value	<password>
Code	Leave the field blank.

APTS_PriceListItemCustomFields

You can use this setting to auto-retrieve the custom fields in the **onPriceItemSet()** method in the Pricing Callback Class. For more information on Pricing Callback methods, see Pricing Callback Class.

Name	*APTS_PriceListItemCustomFields*
Value	Leave the field blank.

Name	APTS_PriceListItemCustomFields
Code	Enter the API names of the custom field of the Price List Item separated by a comma or on a new line.

APTS_PricingCartRestoreMode (formerly known as APTS_RestoreLinesOnChange)

You can use this setting to maintain a replica of the cart on TurboPricing. If line items are modified after the cart is saved, this admin setting communicates those changes to the pricing engine whenever you launch the cart so that pricing is up-to-date.

> ⓘ APTS_RestoreLinesOnChange is renamed as APTS_PricingCartRestoreMode from the Conga CPQ May '22 release.

Name	APTS_PricingCartRestoreMode
Value	Enter one of the following valid values: • Quick: The user interface sends only price pending line items to TurboPricing on cart launch. • Full: The user interface restores full payload for all line items.
Code	Leave the field blank.

APTS_PricingServiceOverrideURI

You can use this setting to enable TurboEngines on an org. For more information, see Setting Up the TurboPricing Endpoint URLs.

APTS_PricingServiceUrl

You must use this setting to set the Turbo Engines URLs. For more information, see Setting Up the TurboPricing Endpoint URLs.

APTS_PricingSocketKeepAliveInterval

You must use this setting to re-establish the WebSocket connection before it is timed out. The WebSocket connection is disconnected every 50 seconds for security reasons.

Name	APTS_PricingSocketKeepAliveInterval
Value	Less than or equal to 45

Code	Leave the field blank.

APTS_ProductDetailsChunkSize

> ❗ You must use the custom setting **Maximum Product Details Per Request** in Config System Properties instead of the admin setting APTS_ProductDetailsChunkSize. For more information, see Config System Properties.

APTS_ProposalConfig

You can use this setting to enable pdf security on the generated Quote/ Proposal document. Before defining this setting you must configure **Enable PDF Security** and **PDF Owner Password**.

Name	*APTS_ProposalConfig*
Value	Enter the value *XML*.

Code	Enter the following code:

```
<ProposalConfig>
<PDFSecurityDefault>
<CanPrint>true</CanPrint>
<CanCopy>true</CanCopy>
<CanChange>false</CanChange>
<CanAddNotes>true</CanAddNotes>
<CanFillFields>true</CanFillFields>
<CanAssemble>false</CanAssemble>
</PDFSecurityDefault>
</ProposalConfig>
```

You can also use this setting to define the line and option sequence.

Name	APTS_ProposalConfig
Value	Enter the value XML.
Code	Enter the following code:

```
<ProposalConfig>
<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>
</ProposalConfig>
```

Depending upon your business object (Quote/Proposal or Agreement), you can enable this for *ProposlConfig* or *ComplyConfig* respectively.

APTS_RefineSearchCheckboxDisplayLimit

You must use the custom setting **Refine Search Checkbox Display Limit** in Config System Properties instead of the admin setting APTS_RefineSearchCheckboxDisplayLimit. For more information, see Config System Properties.

APTS_RollUpOptionNetAdjustment

You can use this setting to display the adjustments on options at the bundle and summary totals level.

Name	APTS_RollUpOptionNetAdjustment
Value	Enter one of the following valid values: • *True* • *False*
Code	Leave the field blank.

APTS_ShowSendEmailConfirmation

You can use this setting to display a confirmation popup of email sent after presenting a quote using the **Present** button.

Name	APTS_ShowSendEmailConfirmation
Value	Enter one of the following valid values: • *True* • *False*
Code	Leave the field blank.

APTS_SortDocumentsDesc

This admin setting is deprecated. You must use the custom setting **Sort Documents Descending** in Proposal System Properties instead of the admin setting APTS_SortDocumentsDesc. For more information, see Proposal System Properties.

> ⓘ If you are upgrading from the May '22 or September '22 to February '23 or later releases of CPQ, you must remove the **APTS_SortDocumentsDesc** admin setting. The admin setting is no longer supported.

APTS_TranslatePickValuesInDocs

You can use this setting to, when the value is set to True, get translated value of picklist options in the generated document.

Name	APTS_TranslatePickValuesInDocs
Value	Enter one of the following valid values: • *True* • *False*
Code	Leave the field blank.

