



Pricefx Accelerators

Accelerate CPQ Package

March 2022

Accelerate CPQ Package (CPQ)

The CPQ Package consists of various components to support the whole process of negotiating a deal - selecting customers and products, pricing and negotiating the deal, approval workflow, quote output. The setup does not require heavy configuration, you only need to provide a few data tables and map transactional data to provide sales insight during the negotiation.

- [Product Info \(CPQ\)](#)
- [Quoting Tool](#)
- [CPQ Package Configurations](#)
- [CPQ Package Version 2.0](#)
- [CPQ Configuration Wizard](#)
- [CPQ Package Architecture](#)
- [Configurator Package Integration](#)
- [Configurations Per Quote Type](#)
- [CPQ Element Names for Approval Workflow](#)
- [CPQ Package Release Notes](#)
- [Archive of Previous Versions of CPQ Package](#)

Product Info (CPQ)

Accelerate CPQ Package provides sales representatives with the latest product and customer details, discount guidelines, recommended target, stretch, and floor prices, and other relevant details to help them negotiate the right price with their customers.

This package includes:

- Customer discounting at the customer and/or item level
- Product configuration
- Excel/Word/PDF based quote templates
- Email quote to customer capability
- Approval workflow through the Accelerate Approval Workflow Package

Quoting Tool

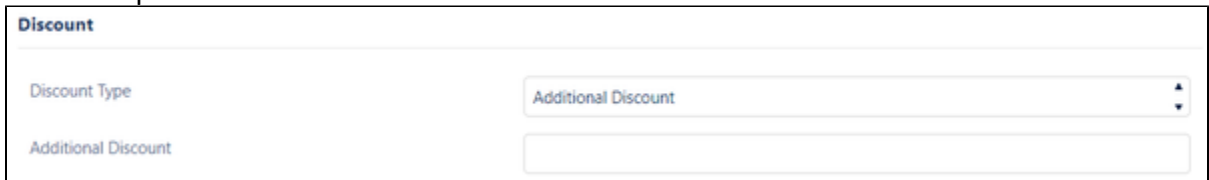
- [Header Level View](#)
- [Line Item Level](#)
- [Data Requirements](#)
- [Setup](#)

Header Level View

On the header level, the package provides standard input entries, summarized KPIs of quoted items, historical purchases of the customer for quoted items and some general KPIs.

User Inputs

- Customer - Allows the user to choose a customer for the quote; the package allows to set a filter which can limit availability of the customers for selection.
- Currency - Allows the user to choose a customer currency that is defined in the ccy Data Source or from Price Parameters table (version 1.2.2).
- Discount % - Optional user input which can enable one of the predefined discount models to make negotiating easier.
 1. Additional Discount - Allows to define an additional discount (absolute value) on top of discounts provided on the header level.

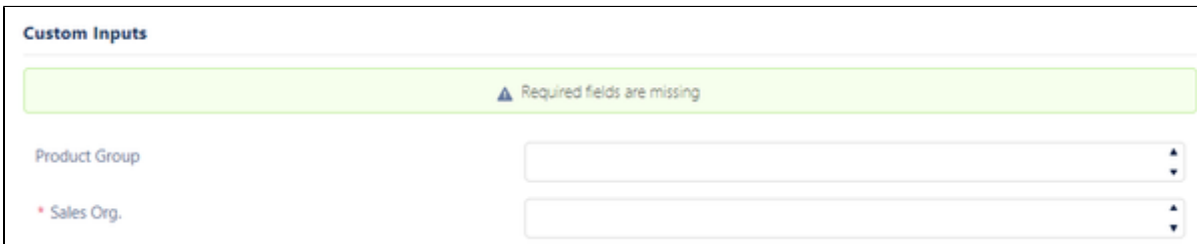


The screenshot shows a form titled "Discount". It contains two input fields: "Discount Type" with a dropdown menu currently showing "Additional Discount", and "Additional Discount" with a text input field.

2. Default Discount % - Allows to define a default discount % for line items; it can be overridden on the line item level.
3. Flat Discount % - Allows to define a discount % which is applied to all line items, overriding discounts on the line item level.

Additional Input Fields

Input fields for the quote header level can be configured without changing the code. These fields can be also applied as parameters for filters used in the customer/product selection screens.




The screenshot shows a form titled "Custom Inputs". At the top, there is a green error message: "Required fields are missing". Below this, there are two input fields: "Product Group" and "Sales Org.", both with dropdown menus.

You can configure this manually in the [CPQInputConfiguration PP](#) or using [CPQ Configuration Wizards](#) with *Name, Label, Input Type, Source Table, Source Type, Source Field, Value Options* (if there is no source field) , *Default Value, Read Only, Required, Level (Header/Line), Customer Filter, Product Filter, Value Options Filtered By Customer, Value Options Filtered By Product*. If the input values are set to be applied to the customer/product filter, the logic will get the value of these inputs and pass them to the filter logic using `filterFormulaParam`.

i You need to create your own filter logic for the customer/product filter and build your filter based on the custom input value provided from `filterFormulaParam`. From version 1.2, we provide the default implementation of customer/product filters for `Source Type = P/C` and `Input Type = OPTION/OPTIONS`.

Calculated Results

In the Calculation Results, there are two sections: Pricing Detail and Customer History.

Calculation Results	
Pricing Detail	
Currency	EUR
Sales Overview	Show
Revenue	1,200.00
Cost	6.66
Net Price	1,182.00
Margin	1,175.34
Margin %	0.00%
Quantity	12
Total Deal Score %	750.75%
Total Deal Score	
Customer History	
Historical Period	14 months (29/02/2020 - 29/04/2021)
Customer Revenue	542,406.53
Customer Quantity	187,038
Customer Margin	278,307.95
Customer Margin %	69.71%

Pricing Detail

- Currency - Displays customer currency (the default is based on *defaultCurrency* in the CPQConfiguration PP)
- Sales Overview
 - If some products were purchased by the customer in last x months, the table can show the following KPIs and allows to compare the currently quoted price and margin:
 - X-months Revenue
 - X-months Volume
 - X-months Margin %
 - X-months Weighted Price Per Unit = $\text{sum}(\text{Invoice Price}) / \text{sum}(\text{Quantity})$ (if Invoice Price is multiplied by Quantity)
 - Quoted Price Per Unit (from the current quote)
 - Quoted Margin % (from the current quote)

SKU	16-months Revenue	16-months Volume	16-months Margin %	16-months Weighted Price Per Unit	Quoted Price Per Unit	Quoted Margin %
<input type="checkbox"/> MB-0006	24,627.55	9,668	52.02%	2.55	27.00	57.17%
<input type="checkbox"/> MB-0001	6,940.40	2,814	70.60%	2.47	31.04	NaN
<input type="checkbox"/> MB-0002	24,170.92	9,032	51.44%	2.68	222.22	NaN
<input type="checkbox"/> MB-0003	24,014.27	9,731	52.06%	2.47	46.52	NaN
<input type="checkbox"/> MB-0004	25,078.56	9,948	52.15%	2.52	1.49	NaN

Showing 1 to 5 of 5 entries

- Revenue
- Cost
- Net Price
- Margin %

- Margin
- Quantity
- Total Deal Score % = $\text{Sum}(\text{Invoice Price} * \text{Quantity}) / \text{Sum}(\text{Target Price} * \text{Quantity})$
- Total Deal Score

Customer History

If the historical sales data are available and you configure *historicalPeriod* in the CPQConfiguration PP (the default is 12 months), it shows summary data for the time period defined during the installation process. The following KPIs are available:

- Historical Period
- Customer Revenue
- Customer Quantity
- Customer Margin
- Customer Margin %

Line Item Level

Label	Product Id	Quantity	Price	Discount %	Quantity	Price	List Price	Invoice Price	Discount Amount	Discount %	Cash Discount	Net Price	Cost	Margin %	Total Line Value
New Quote															
<input checked="" type="checkbox"/>	Meatball BS	100			100	0.00	10.00	10.00	0.00	0.00%	0.00	10.00	0.00	100.00%	1,000.00
<input type="checkbox"/>	Meatball BM				0	0.00	0.00	0.00	0.00	0.00%	0.00	0.00	0.00	0.00%	0.00
<input type="checkbox"/>	Meatball BI				0	0.00	0.00	0.00	0.00	0.00%	0.00	0.00	0.00	0.00%	0.00
<input type="checkbox"/>	Meatball PS				0	0.00	0.00	0.00	0.00	0.00%	0.00	0.00	0.00	0.00%	0.00
<input type="checkbox"/>	Meatball PM				0	0.00	0.00	0.00	0.00	0.00%	0.00	0.00	0.00	0.00%	0.00

User Inputs

For quoted products the user can define the following inputs:

- Quantity
- Price
- Discount %

i If there are values in both fields (Price and Discount %), the priority level depends on your configuration *priceVsDiscountPctInputPriority* in the CPQConfiguration PP.

Input Parameters + -

Quantity	Price
<input type="text"/>	<input type="text"/>
Discount %	
<input type="text"/>	

Calculated Results

Pricing Details

This section provides detailed information about pricing of the item: it starts with a List Price and shows the different adjustments to pricing leading to the Net Price and Margin values.

List Price is obtained from two sources in this order: Price Lists and Product Extensions.

Pricing Detail	
Quantity	25
List Price	50.63
List Price Source	PL-679
Discount Amount	0.23
Discount %	0.45%
Invoice Price	50.40
Rebate	0.00
Cash Discount %	1.50%
Net Price	49.65
Cost	14.42
Margin	35.23
Margin %	70.96%
Revenue	1,260.11

Previous Price provides information about the last transaction done by the customer and product for the quoted item.

Previous Price	
Label	Result
Unit Price	1.87
Margin %	52.00%
Date	2019-02-01

A product price is queried from a price list (PriceBuilder module) which:

- Has the Approved status (approvalState == APPROVED).
- Is assigned to the quoted customer or not assigned to any customer.
- Has a target date <= quote target date.

If there are multiple price lists found, the one with the latest target date takes priority. A price list assigned to the quoted customer will be used for the lookup first and if the product price is not found, the system will continue to a price list not assigned to any customer. When multiple PLs are assigned to a customer, the Quote will check the priority order. In case of the same priority and customer assigned PL, the first valid result price is used.

From version 2.0, CPQ supports getting product prices from the matrix type. The product price can differ based on secondary keys. By default, CPQ uses volume breakdown configured by Price Setting Accelerator ([Volume Breakdown](#)) to look up the price. For details see [Matrix Price List Lookup](#).

If the price list is not found, the product price will do a lookup from a product extension (PX) (4 levels):

- Price By Country
- Price By Customer
- Price By Region
- Price By Segment

PX records should meet these conditions:

- The target date is not set or should be <= quote target date.
- The price is set.

If there are multiple records found, the one with the latest target date takes priority. **Level keys on supported must be available on customer master attributes.**

From version 2.0, the List Price Source element is added in Pricing Details. It helps you to know which PL-ID or product extensions PX is used to get a value from and display in Quote. You can turn it on or off in CPQ_Default_Input_Output_Configuration PP.

Discount

Discount Type	Level	Priority	Description
Additional Discount	Header level		Applies to the overall quote invoice price
Flat Discount %	Header level	1	
Discount %	Line Item level	2	
Default Discount %	Header level	3	

Exception Discount and Standard Discount

If we have Exception Discount and Standard Discount:

- Exception Discount takes priority over Standard Discount.
- Will apply together with Flat Discount %, Discount %, Default Discount %.

Exception Discount

It is a predefined PP table named "CPQExceptionDiscount" created to configure an exception discount for quoted products with combination of customer ID and SKU.

Standard Discount

It is a predefined PP table named "CPQStandardDiscount" and there are six keys to configure the standard discount for a product. It is possible to define that a record in the table is valid for any key by using a generic key alias (the default is *) as the key value. The value can be set as an amount discount or % discount.


 The lookup will ignore any discount less than 0.

Discount Amount = List Price - Invoice Price


or

Discount Amount = Discount % * List Price

If Discount Amount > List Price, it will not be applied and then a warning message for this "Discount Amount must be less than List Price" should be shown.

- **Invoice Price** - It is the Price input if set; otherwise it is the difference between List Price and Discount Amount.
- **Rebate** - Rebate data for a customer and product from RebateManager. There are two calculation schemes: current or forecast (the maximum rebate is temporarily not be applied). By default: Forecast. The configuration is [here](#).
 - To calculate a rebate, you need data from Rebate Records (RR) and you need to meet these requirements:
 - Agreement Status == Approved
 - Start Date (RR) <= Effective Date (Quote)
 - End Date (RR) >= Expiry Date (Quote)
 - Applies to Customer Grouping, Customer and Product Grouping, Product
 - Formula:
current = **List Price * CurrentRebate / CurrentBaselineValue**
forecast = **List Price * Forecast Rebate / ForecastBaselineValue**
 - If we have many rebate types applied to the specific combination of Customer and Product, we will sum up all these rebate values.
- **Cash Discount %** - Value from the customer extension "CustomerCashDiscount". (In case of customerId duplication, the first valid looked-up record is used.)
 -  Ignore if Cash Discount % is null, zero, negative.
- **Net Price** = Invoice Price - Rebate - Cash Discount % * Invoice Price
- **Cost** - Value from the product extension "ProductCost". Cost records should have:
 - Target Date attribute not set.
or
 - Target Date <= quoted Target Date
 - Cost should be greater than 0.

If there are multiple records found, the one with the latest target date takes priority.

-  Show the red alert and message "Invalid Cost" if the cost is not provided (null, zero, negative).
- **Margin** = Net Price - Cost (do not show a value if no cost is provided)
- **Margin %** = Margin / Net Price (do not show a value if no cost is provided)
- **Revenue** = Invoice Price * Quantity

Competitor

If available, this section can provide the user with the information about competitors and their pricing. It shows Median Competitor Price, Top 5 Competitors and their prices and the lowest Competitor Price.

Competitor	
Competitor Median	999.50
Top 5 Competitors	Show
Lowest Competitor Price	10.00

Top 5 Competitors		Export to Excel	Filters
Competitor Name	Price		
<input type="checkbox"/> Meater	10.00		
<input type="checkbox"/> FancyMeat	999.00		
<input type="checkbox"/> TastyMeat	1,000.00		
<input type="checkbox"/> Meatlover	1,200.00		

Showing 1 to 4 of 4 entries

Previous 1 Next

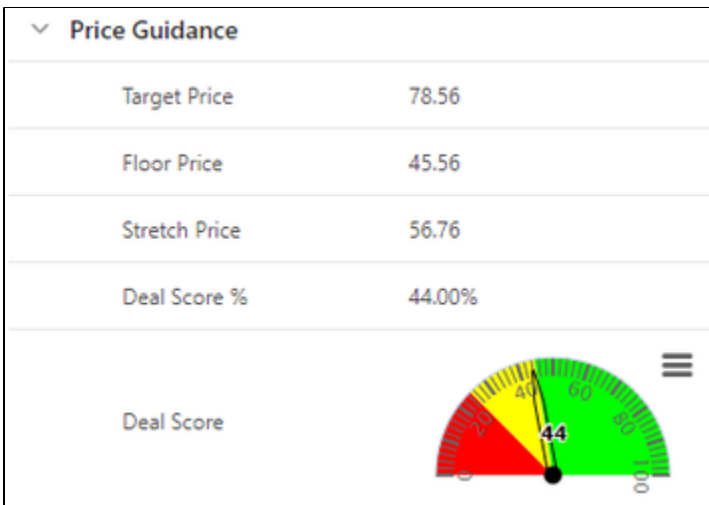
OK

You can get the value from Competition Data, Price Competitors PX or both. Target date (Info date) will be used only if we have the same competitor for a given product, then we get a value based on the latest target date.

i Ignore if the competitor price is zero, null, negative.

Price Guidance

If there are data for price guidance from a PX named "PriceGuidance" (can be also populated by PriceOptimizer), this section provides guidance to the user about Target, Floor and Stretch Prices.



Deal Score % = Invoice Price / Target Price

i Ignore if the one of the values is zero, null, negative.

History

If transactional data is available, this section provides an insight about the size of historical purchases of the product.

Product History	
Historical Period	16 months (03/02/2019 - 03/06/2020)
Product Revenue	1,790,328.01
Product Quantity	531,058
Product Margin	629,310.01
Product Margin %	47.15%
Product Weighted Avg. Price	3.37
Avg. Revenue Per Invoice	167.20
Avg. Quantity Per Invoice	49.59
Avg. Margin % Per Invoice	47.15%

- Historical Period (based on *historicalPeriod* in the CPQConfiguration PP, the default is 12 months)
- Product Revenue
- Product Quantity
- Product Margin
- Product Margin %
- Product Weighted Avg. Price = Sum of Invoice Price / Sum of Quantity (if Invoice Price is multiplied by Quantity)
- Avg. Revenue Per Invoice
- Avg. Quantity Per Invoice
- Avg. Margin % Per Invoice = Sum of Margin / Sum of Net Price

Data Requirements

- Mandatory
 - Price list / Base price (can be linked to the price list in PFX PriceBuilder or it can use the data from a Product Extension)
 - Costs
- Optional
 - Competition Data
 - Customer Cash Discount
 - Transactional Data
 - Price Guidance Data (Target, Floor, Stretch margins)

Setup

You can set up this package manually or through PlatformManager.

CPQ Package Configurations

These configurations can be used to configure how the Quoting Tool works or gets data (e.g. prices, cost, competitors, inputs, outputs, etc.):

- [Core Configuration](#)
- [Inputs Configuration](#)
- [Outputs Configuration](#)
- [Matrix Price List Lookup](#)

Core Configuration

- [Package Configuration Source](#)
- [Historical Data Configuration](#)
- [Product Prices Configuration](#)
- [Customer Mapping Configuration](#)
- [Cost Configuration](#)
- [Price Competitors Configuration](#)
- [Price Guidance Configuration](#)
 - [PriceOptimizer Configurations](#)
 - [Product Extension Configuration](#)
- [Standard Discount Configuration](#)
- [Rebate Configuration](#)
- [Publishing Template Data Configuration](#)
- [Other Configurations](#)

Package Configuration Source

CPQ Package reads configurations from:

- Advanced Configuration with the key named "cpq-default"
- Price Parameter table named "CPQConfiguration"

Those can be different per quote type, see [Configuration Per Quote Type](#) for more details.

Configurations defined in the Price Parameter table will override those in Advanced Configuration.

Historical Data Configuration

Name	Description
historicalPeriod	Optional. Historical period (the default is 12 months).
dm_name	Datamart name
dm_invoice_id_column	Invoice ID column name
dm_invoice_price_column	Invoice price column name

dm_net_price_column	Net price column name
dm_margin_column	Margin column name
dm_pricing_date_column	Pricing date column name
dm_product_id_column	Product ID column name
dm_quantity_column	Quantity column name
dm_per_unit_price_column	Price per unit column. The default is: invoice_price_column / quantity_column.
dm_customer_id_column	Customer ID column name
revenue_FilterField	Revenue filter column It applies to Product Weighted Avg. Price in Quote (Product History result).
revenue_FilterType	Revenue filter type. Supported types: <ul style="list-style-type: none"> • C - customer • ABS - absolute value It applies to Product Weighted Avg. Price in Quote (Product History result).
revenue_FilterValue	Revenue filter value. It can be a Customer attribute to get data or absolute data to filter. It applies to Product Weighted Avg. Price in Quote (Product History result).

Product Prices Configuration

Name	Description
pxProductPriceLookupLevel	Product price lookup level. There are four levels supported: <ul style="list-style-type: none"> • Customer • Region • Country • Segment Multiple levels can be used - separate them by a comma and give priority by their order.

Customer Mapping Configuration

Customer attribute mapping is required to get a product price for each level.

Name	Description
customerCountryAttribute	Customer country attribute
customerRegionAttribute	Customer region attribute
customerSegmentAttribute	Customer segment attribute

customerCurrency	Customer currency
------------------	-------------------

Cost Configuration

Name	Description
pxCostTable	Required if the configuration at quote type level is not defined. The PX name that contains the product cost.
pxCost	Required. The column contains cost data. Default "attribute1".
pxCostCurrency	Optional. The column contains currency data.
pxCostTargetDate	Optional. The column contains valid from date data.
pxCostValidToDate	Optional. The column contains valid to date data.
pxCostFilterKey1 2 3 4 5 6	Optional. Additional dimension to configure the cost. There are maximum six dimensions allowed.
pxCostFilterType1 2 3 4 5 6	Optional. Value type for each dimension. Supported types: <ul style="list-style-type: none"> • P - Product • C - Customer • PX - Product extension • CX - Customer extension
pxCostFilterValue1 2 3 4 5 6	Optional. Value attribute for each dimension.
pxCostFilterValueSource1 2 3 4 5 6	Optional. Name of PX or CX to get the value.

Price Competitors Configuration

Price Competitors can be obtained from both PX and Competition Data.

Name	Description
pxCompetitorDate	Target Date column
pxCompetitorName	Competitor Name column
pxCompetitorPrice	Competitor Price column
pxCompetitorCurrency	Competitor currency column

Price Guidance Configuration

Price Guidance can be obtained from PX and PriceOptimizer.

PriceOptimizer Configurations

Name	Description
poPriceGuidanceModelName	PriceOptimizer model ID for the product price guidance
poPriceGuidanceTargetPrice	Target price column
poPriceGuidanceFloorPrice	Floor price column
poPriceGuidanceStretchPrice	Stretch price column
poPriceGuidanceCurrency	Currency column
poPriceGuidanceSegmentEntryLookupLevel1 2 3 4 5 6 ...	Segmentation name for each level. You need to configure all segmentation levels that the selected model defines.
poPriceGuidanceSegmentEntryLookupTypeLevel1 2 3 4 5 6 ...	<p>The source data type for the segmentation. Supported types:</p> <ul style="list-style-type: none"> • P - Product • C - Customer • PX - Product extension • CX - Customer extension <p>The system will get the data from the source for specific product/customer and search for the segmentation.</p>
poPriceGuidanceSegmentEntryLookupColumnLevel1 2 3 4 5 6 ...	The attribute of the source type that is used to get data for each Segmentation.
poPriceGuidanceSegmentEntryLookupTableLevel1 2 3 4 5 6 ...	Name of PX or CX to get the value.

Product Extension Configuration

Name	Description
pxGuidanceTargetPrice	Target price column
pxGuidanceFloorPrice	Floor price column
pxGuidanceStretchPrice	Stretch price column
pxGuidanceTargetDate	Target date column
pxGuidanceCurrency	Currency column
pxGuidanceFilterKey1 2 3 4 5 6 ...	Optional. The column of price guidance product extension table that is used in the filter.

pxGuidanceFilterType1 2 3 4 5 6 ...	Optional. The source data type. Supported types: <ul style="list-style-type: none"> • P - Product • C - Customer • PX - Product extension • CX - Customer extension
pxGuidanceFilterValue1 2 3 4 5 6 ...	Optional. The attribute of the source type.
pxGuidanceFilterValueSource1 2 3 4 5 6 ...	Optional. Name of PX or CX to get the value.

Standard Discount Configuration

Name	Description
ppStandardDiscountFilterType1 2 3 4 5 6	Value type for each dimension. Supported types: <ul style="list-style-type: none"> • P - Product • C - Customer • PX - Product extension • CX - Customer extension
ppStandardDiscountFilterValue1 2 3 4 5 6	Attribute to get the value using a filter.
ppStandardDiscountFilterValueSource1 2 3 4 5 6	Name of PX or CX to get the value.

Rebate Configuration

Name	Description
rebateCalculation	Rebate calculation plan. Supported values are: <ul style="list-style-type: none"> • Forecast • Current

Publishing Template Data Configuration

Name	Description
publishing_template_docx_quote_{name_use_in_template}	Gets data from the quote object. The value of the config should be the attribute you want to get. E.g. publishing_template_docx_quote_quoteName: uniqueName
publishing_template_docx_customer_{name_use_in_template}	Gets data from the customer object.
publishing_template_docx_lineItem_{name_use_in_template}	Gets data from the lineItem object. The value on the lineItem level can be an input name or output name.

Other Configurations

Name	Description
productFilterLogic	Product filter logic name used to the filter quote products.
customerFilterLogic	Customer logic name used to filter the customers.
customerFilterOnQuoteType	Optional. Default value: default.
productFilterOnQuoteType	Optional. Default value: default.
displayPriceInput	Shows/hides the price input on each line item. Possible values: true false
displayDiscountPercentageInput	Shows/hides the discount % input on each line item. Possible values: true false
priceVsDiscountPctInputPriority	Defines priority of the price input and discount % input. Possible values: priceInputPriority, discountPercentageInputPriority discountPercentageInputPriority, priceInputPriority
predefinedBindingAsScriptProperty	Auto-injected predefined binding as library elements variables. See Predefined Variables for more details.

Currency Conversion Configuration

The Currency Conversion Configuration is used to convert currency based on the customer currency selection in a Quote header. If your data (price, cost, discount...) contains information about currency, it can be used in the CPQ Package. The currency conversion will be applied based on your provided conversion rate.

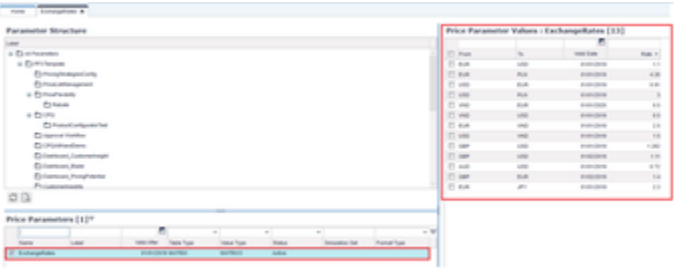
Conversion Rate Lookup

The ccy data source is the default source for your conversion rates. You can also define your custom conversion rates. CPQ supports fetching the conversion rate from a Price Parameters (PP) table. The configuration is:

Name	Description
ppExchangeRate	Name of a Price Parameters table which contains conversion rates
ppExchangeRateFromCurrency	From Currency column
ppExchangeRateToCurrency	To Currency column
ppExchangeRateValidFrom	Valid From column

ppExchangeRateValidTo	Valid To column
ppExchangeRateValue	Column contains rate value

For example, you can define you exchange rates in Price Parameters like this:



Then you can configure to read your rates this way:

Price Parameter Values : CPQ Logic Configuration [6]

Name	Value
ppExchangeRateValidFrom	key3
ppExchangeRateFromCurrency	key1
ppExchangeRateValue	attribute1
ppExchangeRateToCurrency	key2
ppExchangeRateValidTo	attribute2
ppExchangeRate	ExchangeRates

Lookup Details Configuration

Lookup	Name	Description
Default Currency	defaultCurrency	Displays the default currency. It will be used if there is no defined currency.
Price List lookup	priceListItemCurrency	Lookup currency column in Price List
Product Prices lookup	priceByCustomerLevelCurrency	Lookup currency column in PX Price By Customer
	priceByCountryLevelCurrency	Lookup currency column in PX Price By Country
	priceByRegionLevelCurrency	Lookup currency column in PX Price By Region
	priceBySegmentLevelCurrency	Lookup currency column in PX Price By Segment
Exception Discount PP	Attribute4	Currency read from predefined column
Standard Discount PP		

	ppStandardDiscountCurrency	Lookup currency column in PP CPQStandardDiscount
Cost lookup	pxCostCurrency	Lookup currency column in PX cost data
Price Competitors PX	pxCompetitorCurrency	Lookup currency column in PX Price Competitors
Competitor Data	Currency	PCOMP - Currency read from a predefined column
Price Guidance PX	pxGuidanceCurrency	Lookup currency column in PX Price Guidance
Price Guidance from Price Optimizer model	poPriceGuidanceCurrency	Lookup currency column in Price Optimization model
Query Currency in DM	dm_query_currency	<p>By default, when using a conversion rate from the ccy data source, currency conversion for historical data will be done by setting the currency option for the query.</p> <p>If your Datamart is not set up for currency, the conversion will not be applied.</p> <p>If you set this configuration to 'false', the conversion will always be applied in the Groovy code, so the problem can be fixed.</p>

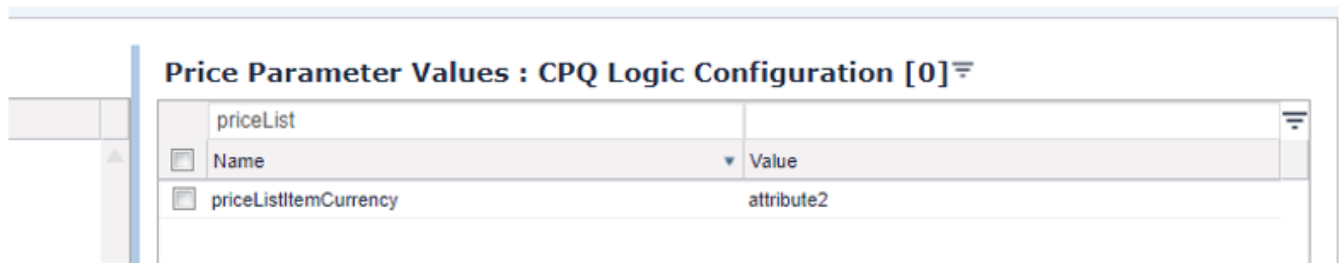
i For Product Prices Lookup, it requires you to change the attribute size when adding the currency column. Please keep in mind that it may lose your data after each deployment because the original attribute size (3) will be deployed again. Please back up first in this case.

How It Works

In the price list item, there is a specified currency, for example, when using a price list that was created by Accelerate Price Setting Package (PSP):



Then you can define the currency of your product's price in CPQ using the 'priceListItemCurrency' configuration:



We use the same definition for other money data used in CPQ, such as Standard Discount, Exception Discount, Price Guidance and Price Competitors, etc.

Sample configuration:

Price Parameter Values : CPQ Logic Configuration [11]

<input type="checkbox"/>	Name	Value
<input type="checkbox"/>	customerCurrency	attribute12
<input type="checkbox"/>	defaultCurrency	GBP
<input type="checkbox"/>	ppExchangeRateFromCurrency	key1
<input type="checkbox"/>	ppExchangeRateToCurrency	key2
<input type="checkbox"/>	priceListItemCurrency	attribute1
<input type="checkbox"/>	pxCostCurrency	attribute4
<input type="checkbox"/>	priceByCustomerLevelCurrency	Currency
<input type="checkbox"/>	priceByCountryLevelCurrency	attribute4
<input type="checkbox"/>	priceByRegionLevelCurrency	attribute4
<input type="checkbox"/>	priceBySegmentLevelCurrency	attribute4
<input type="checkbox"/>	ppStandardDiscountCurrency	attribute4

i If there is no conversion rate found, the value 0 will be displayed for historical data, Price Guidance and <empty> for others.

Note:

- If we have defaultCurrency and do not map the Currency configuration (e.g. Cost, Price Guidance,...), the default currency is used.
- If we do not map both defaultCurrency and Currency configuration (e.g. Cost, Price Guidance), the original value is used without currency conversion.

Inputs Configuration

i The CPQ Inputs Configuration is deprecated from version 2.0. It is recommended to use CPQ_Default_Input_Output_Configuration PP instead.

The CPQ Inputs Configuration is used to create a custom input in the CPQ Package. Custom inputs can be displayed on the header or line item level.

Custom inputs are generated under the input configurator:

- LineItemInputConfigurator - Contains inputs for each line item.
- HeaderInputConfigurator - Contains inputs for the header level.


Details

Name	Description
------	-------------

Name	Name of the input
Label	Input label to display on a quote
Input Type	Type of the input. Supported values are: <ul style="list-style-type: none"> • USERENTRY - normal user input • STRINGUSERENTRY - string user input • INTEGERUSERENTRY - integer user input • BOOLEANUSERENTRY - Boolean user input • DATEUSERENTRY - date user input • TIMEUSERENTRY - time user input • DATETIMEUSERENTRY - datetime user input • OPTION - single option selection • OPTIONS - multiple options selection • RADIO - radio input • BOOLEAN - Boolean input
Source Table	Used if the input type is OPTION or OPTIONS. Defines a table name where to get the values options.
Source Type	Used if the input type is OPTION or OPTIONS. Defines a table type (P, C, PX, CX, PP) where to get the values options.
Source Field	Used if the input type is OPTION or OPTIONS. Defines a column where to get the values options.
Value Options Filter By Customer	Defines if the selected customer will be used to filter value options from the source query.
Value Options Filter By Product	Defines if the selected product will be used to filter value options from the source query.
Value Options	Used if the input type is OPTION or OPTIONS and the source is not defined. Defines the value options, separated by comma.
Default Value	Defines an input default value.
Value Hint	Defines an input value hint.
Read Only	Defines if the input is read only.
Required	Defines if the input is required.
Level	Defines where to display the input. Supported values are: Header, Line.
Customer Filter	Defines if the input value can be passed to a customer filter logic as a parameter.
Product Filter	Defines if the input value can be passed to a product filter logic as a parameter.
Customer Filter Operator	Provides the operator for Customer Filter. Supported values are: <ul style="list-style-type: none"> • OP_EQUAL (default for other supported input types)

	<ul style="list-style-type: none"> • OP_NOT_EQUAL • OP_LESS_THAN • OP_GREATER_THAN • OP_LESS_OR_EQUAL • OP_GREATER_OR_EQUAL • OP_LIKE • OP_ILIKE • OP_IN (default for input type is OPTIONS) • OP_NOT_IN
Product Filter Operator	<p>Supports the operator for Product Filter. Supported values are:</p> <ul style="list-style-type: none"> • OP_EQUAL (default for other supported input types) • OP_NOT_EQUAL • OP_LESS_THAN • OP_GREATER_THAN • OP_LESS_OR_EQUAL • OP_GREATER_OR_EQUAL • OP_LIKE • OP_ILIKE • OP_IN (default for input type is OPTIONS) • OP_NOT_IN

Outputs Configuration

 This CPQ Outputs Configuration is obsolete from version 2.0.