APTS_UIPageConfig

You can use this setting to clone a quote with line items and redirect it to a custom VF page.

Name	APTS_UIPageConfig
Value	Enter the value *XML*.

Code	Enter the following code:
	
	 ``xml
	<UIPageConfig>
	<UIAction name="clone_proposal">
	<CustomPage>
	<Name>CustomCloneProposal</Name>
	</CustomPage>
	</UIAction>
	</UIPageConfig>
	``
	
	 Here "CustomCloneProposal" is a custom VF page.

APTS_UpdateViewApiBatchSize

You can use this setting to specify the batch size for the Category Maintenance batch job when the job is executed using the exposed global Apex method CPQWebService.updateHierarchyViews(). You must specify a batch size value only when you notice limit exceptions in batch jobs because of complex categories with a lot of hierarchy levels. In such cases, you must decrease the batch size to execute the batch jobs successfully.

Name	APTS_UpdateViewApiBatchSize
Value	Enter a value between 1 and 2000. The default batch size for the Category Maintenance batch job set in this global method is 2000.
Code	Leave the field blank.

APTS_UpdateViewProductBatchSize

You can use this setting to define the product batch size for update view.

Name	APTS_UpdateViewProductBatchSize

Value	Enter a number. • If the APTS_UpdateViewUseDmlLimit admin setting set to True, the value of the APTS_UpdateViewProductBatchSize setting must be (10,000/# of price lists)*2 • If the APTS_UpdateViewUseDmlLimit admin setting set to False, the value of the APTS_UpdateViewProductBatchSize setting must be (50,000/# of price lists)*2
Code	Leave the field blank.

APTS_UpdateViewProductClasses

You can use this setting to execute the Category Maintenance batch job for the changes you make modifications to existing records instead of executing the batch job on the entire category definition.

Name	*APTS_UpdateViewProductClasses*
Value	Leave the field blank
Code	Enter the Product Classification records of the newly added products. ⓘ These records are created after you associate products to the category. In case, you want to execute a maintenance batch job for the entire definition, clear the **Code** field, and then execute the Category Maintenance batch Job.

APTS_UpdateViewUseDmlLimit

You can use this setting to, when the value is set to True, consider DML Salesforce limits for executing category maintenance jobs.

Name	*APTS_UpdateViewUseDmlLimit*
Value	Enter one of the following valid values: • *True* • *False*

Code	Leave the field blank.

APTS_UseWizardServer

Conga Contract Lifecycle Management uses this admin setting. For more information, see Admin Entries in CLM documentation.

APTS_VirtualizationThreshold

This admin setting is deprecated. You must use the custom setting **Grid Render Threshold** in Config System Properties instead of the admin setting **APTS_VirtualizationThreshold** to manage scroll performance. For more information, see Config System Properties.

Apttus_QPConfig__CloneWithApprovalStatus__c

You must use the custom setting **Clone With Approval Status** in Proposal System Properties instead of the admin setting Apttus_QPConfig__CloneWithApprovalStatus__c. For more information, see Proposal System Properties.

EnableContextualTotals

You must use the custom setting **Enable Contextual Totals** in Config System Properties instead of the admin setting EnableContextualTotals. For more information, see Config System Properties.

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 Conga 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 Conga 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**
>
> Conga 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 Conga 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 Conga 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
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.

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.
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 Conga Integration Adaptor cannot auto-populate these fields.
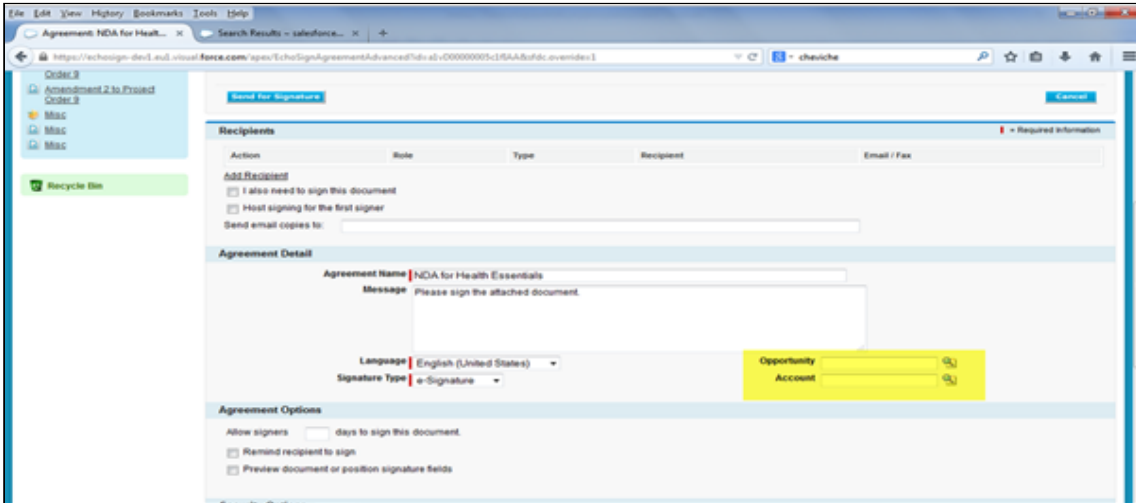
To auto-populate these fields, employ one of the following work-around.