The CPQ Outputs Configuration is used to define configurations for a CPQ output.

Details

Name	Description
Name	Name of the output to configure.
User group	Defines a user group for the output.

Configuration Wizard

Configuration wizard makes the output configuration easier by providing a user interface for selecting an output and configurations. The CPQ Package defines a logic for the configuration wizard and configuration wizard executor. You need to define the configuration wizard in Administration > Configuration Wizards Admin where you add a new wizard. After the wizard is created, you can use it to set up the output configuration:

The wizard interface helps create/update/delete output configurations:

Options:

- **Quote Type** - If the CPQ logic is provided under another quote type rather than the default one, you need to select the correct quote type.
- **User Group** - Select a user group to apply.
- **Output Configurations** - List of configured outputs.
 - **Name** - Name of the CPQ output (a dropdown where you can select).
 - **User Group** - If you need to define a user group for that particular output.
 - **User Group Selection** - Defines if you want to merge or override the user group selected above.
 - **Merge** - Merges the user groups from the User Group input and the one defined in the row.
 - **Override** - Skips the user group defined in the row and uses the values from the User Group input.
 - **Leave empty** - Does nothing, uses the value in the row.
 - **Delete** - Defines if you want to delete the configurations.

Matrix Price List Lookup

In a Matrix Price List, the product price can differ based on secondary keys.

You need to define a logic (we call it "keys provider") that specifies which secondary keys you want to use for price lookup. Your logic can be configured by setting 'matrixPriceListSecondaryKeyProvider' in CPQ configurations.

Your keys provider must be a Groovy library and accept a list of product IDs as parameter. We expect it to return a map of desired secondary keys per product.

You can also access the current quote view in your provider by using 'quote' variable, e.g. quote.uniqueName.

```
Map mySecondaryKeys(List skus){
    def mySkus = skus[0]
    api.logInfo("quote-${quote.uniqueName}", mySkus)
    return [(mySkus): ["key1", "key2"]]
}
```

Keys provider runs in the quote pre-phase context.

If you use Price Setting Package to create the matrix price list, there is a predefined volumes breakdown key provider that can be used.

CPQ Package Version 2.0

- [Design](#)
- [Configuration Per Quote Type](#)
- [Custom Logics in CPQ](#)
- [Error Handler](#)

Design

The main requirement of the CPQ Package version 2.0 is to allow custom code from the library to be executed by CPQ logics.

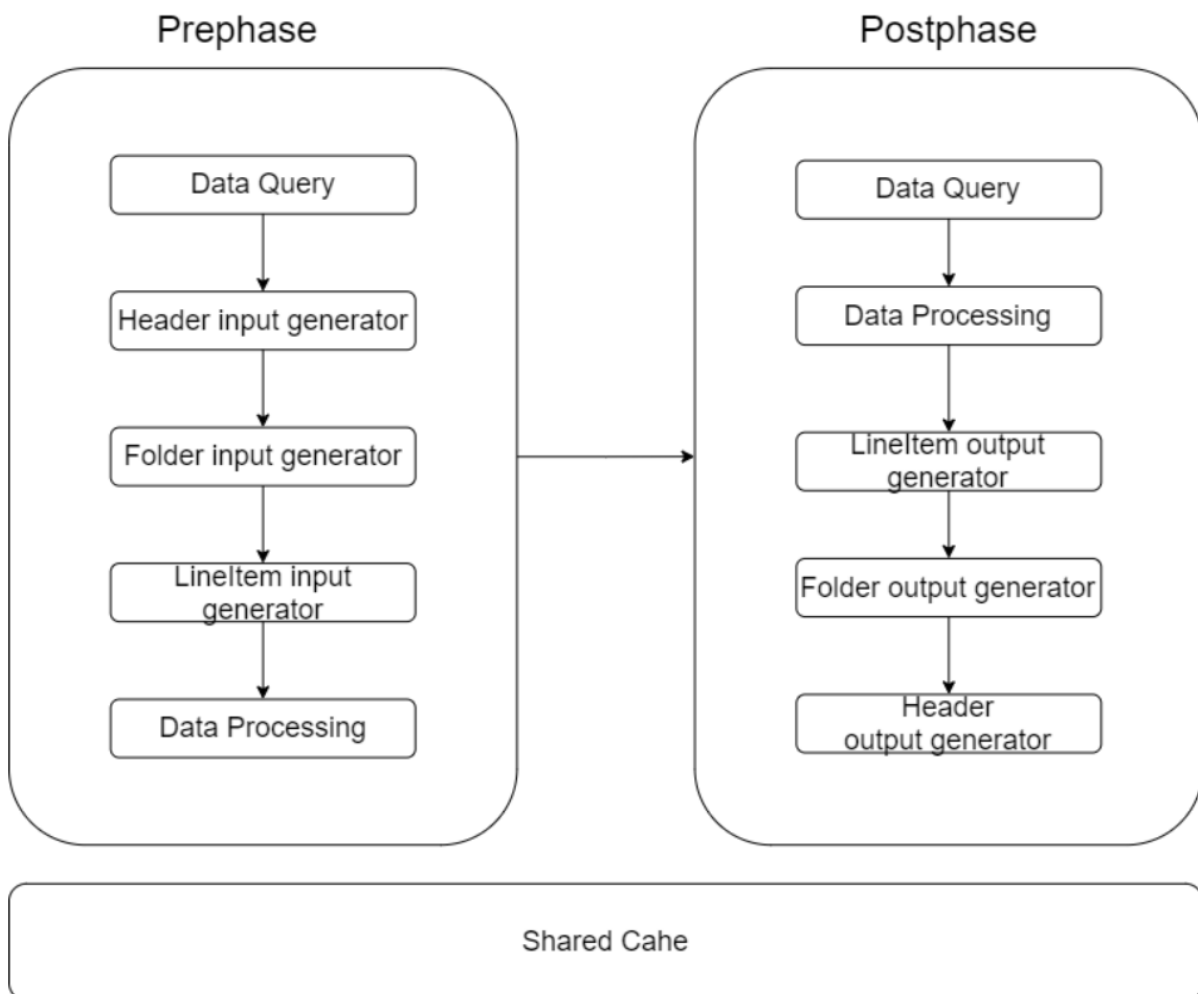
In this section:

- [Overview](#)
- [Per Quote Type Configuration](#)
- [Task Runner](#)
- [Data Sharing](#)

Overview

CPQ Package version 2.0 has the following functionality:

- Allows you to run custom libraries on the quote header. The libraries can run both in pre-phase and post-phase.
- There are **task runners** for data query, data processing, input, and output generator
- Allows you to set the priority of tasks for each runner. In addition, some tasks can depend on the results of other tasks.
- Existing logics are split into multiple modules and plugins as CPQ default configurations.



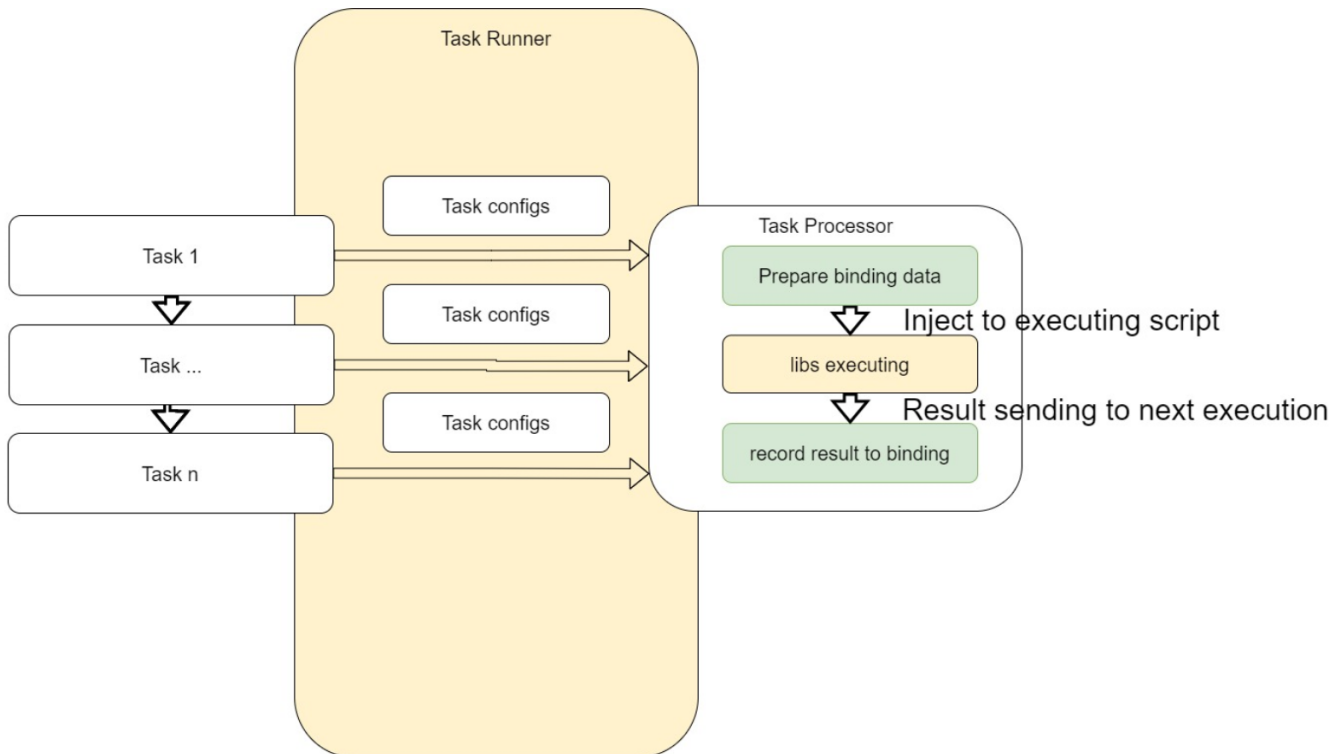
Per Quote Type Configuration

Each quote type defines a name of the PP table which is the main entry for other configurations; the table contains other configurations:

- **Configurations** - Define the PP table which contains core configurations used by CPQ.
- **Processing_Configurations**- Define the PP table which contains configurations for data query and data processing.
- **Input_Output_Configurations** - Define the PP table which contains configurations for inputs and outputs generator.

Task Runner

Task Runner is an element in the header and line item logic that allows running the custom library and handles data sharing between a task and other Task Runners.



Tasks define the library path that contains the logic for the execution.

Task Processor is a library that is responsible for running the library defined in the task, recording the result, and sharing it with other tasks.

Data Sharing

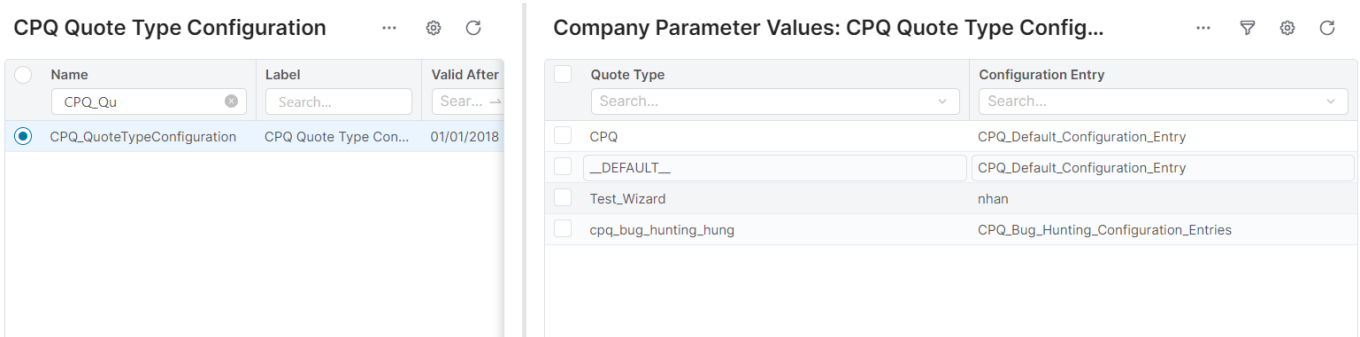
Data are shared from pre-phase to post-phase using the shared cache.

Configuration Per Quote Type

Price Parameter Configuration

There is a price parameter table **CPQ_QuoteTypeConfiguration** dedicated to the quote type configuration setting.

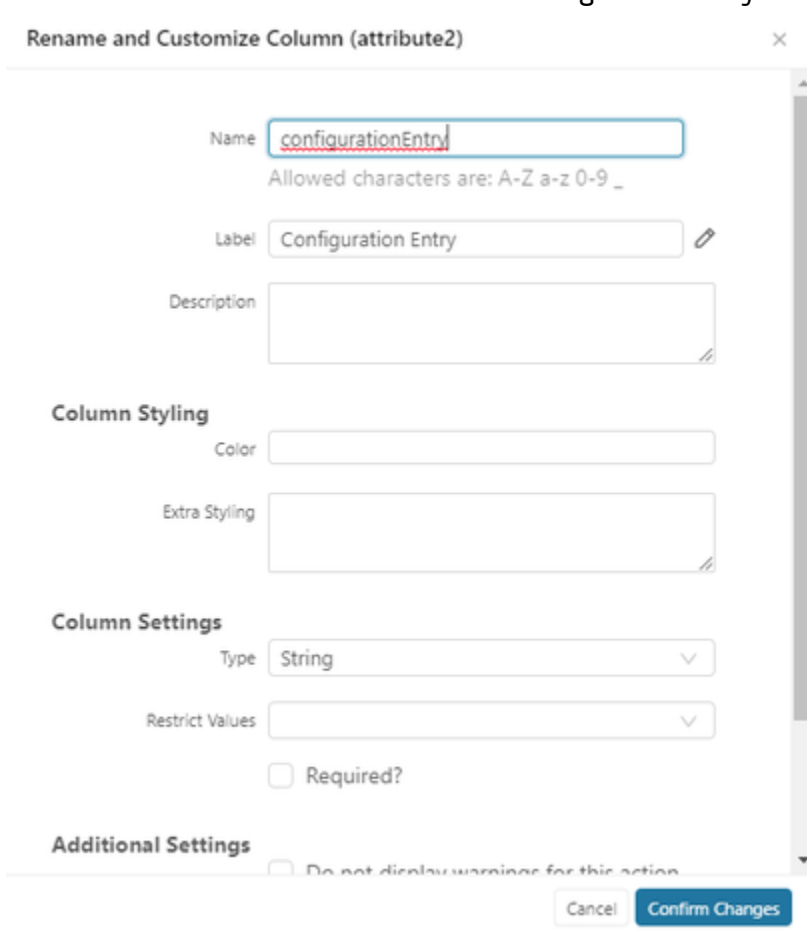
i The configurations in the price parameter take priority when both PP and Quote Types configurations in the Quoting module are present.



Quote Type Setting

To create a per quote type configuration:

1. Open Quote Types.
2. Rename a column and set the name to "configurationEntry".



3. Set the name of the PP table which contains the configuration entries.

Configuration Entry

Search...

CPQ_Default_Configuration_Entry

Configuration Entry PP Structure

Name	Label	Table Type	Value Type	Valid After
configuration_entry	Search...	Select Value	Select Value	Search...
CPQ_Default_Configuration_Entry		SIMPLE	STRING	1/1/2000

Price Parameter Values: CPQ_Default_Configuration_Entry

Export Add Record

Name	Value
Search...	Search...
<input checked="" type="checkbox"/> Configurations	CPQConfiguration
<input type="checkbox"/> Processing Configurations	CPQ_Default_Header_Configuration
<input type="checkbox"/> Input Configurations	CPQ_Default_Input_Configuration
<input type="checkbox"/> Output Configurations	CPQ_Default_Output_Configuration
<input type="checkbox"/> Error Configurations	CPQV2.0_Error_Configurations

Configuration entry setting:

- **Advanced_Configuration_Key** - Advanced property (AP) key which defines AP keys from which to create the configuration.
- **Configurations** - PP table name which contains CPQ configurations.
- **Processing_Configurations** - Defines the PP table which contains configurations for data query and data processing.
- **Input_Output_Configurations** - Defines the PP table which contains configurations for inputs and outputs generator.
- **Error_Configurations** - PP table which contains configurations for errors.
- **Historical_Datamart_Name** - Name of a Datamart which contains historical data.
- **Price_By_Country_PX_Name** - PX table which contains price by country values.
- **Price_By_Customer_PX_Name** - PX table which contains price by customer value.
- **Price_By_Region_PX_Name** - PX table which contains price by region values.
- **Price_By_Segment_PX_Name** - PX table which contains price by segmentation values.
- **Exception_Discount_PP_Name** - PP table which contains exception discount values.
- **Standard_Discount_PP_Name** - PP table which contains standard discount values.
- **Customer_Cash_Discount_CX_Name** - CX table which contains customer cash discount values.
- **Product_Cost_PX_Name** - PX table which contains product costs.
- **Competitor_PX_Name** - PX table which contains competitor prices.
- **Price_Guidance_PX_Name** - PX table which contains price by segmentation values.
- **Input_Configuration_PP_Name** - PP table which contains custom input configurations in version 1.0 (deprecated from version 2.0).

From now, the system knows where (from which tables) to get configurations and data for your Quote Type.

i If the configuration is not provided for the specific quote type, the default one is used. See [CPQ Package Architecture](#).

Custom Logics in CPQ

You can add your custom calculations (at header and line item level) to CPQ without modifying the core logic/library.

There are two PP tables defining the structure of the quote logic which can be configured in the configuration entry Price Parameter table:

- **Processing_Configurations** - PP table which contains header elements configurations. The default table name is **CPQ_Default_Header_Configuration**.
- **Input_Output_Configurations** - PP table which contains line item elements configurations. The default table name is **CPQ_Default_Input_Output_Configuration**.

Your custom logic should be written as a library so that it can be configured and run by CPQ.

If you want to use a custom PP table, ensure that its structure is the same as defined in default PP tables.

Further in this section:

- [Custom Header Logic](#)
- [Custom Input & Output logic](#)
- [Predefined Variables](#)

Custom Header Logic

This Price Parameter table defines custom logics that can be run on the header level.

Default Price Parameter table: **CPQ_Default_Header_Configuration**

In this section:

- [Configurations](#)
- [Custom Logic](#)

Configurations

You can set the following options:

- **Name** - Name of your element. This can be used to access returned data from another element.
- **Processor** - Defines an element in the CPQ header to run your element. There are 2 processors available: **Data Query** and **Data Processing**.
- **Running Phase** - Select here the phase in the header to run your element. Can be pre-phase, post-phase or both.
- **Library Name** - Name of the library which contains the calculation for your element.
- **Element** - Element from the library which contains the calculation for your element.
- **Function** - Function name from the library element which contains the calculation for your element.
- **Skip** - Skips executing your element. Can be Yes/No.
- **Priority** - Defines priority of running within the processor.

- **Data Shared to Post-phase** - Defines if your element result can be accessed from the post-phase. This can be shared only when your running phase is pre-phase. Data is shared between phases using the shared cache.

The screenshot shows a dialog box titled "Add New Price Parameter Value" with a close button (X) in the top right corner. The dialog contains the following fields:

- Processor:** A dropdown menu with the value "10".
- Running Phase:** A dropdown menu with the value "PrePhase/PostPhase".
- Library Name:** A text input field containing "Your_Library_Name" and a clear button (X).
- Element:** A text input field containing "Element_To_Run_On_Your_Library" and a clear button (X).
- Function:** A text input field containing "Function_To_Run_From_The_Element" and a clear button (X).
- Skip:** A dropdown menu with the value "No".
- Priority:** A text input field containing "1".
- Data Shared To PostPhase:** A dropdown menu with the value "Yes".

At the bottom right of the dialog, there are two buttons: "Cancel" and "Add".

Custom Logic

Your logics should be defined in the Pricefx Groovy library.

You need to provide the "library name", "element name", and "function name" of your library.

Your function will run in the quote header context.

There are predefined objects that you can use in your function:

- **elements** - Map of processed elements value.
- **quoteProcessor** - Pricefx predefined object, this is auto-binding to your function.

Your function should not take any arguments and should return the value if you need to access the result from other elements.

Custom Input & Output logic

This Price Parameter table defines custom logics that can be run on the line item level.

Default Price Parameter table: **CPQ_Default_Input_Output_Configuration**

In this section:

- [Configurations](#)
- [Custom Logic](#)

Configurations

You can set the following options:

- Generation Type - Available options are:
 - Header Input
 - Header Output
 - Line Item Input
 - Line Item Output
 - Folder Input
 - Folder Output
- Name - Element name, will be used as the name of the generated output.
- Label - Element label, will be used as the label of the generated output.
- Input Type - The type of input to generate.
- Output Type - The type of output to generate.
- Processor - Select here an element on the CPQ line item level to run your element. There are 30 elements out of the box that can be configured to run your logic.
- Library Name - Name of the library which contains the calculation for your element.
- Element - Element from the library which contains the calculation for your element.
- Function - Function from the library element which contains the calculation for your element.
- Library runner - Defines in which phase the library should be run. The options are: Input, Output, or Both.
- Skip
- Priority
- Result group
- Format type

Add New Price Parameter Value ×

* Name

Library

Element

Function

Label

Label Translations

Display

Result Group

User Group

Format Type

Result Type

Result Description

CSS Property

Override Allow Empty

Overridable

Overridden

Override Value Options

Suffix

Warnings

Critical Alert Condition

Critical Alert Message

Red Alert Condition

Red Alert Message

Yellow Alert Condition

Custom Logic

Your logics should be defined in the Pricefx Groovy library.

You need to provide the "library name", "element name", and "function name" of your library.

There are predefined objects that you can use in your function:

- **elements** - Map of processed values.
- **header** - Map of processed values in header processing.

Your function should not take any arguments and should return the value of your output.

Predefined Variables

There are some predefined variables that are already initialized and shared across custom logic.

- **quoteProcessor** - Pricefx predefined processor available in the quote header context and already bound to your custom library as a predefined variable. It can be accessed from all contexts.
- **elements** - Map containing previous logic results. It can be accessed from all contexts.
- **header** - Map containing header processed results. This variable can be accessed in the inputs and outputs generation context.
- **lineItem** - Current processing line item. This variable can be accessed in the inputs and outputs generation context.
- **inputs** - Current processing line item input values. This variable can be accessed in the inputs and outputs generation context.
- **inputConfig** - Current processing input configuration. We can use this to update the current input config. This variable can be accessed in the inputs generation context.
- **outputs** - Current processing line item outputs values. This variable can be accessed in the inputs and outputs generation context.
- **outputConfig** - Current processing output configuration. We can use this to update the current output config. This variable can be accessed in the outputs generation context.

The predefined variable can be accessed either by its name or TaskProcessor.

- name

i From version 2.0.3, this is only available when `predefinedBindingAsScriptProperty` configuration is enabled.

```
function myElement(){
    // previous element name from PP table is "myCustomQuantity"
    def quantity = elements.myCustomQuantity

    return quantity
}
```

- TaskProcessor

```
function myElement(){
    // previous element name from PP table is "myCustomQuantity"
```

```

    Map elements = libs.CPQ_SharedLib.TaskProcessor.getProcessingBinding
("elements")
    def quantity = elements.myCustomQuantity

    return quantity
}

```

The predefined variables are automatically injected into the custom library. This can slow down the process. In version 2.0.3, a new configuration named `predefinedBindingAsScriptProperty` is added and enabled as default. So you can decide whether to inject those variables automatically or not.

If it is disabled, you cannot use predefined bindings directly; a quick fix is to wrap the method implementation under `libs.CPQ_SharedLib.TaskProcessor.runWithPredefinedBindings`.

```

// When 'predefinedBindingAsScriptProperty' is enabled
BigDecimal execute() {
    boolean isPriceInput = libs.CPQ_Output_Utils.Common.
isPriceInput()
    // 'header' is predefined binding. You can access it directly
in custom library as it is injected automatically
    BigDecimal priceFromLookup = header.ProductPrices?.getAt
(lineItem.sku)?.resultPrice

    return isPriceInput ? BigDecimal.ZERO : priceFromLookup
}

// When 'predefinedBindingAsScriptProperty' is disabled
BigDecimal execute() {
    // 'header' is predefined binding. You cannot access it directly in
custom library as it is not injected automatically.
    // A quick fix is to wrap the implementation under "libs.
CPQ_SharedLib.TaskProcessor.runWithPredefinedBindings"
    return libs.CPQ_SharedLib.TaskProcessor.runWithPredefinedBindings
{
        boolean isPriceInput = libs.CPQ_Output_Utils.Common.
isPriceInput()
        BigDecimal priceFromLookup = header.ProductPrices?.getAt
(lineItem.sku)?.resultPrice

        return isPriceInput ? BigDecimal.ZERO : priceFromLookup
    }
}

```

Error Handler

Starting with version 2.0, there is a simple error handler for configuring error levels and messages.

The error configurations can be configured in a PP table defined in [Configuration Per Quote Type](#).

The default PP table for error handler named **CPQ_Default_Error_Configuration** will be used if the configuration is not specified on the quote type level.

Price Parameter Values: CPQ Default Error Configuration

Name	Message	Level	Error Handler	Abort Handler Execution
<input type="checkbox"/> COLECTION_UTILS_COMPARATOR	Comparator not provided	ERROR		No
<input type="checkbox"/> HLU_CAST_VALUE_FOLLOW_FILTER	Revenue Filter Value in CPQ Historical Data...	ERROR		No
<input type="checkbox"/> LIBRUNNER_INCORRECT_PATH	Incorrect library path	ERROR		No
<input type="checkbox"/> LIBRUNNER_LIB_NOT_EXIST	Library not exist	ERROR		No
<input type="checkbox"/> LOOKUP_UTILS_GROUP_RESULT	LookupUtils: Invalid Grouping Configuration	ERROR		No
<input type="checkbox"/> LOOKUP_UTILS_PREPARE_LOOKUP_VALUE	LookupUtils: Invalid Filter Value Configured	ERROR		No
<input type="checkbox"/> CUSTOMER_INVALID	Customer not set	WARNING		
<input type="checkbox"/> REBATE_PLAN_IS_NOT_SUPPORTED	Rebate calculation plan not supported	WARNING		No

Usage

In your logic, you just need to throw an exception with the key configured in the PP table.

Then you can use the PP table to configure the level, error message, etc.

```
def yourLogic(){
  try{
    // your logics
  }catch(e){
    api.throwException("ERROR_CONFIG_KEY")
  }
}
```

CPQ Configuration Wizard

The CPQ Configuration Wizard makes the package configuration easier by providing a user-friendly interface for the setup (instead of manual changes in Price Parameters).

Configuration Wizards Admin

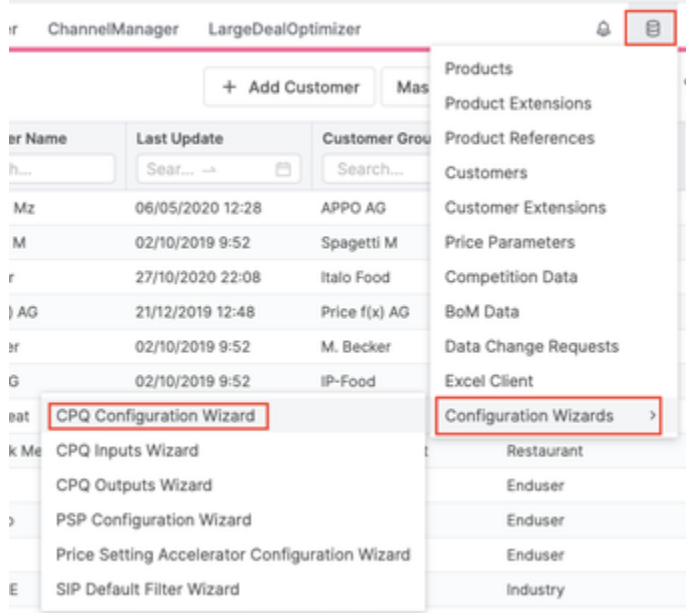
In the CPQ Package, we defined a default logic for the configuration wizard and wizard executor.

Go to Configuration Configuration Wizards Configuration Wizards Admin:

Configuration Wizards Admin

Name	Label	Wizard Formula (Configurator)	Execution Formula	User Group
<input type="checkbox"/> CPQ_Configuration_Wizard	CPQ Configuration Wizard	CPQ Configuration Wizard	CPQ Configuration Wizard Executor	

The Configuration Wizards Admin is already configured after deployment, you can use it for the configuration. Go to Master Data Configuration Wizards select CPQ Configuration Wizard.



CPQ Configuration Wizard

Quoting Tool first to get familiar with the package and the CPQ Package Configurations to understand how it is configured properly.

Step 1: Configure Your Quote Type

Step 2: Configure Your Quote Structure

Step 3: Configure Your Quote Default Modules

Step 1: Configure Your Quote Type

Allows you to select which quote type is configured, as well as the configuration entry and configuration table.

CPQ Configuration Wizard

Configuration Entry

Configuring Quote Type - Default

The [Configuration Entry](#) helps you to configure each quote type that reads data and configurations from different advanced configuration keys and tables (PP, PX, CX). This way you can control the behaviour of different types of quotes.

Select Your Entry Table

Entry Table is price parameter table that stores all table names using by your quote type.

CPQ_Default_Configuration_Entry

Select Your Configuration Table

[Configuration Table](#) is price parameter table that stores all configurations using by your quote type.

CPQConfiguration

Back

Step 2: Configure Your Quote Structure

Allows you to configure quote structures including input/outputs, header processing, and errors configuration.

CPQ Configuration Wizard

Quoting Tool first to get familiar with the package and the [CPQ Package Configurations](#) to understand how it is configured properly.

Step 1: Configure Your Quote Type

Configure

Step 2: Configure Your Quote Structure

- Error Handler Configuration
- Header Processing Configuration
- Input Output Configuration

Step 3: Configure Your Quote Default Modules

Configure

Step 3: Configure Your Quote Default Modules

This step allows you to configure the core modules of CPQ:

- Cost
- Currency
- Customer Mapping
- Historical Data
- Price Competitors
- Price Configuration
- Price Guidance
- Publishing Templates
- Rebate

- Discounts
- Other Configurations

CPQ Configuration Wizard

Quoting Tool first to get familiar with the package and the CPQ Package Configurations to understand how it is configured properly.

Step 1: Configure Your Quote Type

__DEFAULT__

- Cost
- Currency
- Customer Mapping
- Historical Data
- Price Competitors
- Price Configuration
- Price Guidance
- Publishing Templates

|

Configure

i For details see:

- [CPQ Package Configurations](#)
- [Version 2.0](#)

Note:

- There is a known issue when you input the value in the last row as below. You should press the Tab key or click outside this row, then the value is saved and applied.

CPQ Configuration Wizard

Publishing Templates

The [Publishing Templates](#) wizard helps you to get data from quote, customer or line item. It is used to export the PDF file for a Quote.

Value Getting From
Customer

Attribute To Get Value
Customer Group

Name Used In Template
CustomerGroup

Back

Apply

- Product Prices configuration wizard

CPQ Configuration Wizard

Product Prices

The [Product Prices](#) wizard helps you to define the Product Extension (PX) lookup priority if the price list is not found. The Price from the lookup is used in the List Price output element.

Product Prices Lookup Level
Customer x Region x Country x Segment x

Back

In Product Prices Lookup Level, it will apply the priority by the order you selected.

- If you want to delete a configuration for the required fields, you need to delete it manually in Company Parameters (e.g. Cost in CPQConfiguration, Product_Cost_PX_Name in CPQ_Default_Configuration_Entry,...) where it is saved. For this case, the default one is used. See [CPQ Package Architecture](#).

CPQ Inputs Configuration - CPQ Outputs Configuration in Wizard

⚠ They are both obsolete from version 2.0, as they were replaced by [CPQ Configuration Wizard](#).