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__PriceLis
tItem__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, 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 Conga Managed field?

To change the label of a Conga Managed field,

1. Click **Setup** > **Translation Workbench** > **Override**.
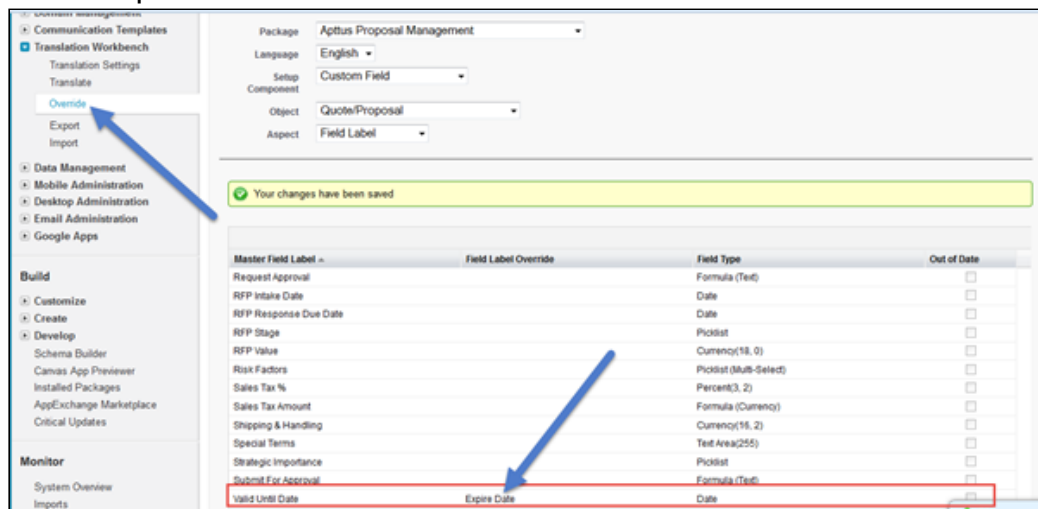
2. Double click in the **Field Label Override** column to enter a new value.
3. Click **Save**.

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 Conga Generate document functionality on any custom object?