- [CPQ Inputs Wizard](#)
 - [Details on "Create"](#)
 - [Details on "Edit" and "Delete"](#)
- [CPQ Outputs Wizard](#)

Price Parameters

- CPQInputConfiguration
- CPQOutputConfiguration

Setup

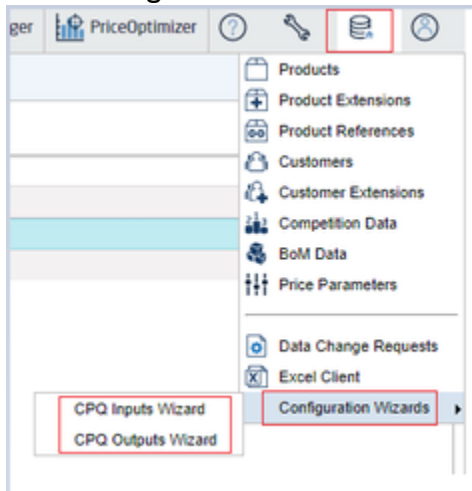
1. In the CPQ Package you need to define a logic for the configuration wizard and wizard executor first. Go to Administration Configuration Wizard Admin and define Name, Label, Wizard Formula (Configurator), Execution Formula (Configurator), Execution Formula.

Configuration Wizards Admin

Name	Label	Wizard Formula (Configurator)	Execution Formula	User Group
CPQOutputConfigurationWizard	CPQ Outputs Wizard	CPQ_Output_Configuration_Wizard	CPQ_Output_Configuration_Wizard_Executor	
CPQInputConfigurationWizard	CPQ Inputs Wizard	CPQ_Input_Configuration_Wizard	CPQ_Input_Configuration_Wizard_Executor	

To provide a better guidance, we created an example label with “CPQ Inputs Wizard” and “CPQ Outputs Wizard”.

2. After the Configuration Wizard Admin is created, you can use it for the configuration. Go to Master Data Configuration Wizards choose which wizard you want to use.

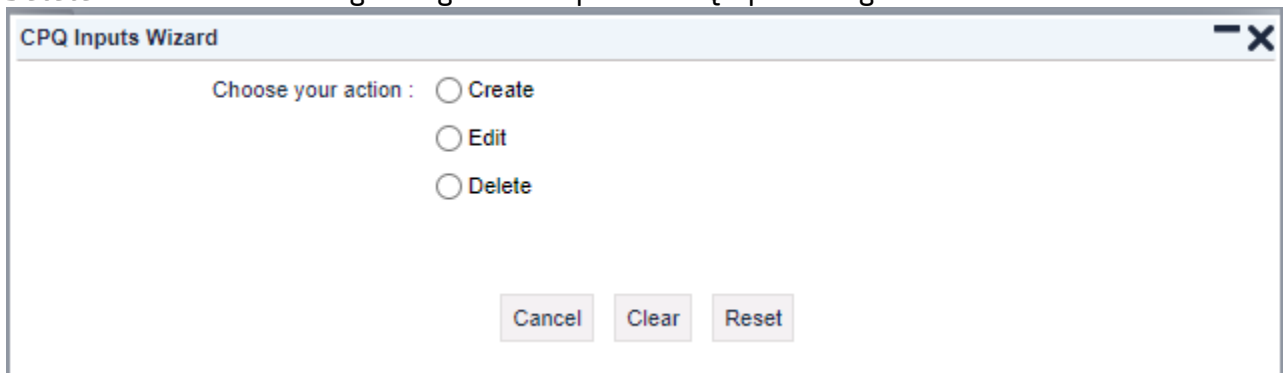


How It Works

CPQ Inputs Wizard

The CPQ Inputs Configuration defines input fields which should be available in the Quote Header and defines if the field is also used in the Customer or Product filter. The CPQ Inputs Wizard provides you three options when you open it:

- **Create** - Creates a new configuration input for CPQInputConfiguration PP. See details below.
- **Edit** - Edits an existing configuration input for CPQInputConfiguration PP. See details below.
- **Detete** - Deletes an existing configuration input for CPQInputConfiguration PP. See details below.



Details on “Create”

CPQ Inputs Wizard

Choose your action : Create
 Edit
 Delete

Name : !

Enter a **Name** (preferably with no spaces, e.g. cClass, pGroup). Then press the **tab** key to apply the setting. If the name already exists, you will be notified.

After the name check, the next configuration is shown (after pressing the tab key): **Required fields**. You need to fill in Name, Label, Level (if you do not fill all required fields, the OK button will not be displayed).

The default configuration:

CPQ Inputs Wizard

Choose your action : Create
 Edit
 Delete

Name :

Label : !

Input Type :

Default Value :

Value Hint :

Read Only :

Required :

Level : !

Customer Filter :

Customer Filter Operator :

Product Filter :

Product Filter Operator :

If you choose OPTION/ OPTIONS as Input Type, some hidden input fields will be displayed:

CPQ Inputs Wizard

Choose your action : Create
 Edit
 Delete

Name : cClass

Label : !

Input Type : OPTION

Source Type :

Source Table :

Source Field :

Value Options :

Value Options Filtered By Customer :

Value Options Filtered By Product :

Default Value :

Value Hint :

Read Only :

Required :

Level : !

Customer Filter :

Customer Filter Operator :

Product Filter :

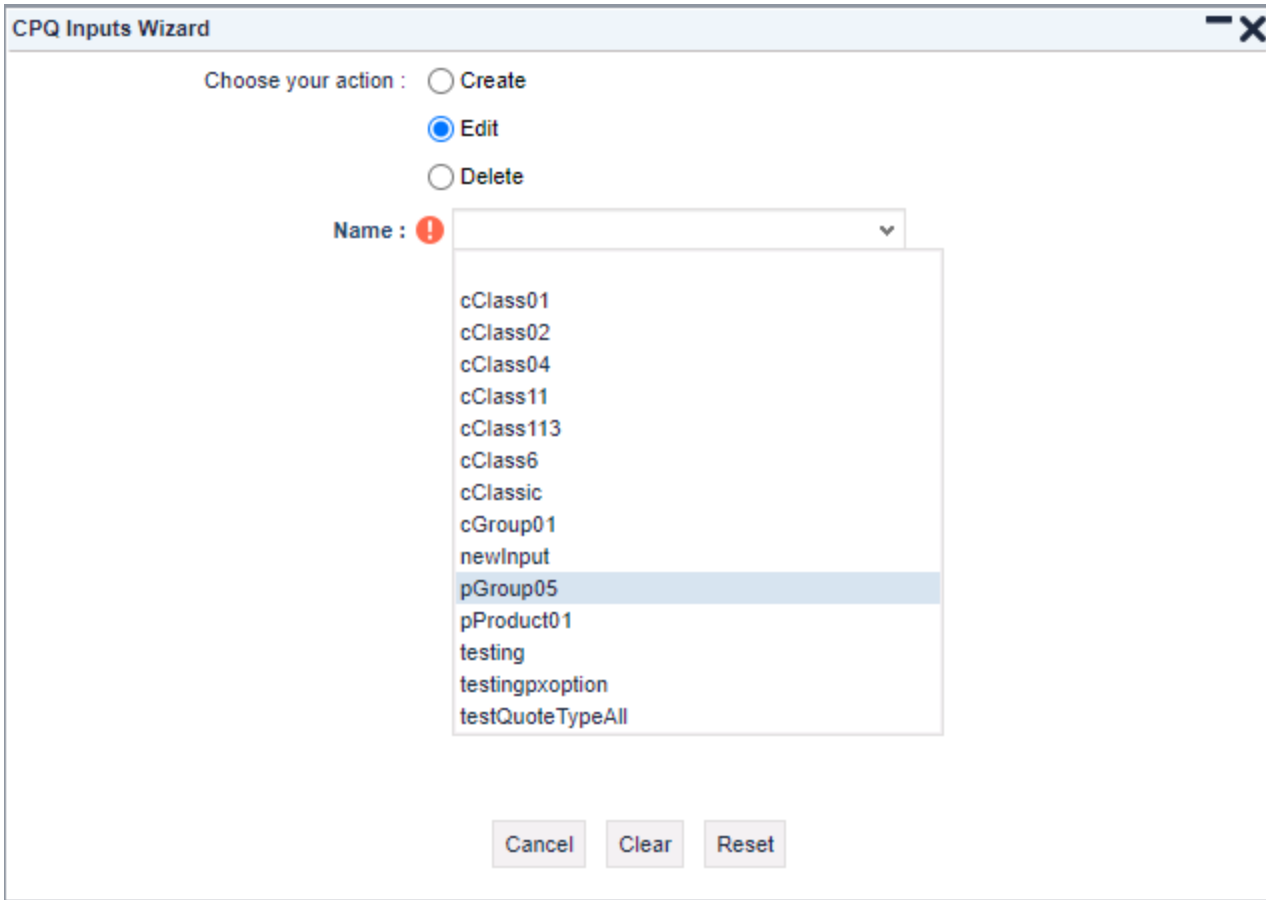
Product Filter Operator :

Cancel Clear Reset

The Source Table (with Source Type: P, C, PX, CX, PP) and Source Field are sorted in the ascending (A-Z) alphabetical order. The displayed lists in the Source Table and Source Field will be 'name', not 'label'. If you do not define a label for fields, the list of Source Fields will not be displayed.

Details on "Edit" and "Delete"

Select the item to edit or delete from the **Name** drop-down list.



After you make your changes and click the OK button, this input configuration will be added automatically into the CPQInputConfiguration PP. You need to click Refresh in the PP table and then the Recalculate button in the Quote to apply it.

Example configurations:

Price Parameter Values : CPQ Inputs Configuration [8]

Name	Label	Input Type	Source Table	Source	Source Field	Value Options	Default Value	Value Hint	Read Only	Required	Level	Customer Filter	Product Filter	Value Options Filtered By Customer	Value Options Filtered By Product	Customer Filter Operator	Product F. ^
cClass04	Customer Type	OPTION	C	C	Customer Type			No	Yes	Header	Yes					OP_NOT_EQUAL	
valueOption01	Sales Org	OPTION	C	C	Sales Org								customerid				
cClass6	Customer Group	OPTIONS	C	C	Customer Type					Header	Yes					OP_IN	
valueOption03	Customer Extension	OPTION	ValueOptionsFilteredByCustomer	CX	Attr_2					Header			attribute1 = attribute3				
pGroup01	Product Group	OPTION	P	P	Product Group					Header		Yes					
pType	Product Class	OPTIONS	P	P	Product Class					Line						sku	
pPP	Product Type	OPTIONS	TestFolderQuoteApproval	PP	folderBeverages					Line						name = attribute4	
pLifeCycle	Product Life Cycle	OPTION	PriceByCountry	PX	Price					Header			attribute1=attribute5				

For more details on configuration see [CPQ Package Configurations](#). You can use the name and label instead of the technical name.

+ Customer Filter / Product Filter: When you use this filter, the logic will get the value of these inputs and pass them to the filter logic. Then it will be applied to the Customer field on the header level or to the Product on the line item level.

How to configure:

- Yes/No
- or Customer/Product attribute based on your input Source Field

We provide the default implementation of customer/product filters for Source Type = P/C and Input Type = OPTION/OPTIONS. But you still need to build your own filter logic for the remaining source types and

input types to work.

Customer/ Product Filter Operator with supported values (drop-down):

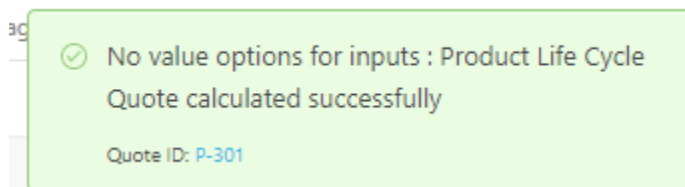
- OP_EQUAL (default for other supported input types)
- OP_NOT_EQUAL
- OP_LESS_THAN
- OP_GREATER_THAN
- OP_LESS_OR_EQUAL
- OP_GREATER_OR_EQUAL
- OP_LIKE
- OP_ILIKE
- OP_IN (default for input type is OPTIONS)
- OP_NOT_IN

+ Value Options Filtered By Customer/ Product (only applies when Input Type is OPTION, OPTIONS): When you select the Customer field on the header level or the Product on the line item level, we will show input values based on that. It is opposed to the Customer / Product Filter.

How to configure:

- If Source Type is P/C you only need to input customerId or sku
- Other Source Types, e.g: attribute of Customer Class (CX) = attribute of Customer Class (C)

There will be the following warning message if the value options are not found:



CPQ Outputs Wizard

The CPQ Outputs Configuration can show/hide outputs in the CPQ Accelerators. It allows business users to set what fields are visible in quoting and for which user groups those fields are available. Up to now this had to be set up in the logic itself.

For more details on configuration see [CPQ Package Configurations](#).

Use the CPQOutputConfiguration PP or go to Master Data Configuration Wizards CPQ Outputs Wizard.

Price Parameter Values : CPQ Outputs [5]

<input type="checkbox"/>	Name	User Group	Attr. 2
<input type="checkbox"/>	customerRevHistory	Admin	
<input type="checkbox"/>	Currency	Admin,Test	
<input type="checkbox"/>	customerQuantityHistory	Admin	
<input type="checkbox"/>	customerMarginHistory	Admin	
<input type="checkbox"/>	customerMarginPctHis...	Admin	

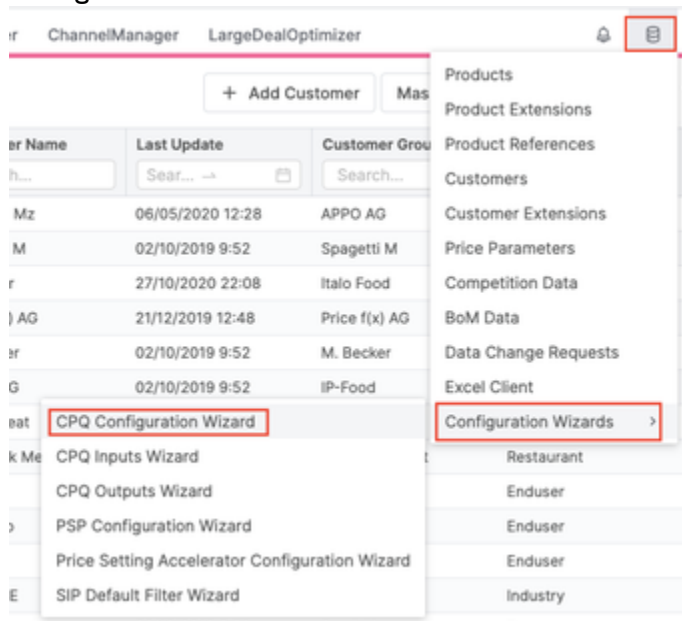
This is how you set up the access: in the PP table above, only user belonging to the given user group can see the items.

Note: When you change, update/delete something in the PP table, please remember to recalculate and save your quote first, then the other user group will not see it when they open it. Both the Header and Line Item level will be hidden. If a user has full admin roles, this configuration will not work.

Get Started

The CPQ Configuration Wizard makes the package configuration easier by providing a step-by-step user-friendly interface for the setup (instead of manual changes in Price Parameters).

The wizard is included in the deployment and can be accessed in Master Data Configuration Wizards CPQ Configuration Wizard.



Overview

The CPQ Configuration Wizard consists of three main steps which guide you through the configuration of each part of the CPQ.

Step 1: Select Quote Type

In this step, a list of quote types using the CPQ_Header logic is displayed to select from. The next configuration steps are applied to the selected quote type.

Then, you can specify the tables where the configurations/data should be used for the selected quote type by clicking the **Configure** button.

If you need a different table for each quote type, you need to prepare it yourself (the wizard does not support cloning/creating tables).

CPQ Configuration Wizard

This wizard helps you to configure the CPQ Package with ease. Please consult the [Quoting Tool](#) first to get familiar with the package and the [CPQ Package Configurations](#) to understand how it is configured properly.

Step 1: Configure Your Quote Type

__DEFAULT__

Configure

Step 2: Configure Your Quote Structure

Configure

Step 3: Configure Your Quote Default Modules

Configure

Step 2: Configuration Quote Structure

In this step, the wizard allows you to configure the quote structure like inputs, outputs, etc.

CPQ Configuration Wizard

This wizard helps you to configure the CPQ Package with ease. Please consult the [Quoting Tool](#) first to get familiar with the package and the [CPQ Package Configurations](#) to understand how it is configured properly.

Step 1: Configure Your Quote Type

Step 2: Configure Your Quote Structure

Step 3: Configure Your Quote Default Modules

Step 3: Configuration Per Module

In this step, each default module in CPQ can be selected and configured.

CPQ Configuration Wizard

This wizard helps you to configure the CPQ Package with ease. Please consult the [Quoting Tool](#) first to get familiar with the package and the [CPQ Package Configurations](#) to understand how it is configured properly.

Step 1: Configure Your Quote Type

Step 2: Configure Your Quote Structure

Step 3: Configure Your Quote Default Modules

How To

- [Add Quote Type To Be Configured by Wizard](#)
- [Separate Configurations per Quote Type](#)
- [Configure Custom Library Running on Header](#)
- [Configure Custom Inputs and Outputs](#)
- [Configure Error Handler](#)
- [Configure Historical Data](#)
- [Configure Product Price Extension Lookup Order](#)
- [Configure Customer Mapping](#)
- [Configure Standard Discount](#)
- [Configure Rebate Integration Configuration](#)
- [Configure Cost Configuration](#)

- [Configure Price Competitors Configuration](#)
- [Configure Price Guidances Configuration](#)
- [Troubleshooting](#)

Add Quote Type To Be Configured by Wizard

i Configuration Entry should be configured for each quote type, see [Configuration Per Quote Type](#).

When CPQ Configuration Wizard is used for CPQ, it loads only Quote Types that use the **CPQ_Header** logic. So, you need to select **CPQ_Header** logic as the **Header Logic** of your Quote Type.

Quote Types

<input type="checkbox"/>	Name	Last Update Date	Pricing Logic	Header Logic
<input type="checkbox"/>	(Default)	12/04/2021 16:39	CPQ Logic	CPQ Header Logic v2.0
<input type="checkbox"/>	CPQ	25/02/2021 15:24	CPQ Logic	CPQ Header Logic v2.0
<input type="checkbox"/>	AnotherCPQ	05/01/2021 15:00	CPQ Logic	CPQ Header Logic v2.0
<input type="checkbox"/>	michal	05/01/2021 14:56	michal-quote	CPQ Header Logic v2.0
<input type="checkbox"/>	Test_LineInput	01/02/2021 23:55	CPQ Logic	CPQ Header Logic v2.0
<input type="checkbox"/>	Test_Wizard	18/06/2021 14:53	CPQ Logic	CPQ Header Logic v2.0

Then it is ready to be configured by the wizard.

CPQ Configuration Wizard

Options «

Select Configuration Wizard

CPQ Configuration Wizard ▾

CPQ Configuration Wizard

This wizard helps you to configure the CPQ Package with ease. Please consult the [Quoting Tool](#) first to get familiar with the package and the [CPQ Package Configurations](#) to understand how it is configured properly.

Step 1: Quote Type

Pick a quote type you want to configure from the list below.

__DEFAULT__

AnotherCPQ

CPQ

michal

Test_LineInput

Test_Wizard

__DEFAULT__

Pick a configuration you want to configure from the list below and the wizard will guide you through the steps.

__DEFAULT__

Separate Configurations per Quote Type

Each quote type can be configured to have different:

- Tables where the custom logics are configured for running on header/line items.
- Tables where data (such as cost, discount, configurations, etc.) are read from.

Pre-requisites

- The tables should be cloned/created manually before starting the wizard.
- Quote Type should be selected in [Step 1](#).
- After the quote type is selected, click the **Configure** button, a new page will show and allow you to configure where to read the configuration and other data tables.

Separate Quote Configuration

Select the PP table that should store all other table names that are used by your quote.

CPQ Configuration Wizard

Configuration Entry

Configuring Quote Type - Default

Select Your Entry Table

Entry Table is price parameter table that store all table names using by your quote type.

CPQ_Default_Configuration_Entry

Select Your Configuration Table

Configuration Table is price parameter table that store all configurations using by your quote type.

CPQConfiguration

Back

Separate Data

Select the PP table that should store all default modules configurations that are used by your quote.

CPQ Configuration Wizard

Configuration Entry

Configuring Quote Type - Default

Select Your Entry Table

Entry Table is price parameter table that store all table names using by your quote type.

CPQ_Default_Configuration_Entry

Select Your Configuration Table

Configuration Table is price parameter table that store all configurations using by your quote type.

CPQConfiguration

Back

Apply Your Changes

Once you have all data selected, you need to click the **Apply** button to store your selections in the system.

If you click the **Back** button, you go back to the main screen and the current changes on the screen are ignored.

Configure Custom Library Running on Header

CPQ allows custom logics to be configured and run along with the default logics. The default logics can also be removed/replaced in the calculation. This can easily be done by using the CPQ Configuration Wizard.

Pre-requisites

- Custom logics are prepared in the form of a Pricefx Groovy library.
- Quote Type is selected in [Step 1](#).
- Tables are configured in [Step 2](#).

Select Configuration

In Step 3, select **Custom Configuration** in the first drop-down input and **Header Processing** in the second input.

CPQ Configuration Wizard

Options

Select Configuration Wizard

CPQ Configuration Wizard

Step 1: Quote Type
Pick a quote type you want to configure from the list below.

__DEFAULT__

Step 2: Configuration per Quote Type
This step is optional. If not, default configuration is configured.

Configuration Entry

Step 3: Select Configuration
Pick a configuration you want to configure from the list below and the wizard will guide you through the steps.

Custom Configuration

Detail Configuration
Header Processing

Configure

Then click the **Configure** button to enter the configuration screen.

Creating New Entry

On the configuration screen, select **Create a new header processing configuration for Quote**.

CPQ Configuration Wizard

Options

Select Configuration Wizard

CPQ Configuration Wizard

CPQ Configuration Wizard

Header Processing

The **Header Processing** wizard helps you to define the custom logics that be generated on the header level.

Create a new header processing configuration for Quote

Modify a header processing configuration for Quote

Remove a header processing configuration from Quote

Back

Fill in the inputs and click the **Apply** button to save the entry.

Modify an Entry

On the configuration screen, select **Modify a header processing configuration for Quote**.

CPQ Configuration Wizard

Options

Select Configuration Wizard

CPQ Configuration Wizard

CPQ Configuration Wizard

Header Processing

The **Header Processing** wizard helps you to define the custom logics that be generated on the header level.

Create a new header processing configuration for Quote

Modify a header processing configuration for Quote

Remove a header processing configuration from Quote

Back

Select an existing entry from the **Name** input. The configuration for the selected entry is loaded and ready for modifying.

Master Data / Configuration Wizards
CPQ Configuration Wizard

Options «

Select Configuration Wizard
CPQ Configuration Wizard

CPQ Configuration Wizard
Header Processing
Modify a header processing configuration for Quote

Name*
Configuration

Processor*
Data Query

Running Phase
PrePhase

Library Name
CPQ_Header_Utilis

Element Name
Configuration

Click the **Apply** button to save the configuration when finished.

Remove an Entry

On the configuration screen, select **Remove a header processing configuration from Quote**.

Master Data / Configuration Wizards
CPQ Configuration Wizard

Options «

Select Configuration Wizard
CPQ Configuration Wizard

CPQ Configuration Wizard
Header Processing
The **Header Processing** wizard helps you to define the custom logics that be generated on the header level.

Create a new header processing configuration for Quote

Modify a header processing configuration for Quote

Remove a header processing configuration from Quote

Back

Select an entry from the **Name** input.

Master Data / Configuration Wizards
CPQ Configuration Wizard

Options «

Select Configuration Wizard
CPQ Configuration Wizard

CPQ Configuration Wizard
Header Processing
Remove a header processing configuration for Quote

Name*
Configuration

Back To Header Processing

Apply

Then click **Apply** button to remove the entry.

Configure Custom Inputs and Outputs

CPQ allows custom inputs/outputs to be configured and generated along with the default inputs/outputs. The default inputs/outputs can also be removed/replaced in the calculation. This can be easily done by using the CPQ Configuration Wizard.

Pre-requisites

- Custom logics are prepared in the form of a Pricefx Groovy library.
- Quote Type is selected in [Step 1](#).
- Tables are configured in [Step 2](#).

Select Configuration

In Step 3, select **Custom Configuration** in the first drop-down input and **Input Output Processing** in the second input.

The screenshot shows the 'CPQ Configuration Wizard' interface. On the left, there is a sidebar with 'Options' and 'Select Configuration Wizard' sections, both containing a dropdown menu with 'CPQ Configuration Wizard' selected. The main content area is titled 'Step 3: Select Configuration' and contains the following elements:

- Step 1: Quote Type**: A dropdown menu with '___DEFAULT___' selected.
- Step 2: Configuration per Quote Type**: A section with a 'Configuration Entry' button.
- Step 3: Select Configuration**: A section with two dropdown menus. The first is set to 'Custom Configuration' and the second is set to 'Input Output Processing'. A red box highlights these two dropdown menus.
- A 'Configure' button is located at the bottom of the Step 3 section.

Then click the **Configure** button to enter the configuration screen.

Create New Entry

On the configuration screen, select **Create a new input output configuration for Quote** to enter the create new entry screen.

The screenshot shows the 'CPQ Configuration Wizard' configuration screen. The sidebar on the left is identical to the previous screenshot. The main content area is titled 'CPQ Configuration Wizard' and 'Input Output Processing'. It includes a description: 'The [Input Output Processing](#) wizard helps you to define the custom inputs/outputs that should be generated on the Quote.' Below this, there are four buttons: 'Create a new input output configuration for Quote', 'Modify an input output configuration for Quote', 'Remove an input output configuration from Quote', and 'Back'. A red box highlights the 'Create a new input output configuration for Quote' button.

Fill in the inputs in the create new entry screen and click the **Apply** button to save the entry.

Modify Entry

On the configuration screen, select **Modify an input output configuration for Quote** to enter the modify existing entry screen.

The screenshot shows the 'CPQ Configuration Wizard' interface. On the left, there is a sidebar with 'Options' and 'Select Configuration Wizard' sections, both containing a dropdown menu with 'CPQ Configuration Wizard' selected. The main content area is titled 'CPQ Configuration Wizard' and 'Input Output Processing'. Below the title, there is a descriptive sentence: 'The **Input Output Processing** wizard helps you to define the custom inputs/outputs that should be generated on the Quote.' Below this text are four buttons: 'Create a new input output configuration for Quote', 'Modify an input output configuration for Quote' (highlighted with a red circle), 'Remove an input output configuration from Quote', and 'Back'.

Select an existing entry from the **Generation Type** and **Name** inputs. The configuration for the selected entry is loaded and ready for modifying.

This screenshot shows the same 'CPQ Configuration Wizard' interface as the previous one, but now the 'Modify an input output configuration for Quote' screen is active. The 'Generation Type*' dropdown is set to 'Header Output' and the 'Name*' dropdown is set to 'Cost'. These two dropdowns are highlighted with a red circle. Below them are other input fields: 'Label*' with 'Cost', 'Input Type' (empty), 'Output Type' with 'SIMPLE', and 'Library Name' with 'CPQ_Output_Utilis'.

Click the **Apply** button to save the configuration when finished.

Remove Entry

On the configuration screen, select **Remove an input output configuration from Quote**.

CPQ Configuration Wizard

Options <<

Select Configuration Wizard

CPQ Configuration Wizard

CPQ Configuration Wizard

Input Output Processing

The [Input Output Processing](#) wizard helps you to define the custom inputs/outputs that should be generated on the Quote.

Create a new input output configuration for Quote

Modify an input output configuration for Quote

Remove an input output configuration from Quote

Back

Select an entry from the **Generation Type** and **Name** inputs.

CPQ Configuration Wizard

Options <<

Select Configuration Wizard

CPQ Configuration Wizard

CPQ Configuration Wizard

Input Output Processing

Remove an input output configuration for Quote

Generation Type*

Header Output

Name*

Cost

Back To Input Output Processing

Apply

Then click the **Apply** button to remove the entry.

Configure Error Handler

CPQ provides a tool to configure the error levels and messages. This configuration can easily be done using CPQ Configuration Wizard.

Pre-requisites

- Quote Type is selected in [Step 1](#).
- Tables are configured in [Step 2](#).

Select Configuration

In Step 3, select **Custom Configuration** in the first drop-down input and **Error Handler** in the second input.

CPQ Configuration Wizard

Options «

Select Configuration Wizard

CPQ Configuration Wizard

Step 1: Quote Type
Pick a quote type you want to configure from the list below.

___DEFAULT___

Step 2: Configuration per Quote Type
This step is optional. If not, default configuration is configured.

Configuration Entry

Step 3: Select Configuration
Pick a configuration you want to configure from the list below and the wizard will guide you through the steps.

Custom Configuration

Detail Configuration
Error Handler

Configure

Then click the **Configure** button to enter the configuration screen.

Create New Entry

On the configuration screen, select **Create a new error handler configuration for Quote**.

CPQ Configuration Wizard

Options «

Select Configuration Wizard

CPQ Configuration Wizard

CPQ Configuration Wizard
Error Handler
The **Error Handler** wizard helps you to define an error handler for configuring error levels and messages.

Create a new error handler configuration for Quote

Modify an error handler configuration for Quote

Remove an error handler configuration from Quote

Back

Fill in the inputs on the create new entry screen and click the **Apply** button to save the entry.

Modify Entry

On the configuration screen, select **Modify an error handler configuration for Quote**.

CPQ Configuration Wizard

Options «

Select Configuration Wizard

CPQ Configuration Wizard

CPQ Configuration Wizard

Error Handler

The [Error Handler](#) wizard helps you to define an error handler for configuring error levels and messages.

Create a new error handler configuration for Quote

Modify an error handler configuration for Quote

Remove an error handler configuration from Quote

Back

Select an existing entry from the **Name** input. The configuration for the selected entry is loaded and ready for modifying.

Master Data / Configuration Wizards

CPQ Configuration Wizard

Options «

Select Configuration Wizard

CPQ Configuration Wizard

CPQ Configuration Wizard

Error Handler

Modify an error handler configuration for quote

Name*

CUSTOMER_INVALID

Message

Customer not set

Level

WARNING

Error Handler

Abort Handler Execution

Back To Error Handler

Apply

Click the **Apply** button to save the configuration when finished.

Remove Entry

On the configuration screen, select **Remove an error handler configuration from Quote**.

CPQ Configuration Wizard

Options «

Select Configuration Wizard

CPQ Configuration Wizard

CPQ Configuration Wizard

Error Handler

The **Error Handler** wizard helps you to define an error handler for configuring error levels and messages.

Create a new error handler configuration for Quote

Modify an error handler configuration for Quote

Remove an error handler configuration from Quote

Back

Select an entry from the **Name** inputs.

Options «

Select Configuration Wizard

CPQ Configuration Wizard

CPQ Configuration Wizard

Error Handler

Remove an error handler configuration for quote

Name*

CUSTOMER_INVALID

Back To Error Handler

Apply

Then click the **Apply** button to remove the entry.

Configure Historical Data

CPQ can show historical sales data for quoted customers and products. Having such data requires a connection between the sales Datamart and CPQ logic. This can be easily done by using CPQ Configuration Wizard.

Pre-requisites

- Custom logics are prepared in the form of a Pricefx Groovy library.
- Quote Type is selected in [Step 1](#).
- Tables are configured in [Step 2](#).

Select Configuration

In Step 3, select **Core Configuration** in the first drop-down input and **Historical Data** in the second input.

CPQ Configuration Wizard

Options <<

Select Configuration Wizard

CPQ Configuration Wizard

Step 1: Quote Type
Pick a quote type you want to configure from the list below.

__DEFAULT__

Step 2: Configuration per Quote Type
This step is optional. If not, default configuration is configured.

Configuration Entry

Step 3: Select Configuration
Pick a configuration you want to configure from the list below and the wizard will guide you through the steps.

Core Configuration

Detail Configuration

Historical Data

Configure

Then click the **Configure** button to enter the configuration screen.

Historical Data Configuration

On the configuration screen, select the Datamart that contains the sales data. The Datamart fields will be fetched to select.

CPQ Configuration Wizard

Options <<

Select Configuration Wizard

CPQ Configuration Wizard

CPQ Configuration Wizard

Historical Data

The [Historical Data](#) wizard helps you to configure the fields from the datamart which are using in the Sales Overview, Customer History on the header level and the Product History, Previous Price on the line item level.

Historical Period

12

Datamart Name *

Standard_Sales_Data

Invoice Id *

DocumentId

Product Id *

ProductId

Customer Id *

CustomerId

Map the Datamart fields and click the **Apply** button when finished.

Configure Product Price Extension Lookup Order

CPQ allows you to set the Product Price Extension lookup order. This can be easily done by using the CPQ Configuration Wizard.

Pre-requisites

- Quote Type is selected and configured in [Step 1](#).

Select Configuration

In Step 3, select **Prices Configuration** in the drop-down input.

CPQ Configuration Wizard

This wizard helps you to configure the CPQ Package with ease. Please consult the [Quoting Tool](#) first to get familiar with the package and the [CPQ Package Configurations](#) to understand how it is configured properly.

Step 1: Configure Your Quote Type

Configure

Step 2: Configure Your Quote Structure

Configure

Step 3: Configure Your Quote Default Modules

Configure

Then click the **Configure** button to enter the configuration screen.

Price Lookup Configuration

This screen allows you to configure where and how the price data is stored in the Product Extension table.