You can add the Conga 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**.
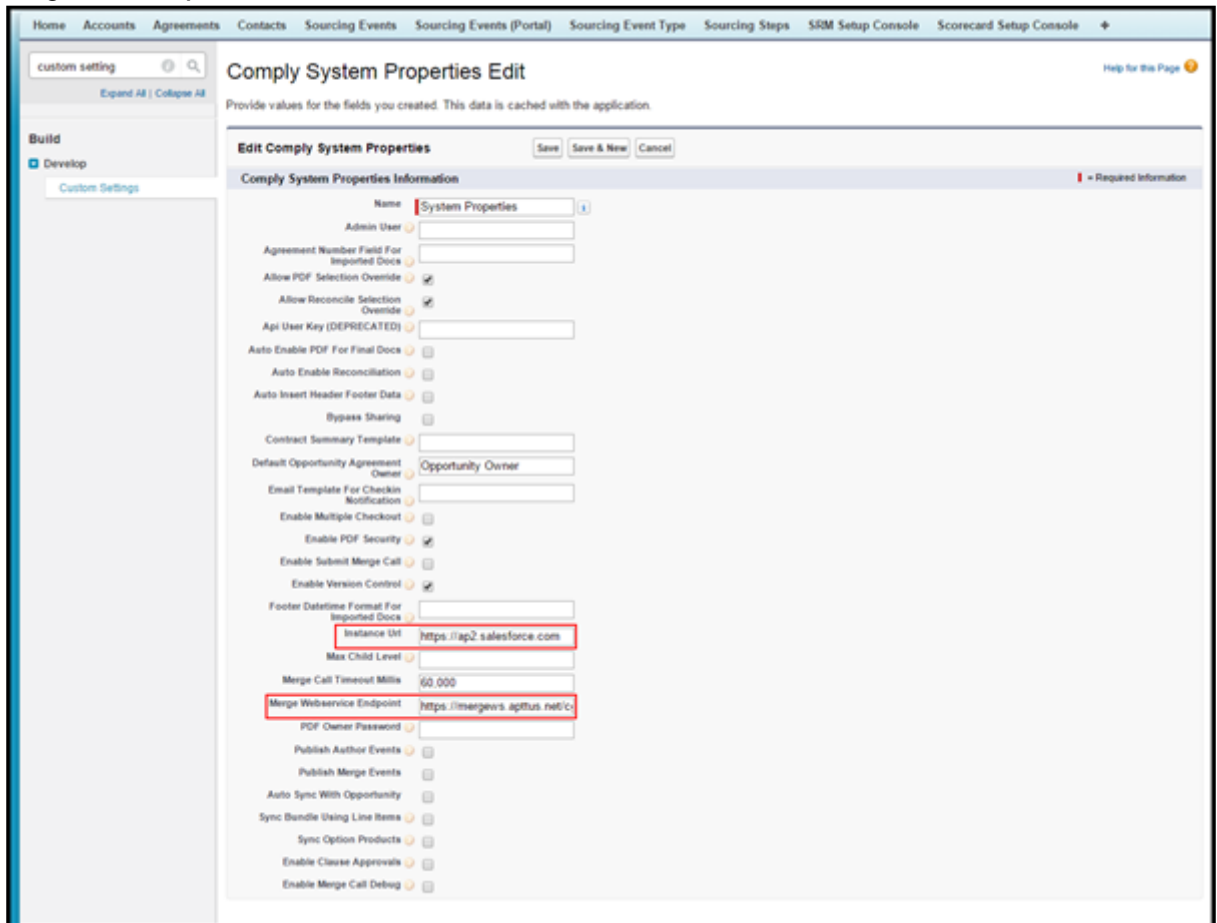
2. Define values for the following fields.
   - **Merge Webservice Endpoint**
   - **Instance URL**
3. Click **Save**.

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_Document&templateTyp
e=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?
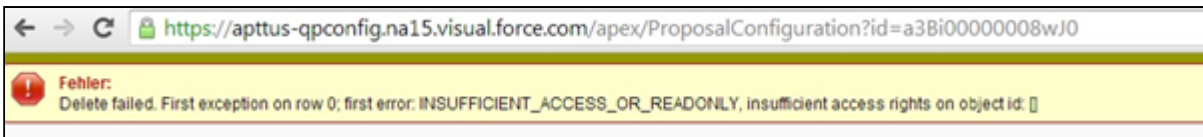
You will need **Modify All** permission on Product Configurations object.

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.
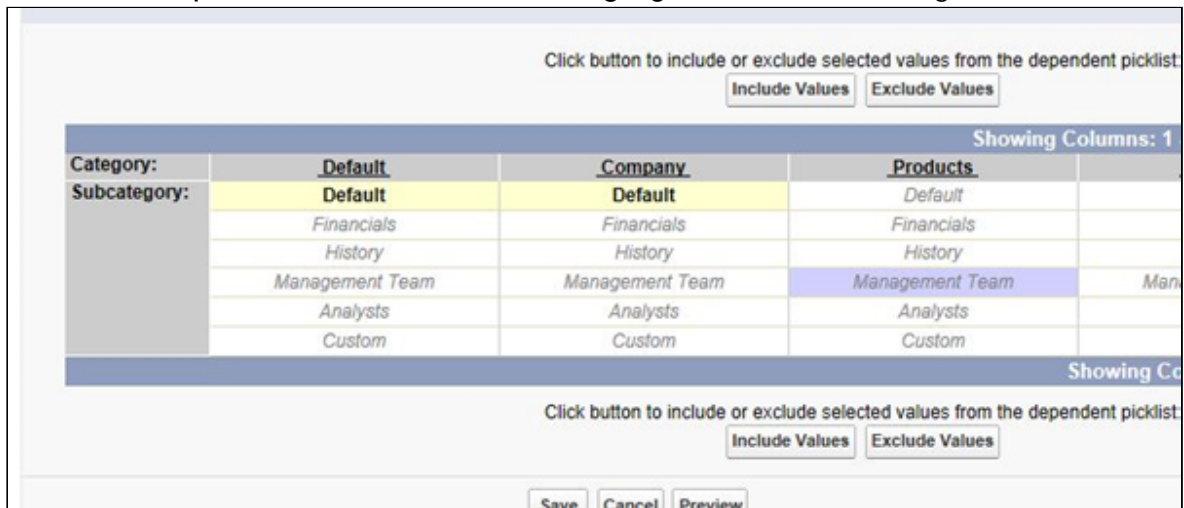
What is the difference between the custom PAV extensions and the out-of-the-box PAV extensions?