CPQ Configuration Wizard

Price Configuration

Configuring Quote Type - Default

Select Your Price By Customer Table

Product Extension that contains your Price Per customer Id

PriceByCustomer

Select Your Price By Region Table

Product Extension that contains your Price Per Region

PriceByRegion

Select Your Price By Segmentation Table

Product Extension that contains your Price Per Segmentation

PriceBySegment

Select Your Price By Country Table

Product Extension that contains your Price Per Country

PriceByCountry

Select Price Lookup Order

Prioritize for Price Configuration

Customer x Country x Region x Segment x

Back

Select your Product Extension table that contains your price data and fill the price lookup order. Then click the **Apply** button.

Configure Customer Mapping

In CPQ, the Product Extension price lookup and currency conversion require mapping to the Customer Master. This can easily be done by using the CPQ Configuration Wizard.

Pre-requisites

- Quote Type is selected and configured in [Step 1](#).

Select Configuration

In Step 3, select **Customer Mapping** in the drop-down input.

CPQ Configuration Wizard

This wizard helps you to configure the CPQ Package with ease. Please consult the [Quoting Tool](#) first to get familiar with the package and the [CPQ Package Configurations](#) to understand how it is configured properly.

Step 1: Configure Your Quote Type

Step 2: Configure Your Quote Structure

Step 3: Configure Your Quote Default Modules

Then click the **Configure** button to enter the configuration screen.

Customer Mapping Configuration

On the configuration screen, select the mapping from the Customer Master.

CPQ Configuration Wizard

Customer Mapping

Configuring Quote Type - Default

Select Your Customer Country Attribute

Column that contains your Customer Country value

Select Your Customer Region Attribute

Column that contains your Customer Region value

Select Your Customer Segment Attribute

Column that contains your Customer Segment value

Select Your Customer Currency Attribute

Column that contains your Customer Currency value

Back

Then click the **Apply** button.

Configure Standard Discount

The Standard Discount table contains discount values for quoted products. The value can be configured in a combination of a maximum of 6 quote/product attributes. This can easily be done by using the CPQ Configuration Wizard.

Pre-requisites

- Quote Type is selected and configured in [Step 1](#).

Select Configuration

In Step 3, select **Discounts** in the drop-down input.

CPQ Configuration Wizard

This wizard helps you to configure the CPQ Package with ease. Please consult the [Quoting Tool](#) first to get familiar with the package and the [CPQ Package Configurations](#) to understand how it is configured properly.

Step 1: Configure Your Quote Type

Configure

Step 2: Configure Your Quote Structure

Configure

Step 3: Configure Your Quote Default Modules

Configure

Click the **Configure** button to enter the discount type selection screen.

Then click the **Standard Discount** button to open the configuration screen.

CPQ Configuration Wizard

Discounts

Configuring Quote Type - Default

Select your discount type to configure

Standard Discount

Exception Discount

Customer Cash Discount

Back

Standard Discount Mapping Configuration

This screen allows you to configure where and how the discount data is stored in the Price Parameter table.

CPQ Configuration Wizard

Standard Discount Configuration

Configuring Quote Type - Default

Standard discount mapping

Standard discount table

CPQStandardDiscount

Standard Discount Table Mapping

Target date column

Target Date

Currency column

Attribute 4

Next

Back

Select your Price Parameter table that contains the discount data and fill in the field mappings by selecting the available fields in the drop-down.

Then, click the **Next** button to move forward to the filter configuration screen.

Standard Discount Filter Configuration

The screen allows you to configure how your discount data should be filtered.

When loading for the first time, the screen shows the filters that you have already configured and saved in the system.

CPQ Configuration Wizard

Standard Discount Configuration

Configuring Quote Type - Default

Standard discount filter configuration

<input type="checkbox"/>	Column	Type	Source	Value
<input type="checkbox"/>	Key1	C	Customers	Cust

Add

Next

Back

You can add, edit or remove the filter configuration by selecting the respective action button.

CPQ Configuration Wizard

Standard discount filter configuration

<input checked="" type="checkbox"/>	Column	Type	Source	Value
<input checked="" type="checkbox"/>	Key1	C	Customers	Cu



Add

Edit

Remove

Next

Back

By clicking **Add** or **Edit**, a list of inputs will show for further configuration.

CPQ Configuration Wizard

Standard Discount Configuration

Configuring Quote Type - Default

Standard discount filter configuration

Column	Type	Source	Value
Key1	C	Customers	Customer Group



Editing: Key1

Column

Type

Source

Value

On the configuration screen for a specific key, describe the data that is used in the key column. This allows the CPQ logic to know where to get quoted product data to filter the standard discount value.

The available fields are:

- **Column** - Column in the product extension table. You can select from a list of columns.
- **Type** - The source where the value in the key column comes from. Available values are:
 - **C** - Customer Master
 - **P** - Product Master
 - **PX** - Product Extension
 - **CX** - Customer Extension
- **Source** - The table name where the filter value come from.
- **Value** - Column where the value in the key columns comes from. You can select from a list of column names.

When finished, click the **Next** button to see the summary.

Configuration Summary

The screen shows all of your configurations that have been done on the previous screens.

CPQ Configuration Wizard

Standard Discount Configuration

Configuring Quote Type - Default

Summary

Selected Table

CPQStandardDiscount

Selected target date column

Target Date

Selected currency column

Attribute 4

Filters

Column	Type	Source	Value
Key1	C	Customers	Customer Group



Confirm

Back

If you still want to modify your configuration, click the **Back** button.

When everything looks good, you need to click **Confirm** and then **Apply** to save all your configurations to the system.

Configure Rebate Integration Configuration

CPQ can access the rebate data generated by the Rebate Accelerator by predefined configurations. There is a Rebate Calculation Plan that can be configured using the CPQ Configuration Wizard.

Pre-requisites

- Custom logics are prepared in the form of a Pricefx Groovy library.
- Quote Type is selected in [Step 1](#).
- Tables are configured in [Step 2](#).

Select Configuration

In Step 3, select **Core Configuration** in the first drop-down input and **Rebate** in the second input.

Master Data / Configuration Wizards
CPQ Configuration Wizard

Options <<

Select Configuration Wizard

CPQ Configuration Wizard

Step 1: Quote Type
Pick a quote type you want to configure from the list below.

__DEFAULT__

Step 2: Configuration per Quote Type
This step is optional. If not, default configuration is configured.

Configuration Entry

Step 3: Select Configuration
Pick a configuration you want to configure from the list below and the wizard will guide you through the steps.

Core Configuration

Detail Configuration

Rebate

Configure

Then click the **Configure** button to enter the configuration screen.

Rebate Calculation Plan Calculation

On the configuration screen, select the desired **Rebate Calculation Plan**.

Master Data / Configuration Wizards
CPQ Configuration Wizard

Options <<

Select Configuration Wizard

CPQ Configuration Wizard

CPQ Configuration Wizard

Rebate
The **Rebate** wizard helps you to define the rebate calculation plan to get the rebate data for a customer and product from Rebate Manager module. It is used in the Rebate output element.

Rebate Calculation Plan

Current

Back

Apply

When finished, click the **Apply** button.

Configure Cost Configuration

CPQ allows you to customize how the cost data is read from the Product Extension table. This can easily be done using the CPQ Configuration Wizard.

Pre-requisites

- Quote Type is selected and configured in [Step 1](#).

Cost Module Configuration Selection

In Step 3, select **Cost** in the drop-down input.

CPQ Configuration Wizard

Quoting Tool first to get familiar with the package and the [CPQ Package Configurations](#) to understand how it is configured properly.

Step 1: Configure Your Quote Type

__DEFAULT__

Configure

Step 2: Configure Your Quote Structure

Configure

Step 3: Configure Your Quote Default Modules

Cost

Configure

Then click the **Configure** button to enter the Cost data configuration flow.

Cost Data Mapping Configurations

This screen allows you to configure where and how the cost data is stored in the Product Extension table.

CPQ Configuration Wizard

Cost

Configuring Quote Type - Default

Select Your Cost Table

Product Extension that contains your cost data.

Cost Table Structure

Configure how the cost is stored on the table.

Cost Column*

Currency Column

Target Date

Valid To Date

Next

Back

Select your Product Extension table that contains the cost data and fill in the field mappings by selecting the available fields in the drop-down menu.

Then, click the **Next** button to move forward to the Cost Filter configuration screen.

Cost Filtering

The next screen allows you to configure how your cost data should be filtered, such as the cost should be different for each customer country.

CPQ Configuration Wizard

Cost Filter

Configuring Quote Type - Default

Configure Cost Filters

<input type="checkbox"/> Column	Type	Source	Value
<input type="checkbox"/> DependencyLevelNa...	C	Customers	Country

Add

Next

Back

When loading for the first time, the screen shows the filters that you have already configured and saved in the system.

You can add, edit or remove the filter configuration by selecting the respective action button.

CPQ Configuration Wizard

Cost Filter

Configuring Quote Type - Default

Configure Cost Filters

<input checked="" type="checkbox"/> Column	Type	Source	Value
<input checked="" type="checkbox"/> DependencyLevelNa...	C	Customers	Country

Add

Edit

Remove

Next

Back

By clicking **Add** or **Edit**, a list of inputs will show for further configuration.

Cost Filter

Configuring Quote Type - Default

Configure Cost Filters

Column	Type	Source	Value
DependencyLevelN...	C	Customers	Country



Editing: DependencyLevelName

Column

DependencyLevelName

Type

C

Source

Customers

Value

Country

Ok

Cancel

The available fields are:

- **Column** - Column in the product extension table. You can select from a list of columns.
- **Type** - The source where the value in the key column comes from. Available values are:
 - **C** - Customer Master
 - **P** - Product Master
 - **PX** - Product Extension
 - **CX** - Customer Extension
- **Value** - Column where the value in the key columns comes from. You can select from a list of column names.

When finished, click the **Next** button to see the summary.

Cost Configuration Summary

The screen shows all of your cost configurations that have been done on the previous screens.

Summary

Configuring Quote Type - Default

Cost Configuration Summary

Cost Table

CostData

Cost Column

Cost

Currency Column

Currency

Target Date Column

ValidFrom

Valid To Date Column

ValidTo

Cost Filters

Column	Type	Source	Value
DependencyLevelN...	C	Customers	Country



Confirm

Back

Apply

To go back and modify your configuration, click the **Back** button.

When everything looks good, you need to click **Confirm** and then **Apply** to save all your configurations to the system.

Configure Price Competitors Configuration

CPQ allows you to customize how the price competitor data is read from the Product Extension table. This can easily be done using the CPQ Configuration Wizard.

Pre-requisites

- Quote Type is selected and configured in [Step 1](#).

Price Competitors Module Configuration Selection

In Step 3, select **Price Competitors** in the drop-down input.

CPQ Configuration Wizard

This wizard helps you to configure the CPQ Package with ease. Please consult the [Quoting Tool](#) first to get familiar with the package and the [CPQ Package Configurations](#) to understand how it is configured properly.

Step 1: Configure Your Quote Type

__DEFAULT__

Configure

Step 2: Configure Your Quote Structure

Configure

Step 3: Configure Your Quote Default Modules

Price Competitors

Configure

Then click the **Configure** button to enter the configuration mapping screen.

Price Competitors Mapping Configuration

This screen allows you to configure where and how the Price Competitors data is stored in the Product Extension table.

CPQ Configuration Wizard

Price Competitors

Configuring Quote Type - Default

Select Your Price Competitors Table

Product Extension that contains your competitors data.

Select Your Price Competitors Table*

PriceCompetitors

Price Competitor Mapping

Select Competitor Name Mapping*

competitor

Select Competitor Price Mapping*

price

Select Date Mapping

TargetDate

Back

Select your Product Extension table that contains your competitor data and fill in the field mappings by selecting the available fields in the drop-down menu.

When finished, click the **Apply** button to save the configuration.

Configure Price Guidances Configuration

The CPQ Configuration Wizard makes it easier to integrate the PriceOptimizer data with the CPQ logic.

Pre-requisites

- Quote Type is selected and configured in [Step 1](#).

Price Guidance Module Configuration Selection

In Step 3, select **Price Guidance** in the drop-down input.

CPQ Configuration Wizard

Quoting Tool first to get familiar with the package and the [CPQ Package Configurations](#) to understand how it is configured properly.

Step 1: Configure Your Quote Type

Configure

Step 2: Configure Your Quote Structure

Configure

Step 3: Configure Your Quote Default Modules

Configure

Then click the **Configure** button to enter the configuration selection screen.

Price Optimization Configuration

To configure the Price Guidance using the Price Optimization, click on the **Price Optimization** button

CPQ Configuration Wizard

Price Guidance

Configuring Quote Type - Default

Select guidance type to configure

Price Optimization

Product Extension

Back

Model Mapping Configuration

This screen allows you to configure where and how to read your Price Guidance from PriceOptimizer.

CPQ Configuration Wizard

Price Optimization Model Configuration

Configuring Quote Type - Default

Price Optimization Model Mapping

Select your model *

PO_v04

Guidance Mapping

Select stretch price mapping *

Stretch

Select target price mapping *

Target

Select floor price mapping *

Floor

Next

Back

Select your model that contains the price guidance and select the mapping for stretch, target, and floor prices.

When finished, click **Next** to enter the Segmentations configuration screen.

Segmentations Configuration

The next screen allows you to configure the segmentations of the PriceOptimizer Model.

When loading for the first time, the screen shows the segmentations that you have already configured and saved in the system.

CPQ Configuration Wizard

Price Optimization Model Configuration

Configuring Quote Type - Default

Segmentation Levels

<input type="checkbox"/> Column	Type	Source	Value
<input type="checkbox"/> ProductClass	P	Products	Product
<input type="checkbox"/> ProductGroup	P	Products	Product
<input type="checkbox"/> BusinessUnit	P	Products	Business



Next

Back

You can add, edit or remove the filter configuration by selecting the respective action button.

CPQ Configuration Wizard

Segmentation Levels

<input checked="" type="checkbox"/> Column	Type	Source	Value
<input checked="" type="checkbox"/> ProductGroup	P	Products	Product
<input type="checkbox"/> BusinessUnit	P	Products	Business



Add

Edit

Remove

Next

Back

After clicking **Add** or **Edit**, a list of inputs will show for further configuration.

CPQ Configuration Wizard

Price Optimization Model Configuration

Configuring Quote Type - Default

Segmentation Levels

Column	Type	Source	Value
ProductGroup	P	Products	Product Group
BusinessUnit	P	Products	Business Unit

Editing: ProductGroup

Column

ProductGroup

Type

P

Source

Products

Value

Product Group

Ok

Cancel

Next

Done

The available fields are:

- The available fields are:
 - **Column** - Column in the product extension table. You can select from a list of columns.
 - **Type** - The source where the value in the key column comes from. Available values are:
 - **C** - Customer Master
 - **P** - Product Master
 - **PX** - Product Extension
 - **CX** - Customer Extension
 - **Value** - Column where the value in the key columns comes from. You can select from a list of column names.

When finished, click the **Next** button to see the summary.

Price Guidance Configuration Summary

The screen shows all of your price guidance configurations that have been done on the previous screens.

CPQ Configuration Wizard

Price Optimization Model Configuration

Configuring Quote Type - Default

Summary

Selected Model

PO_v04

Stretch Price

Stretch

Target Price

Target

Floor Price

Floor

Segmentations

Column	Type	Source	Value
ProductGroup	P	Products	Product Group
BusinessUnit	P	Products	Business Unit



Confirm

Back

If you want still to modify your configuration, click the **Back** button.

When everything looks good, you need to click **Confirm** and then **Apply** to save all your configurations to the system.

Product Extension Configuration

To configure the Price Guidance using the Product Extension table, click on the **Product Extension** button

CPQ Configuration Wizard

Price Guidance

Configuring Quote Type - Default

Select guidance type to configure

Price Optimization

Product Extension

Back

Table Mapping Configuration

This screen allows you to configure where and how the price guidance is stored on the Product Extension table

CPQ Configuration Wizard

Price Guidance from Product Extension

Configuring Quote Type - Default

Product Extension Price Guidance Mapping

Select your table*

Guidance Mapping

Select stretch price mapping*

Select target price mapping*

Select floor price mapping*

Target date column

Currency column

Select your Product Extension table that contains the price guidance and select the mapping for stretch, target, and floor prices.

When finished, click **Next** to enter the Filter configuration screen.

Table Filter Configuration

This screen allows you to configure how your price guidance should be filtered.

CPQ Configuration Wizard

Price Guidance From Product Extension Filter Configuration

Configuring Quote Type - Default

Price Guidance Filter

Column	Type	Source	Value
attribute1	C	Customers	customerid
attribute2	C	Customers	Country



Editing: attribute2

Column

Type

Source

Value

Click the **Next** button to see the summary.

Table Configuration summary

This screen show all configuration that you had done in previous steps.