The custom PAV extensions were the first version of extension attributes to allow customers to create more attributes to use. From this perspective, there is no difference between the custom PAV extensions and the out-of-the-box PAV extensions except for the predefined objects that must be used. However, custom PAV extensions are not officially supported in the CPQ Admin UI when looking for attribute extensions to add in various places, whereas out-of-the-box PAV extensions are supported.

What happens if I upgrade to a release later than September '22 and do not migrate custom PAV extensions the out-of-the-box PAV extensions?

Whatever currently works for custom PAV extensions will continue to work. There will be no removal in functionality and Conga will fix any regression issues.

What happens to my in-flight quotes that are already using custom PAV extension fields?

If a quote if already saved or finalized and has a line item saved with the custom PAV extension data, the attribute information will not change unless the quote is reconfigured (opening the cart). If the quote is reconfigured, the attribute must be reselected for the same changes to be applied. The only caveat to this is for read-only carts that were previously saved and no changes are made.

If this is a large process to undertake for manual reselection, we recommend creating a script to map values selected for the custom PAV extension to the new out-of-the-box PAV extension field.

How do I request an enhancement for custom attribute extensions?

As of September '22, no new enhancement requests will be done for custom PAV extensions. It is recommended to move to the out-of-the-box PAV extensions and see if the enhancement exists there. If it does not, we have an Ideas portal for any enhancement requests for any Conga feature.

When will custom PAV extensions be completely removed from Conga CPQ?

There are no plans to completely remove the objects from Conga CPQ but full end of life (no issue fixes, no enhancements) will occur in 2024.

What if I need more attribute fields than the 3200 given by the out-of-the-box PAV and PAV extension objects?

We can work with customers that have higher attribute requirements to expand our feature set. Please use the Idea portal to make an enhancement request.

Conga Contact Support

If you experience an issue with an Conga product and need help, you can contact Conga Support. Before you contact Conga 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.

What is your Salesforce.com Organization ID?

To determine the Salesforce.com organization ID:

1. Go to **Setup > Administration Setup > Company Profile > Company Information**.
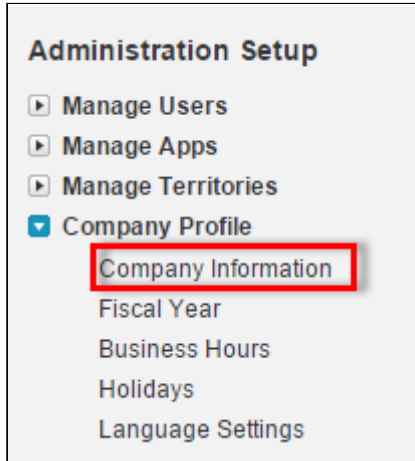
2. From the Organization Detail pane, provide the Salesforce.com Organization ID.

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 Setting**s.
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**.

System Fields

All fields in managed objects are system fields, except for explicit override mechanisms that CPQ provides through user input or APIs.

The following are some examples of the fields for which CPQ accepts user input through the out-of-the-box CPQ UI or globally exposed APIs:

- Price List
- Quantity
- Term
- Start Date and End Date
- Coupon Codes
- Adjustments
- Base Price Override
- Product Attributes

> ❗ Except those fields that are available to be modified using the out-of-the-box CPQ UI and APIs, all other managed fields are system fields. Therefore, you must not manipulate them because of the risk of data integrity issues.

In case you must manipulate a particular system field value because of business requirements, you must contact Conga Support to evaluate if one of the following options is possible:

- Is there any existing out-of-the-box mechanism to do that?
- If not, explore the possibility of CPQ providing such a mechanism.
- If both options above are not possible, Conga Support must get approval from the CPQ Product team so that the Product team can track such fields and maintain backward compatibility when developing an enhancement or fixing any issue related to such system fields.