CPQ Configuration Wizard

Price Guidance From Product Extension Configuration Summary

Configuring Quote Type - Default

Product Extension Price Guidance Summary

Selected table

PriceGuidance

Selected stretch price mapping

Stretch

Selected target price mapping

Target

Selected floor price mapping

Floor

Target date column

Target Date

Currency column

Attr. 5

Filters

Column	Type	Source	Value
attribute1	C	Customers	customerId
attribute2	C	Customers	Country



Confirm

Back

If you want still to modify your configuration, click the **Back** button.

When everything looks good, you need to click **Confirm** and then **Apply** to save all your configurations to the system.

Troubleshooting

In this section:

- [Configuration value is not updated into tables](#)
- [Quote Type is not displayed for selection](#)

Configuration value is not updated into tables

Issue: When selecting a value and clicking the **Apply** button, the value is not updated in the table.

Solution: You might need to press the Tab key or click outside the input before clicking the **Apply** button.

CPQ Configuration Wizard

Publishing Templates

The [Publishing Templates](#) wizard helps you to get data from quote, customer or line item. It is used to export the PDF file for a Quote.

Value Getting From

Customer

Attribute To Get Value

Customer Group

Name Used In Template

CustomerGroup

Back

Apply

Quote Type is not displayed for selection

Issue: Your expected Quote Type is not displayed in the wizard for selection.

Solution: Make sure the Quote Type uses the **CPQ_Header** logic.

CPQ Package Architecture

- [Library Dependency](#)
- [Data Structures](#)
- [Components](#)

Library Dependency

- [ApprovalWorkflow](#)
- [FormulaEvaluator](#)

Data Structures

Mandatory Tables

Table Type	Table Name	Mandatory Fields	Optional Fields	Description
Product Extension	Product Costs	<ul style="list-style-type: none"> Product Id Cost 	<ul style="list-style-type: none"> Currency Target Date Valid To Date Additional dimensions filter 	Contains cost per product which is used in the CPQ cost output element.
Product		<ul style="list-style-type: none"> Product Id (SKU) Label 		
Customer		<ul style="list-style-type: none"> Customer Id Name 		

Optional Tables

Table Type	Table Name	Mandatory Fields	Optional Fields	Description
Product Extension	PriceGuidance	<ul style="list-style-type: none"> Product Id Target Floor Stretch 	<ul style="list-style-type: none"> Additional dimensions Target date 	Contains price guidance per product, included target price, floor price and stretch price.
Product Extension	PriceCompetitors	<ul style="list-style-type: none"> Product Id Competitor Price 	<ul style="list-style-type: none"> Target date 	Contains a competitor price per product.
Customer Extension	CustomerCashDiscount	<ul style="list-style-type: none"> Customer Id Discount 	<ul style="list-style-type: none"> Target date 	Contains discount information per customer (customer ID).
Pricing	StandardDisco		<ul style="list-style-type: none"> Target date 	

Parameter	unt	<ul style="list-style-type: none"> • Key 1 - Key6 • Discount • Value Type 		Standard discount configuration. There are six keys to configure to what a discount value belongs. We can also define if the discount value is a percentage or absolute value.
Pricing Parameter	ExceptionDiscount	<ul style="list-style-type: none"> • Customer (key1) • Product (key2) • Discount (attribute1) 	<ul style="list-style-type: none"> • Target date 	Exception for the discount configuration. The exception discount is defined for customer/product pairs. Values for the customer and product can be an attribute in the customer/product data or in PX/CX.
Pricing Parameter	CPQInputConfiguration	<ul style="list-style-type: none"> • Name • Label • Input Type 	<ul style="list-style-type: none"> • Source table • Source type • Source field • Value options • Default value • Value hint • Required • Level • Customer Filter • Product Filter • Value options filtered by customer • Value options filtered by product 	Contains configurations for custom inputs
Pricing Parameter	CPQOutputConfiguration	<ul style="list-style-type: none"> • Name 	<ul style="list-style-type: none"> • User group 	Contains configurations for outputs

Product Extensions

- PriceByCustomer - Contains a price per product per customer which is considered to be a list price of the product.
 - Product Id (SKU)
 - Customer (attribute1): customer ID/number
 - Price (attribute2)
- PriceByCountry - Contains a price per product per country which is considered to be a list price of the product.
 - Product Id (SKU)
 - Country (attribute1)
 - Price (attribute2)
- PriceByRegion - Contains a price per product per region which is considered to be a list price of the product.
 - Product Id (SKU)
 - Region (attribute1)
 - Price (attribute2)
- PriceBySegment - Contains a price per product per segmentation which is considered to be a list price of the product.
 - Product Id (SKU)
 - Segment (attribute1)
 - Price (attribute2)

Components

- Calculation Logic
 - CPQ - CPQ quoting logic
 - CPQ_SharedLib
 - CPQ_Publishing_Template_Data_Provider
 - CPQ_header - CPQ header logic
 - CPQ_header_discounts_configurator - Configurator logic for the discount type input
 - CPQ_CustomerFilter
 - CPQ_ProductFilter
 - HeaderInputs_Configurator
 - LineInputs_Configurator
 - CPQ_Output_Configuration_Wizard
 - CPQ_Output_Configuration_Wizard_Executor
- Publishing Template
 - publishingTemplates_Q_CPQTemplate.docx
- Message Template
 - Quote_approvalRequired_en.html
- Workflow Formula
 - CPQ_Quote - Workflow formula based on the ApprovalWorkflow library, with the data map extended for the quote summary.
- Quote Type
 - CPQ - Dedicated quote type for CPQ, with logic, header logic and workflow logic setup.

Configurator Package Integration

- Product Configurator
- Usage
- Price Settings
- Customer and Product Filter

Product Configurator

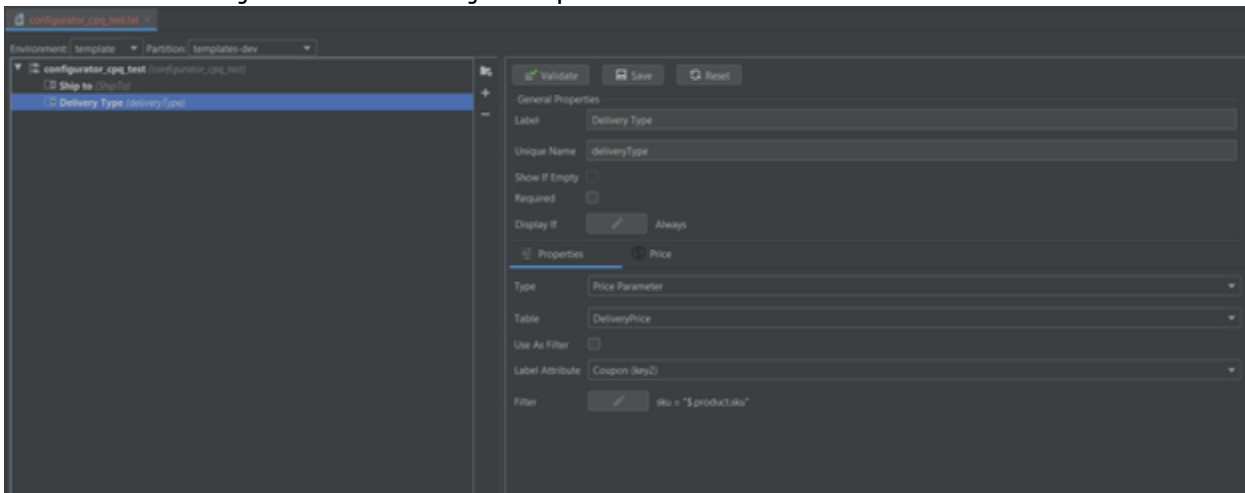
Integration with Configurator Package (<https://gitlab.pricefx.eu/accelerators/configurator-package>)

Note: It is planned to consolidate it with the existing CPQ Input configuration in the next release.

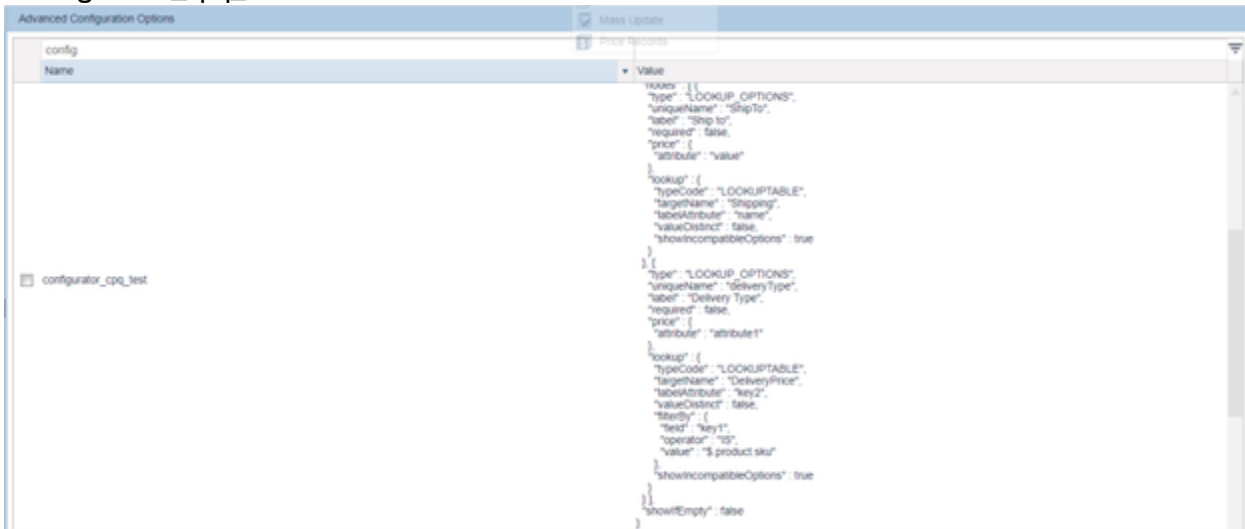
Usage

You can design your input configurators using ConfiguratorDesigner (<https://pricefx.atlassian.net/wiki/spaces/KB/pages/2344190596/Configurator+Designer>) and use it in CPQ.

Let's assume that you have created your inputs:



It is deployed now and its configuration is stored in AdvancedConfiguration under the name "configurator_cpq_test".



To be able to use your inputs in CPQ, all you need is to add a configuration entry to CPQConfiguration PP table or AdvancedConfiguration. For details see [CPQ Package Basic Configurations](#).

Configuration Name	Value
productConfigurator	configurator_cpq_test

Then your inputs are shown in CPQ:

The screenshot shows the CPQ configuration interface. At the top, there is a table with columns: Folder, Product Id, Net Price, Cost, Margin, Margin %, Revenue, and Currency. The table contains two rows: 'Copy of Test Product Configurator' and 'Meatball PI'. Below the table, there is a section for 'Input Parameters' with a '+' and '-' icon. Under 'Default Inputs', there are several input fields: Quantity (12), Price, Discount %, Ship to (Japan), and Delivery Type (Express). The 'Ship to' and 'Delivery Type' fields are highlighted with a red box. Below 'Default Inputs' is a section for 'Custom Inputs' with a 'Product Type' dropdown menu.

Price Settings

Configurator package allows you to [set a price for each input](#). You can use these input prices in CPQ as the product price.

To do so, you need to set "productConfiguratorInputPriority" in "priceVsDiscountPctInputPriority" configuration:

Configuration Name	Value
priceVsDiscountPctInputPriority	productConfiguratorInputPriority, discountPercentageInputPriority,priceInputPriority

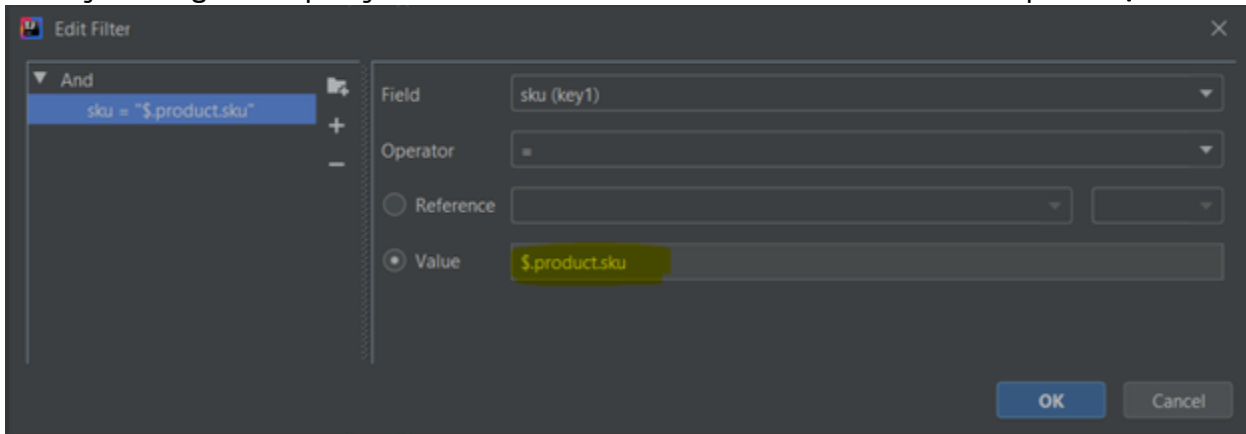
By default, the price will be the total of all input prices. However, you can calculate it using your formula.

Configuration Name	Value
productConfigurator_priceFormula	Your Formula. Ex. " deliveryType * (1 + ShipTo) " Where deliveryType and ShipTo are the input name

Customer and Product Filter

While integrating with CPQ, you can specify whether your input is filtered by the quoted customer or product.

When you design the input, you need to [add a filter](#) and add a value to it with a prefix "\$."



The supported values are:

Name	Description	Example
sku	Current product Id	\$.sku
customerId	Quoted customer Id	\$.customerId
product.[product attribute]	Attribute on of product	\$.product.attribute1
customer.[customer attribute]	Quoted customer attribute	\$.customer.attribute12

Configurations Per Quote Type

i In version 2.0, this configuration is removed.

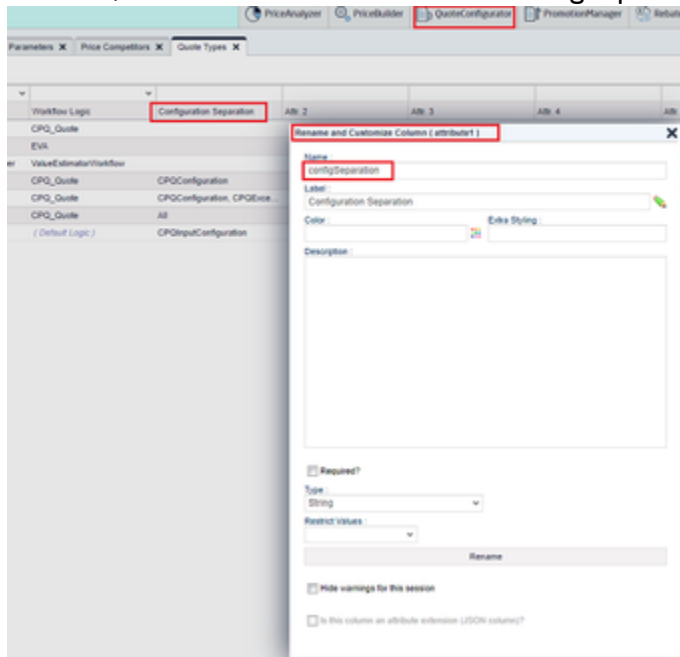
You can configure that each quote type reads data and configurations from difference advanced configuration keys and tables (PP, PX, CX). This way you can control behavior of different types of quotes.

Source Type	Name	Default Type	Quote Type (e.g.: AnotherCPQ)
Advanced Configuration	cpq-default	cpq-default	AnotherCPQ_cpq-default
	cpq-historical-data-default	cpq-historical-data-default	AnotherCPQ_cpq-historical-data-default
	cpq-generic-default	cpq-generic-default	AnotherCPQ_cpq-generic-default
Customer Extension (CX)	CustomerCashDiscount	CustomerCashDiscount	AnotherCPQ_CustomerCashDiscount
Product Extension (PX)	ProductCosts	ProductCosts	AnotherCPQ_ProductCosts

	PriceByCustomer	PriceByCustomer	AnotherCPQ_PriceByCustomer
	PriceByCountry	PriceByCountry	AnotherCPQ_PriceByCountry
	PriceBySegment	PriceBySegment	AnotherCPQ_PriceBySegment
	PriceByRegion	PriceByRegion	AnotherCPQ_PriceByRegion
	PriceGuidance	PriceGuidance	AnotherCPQ_PriceGuidance
	PriceCompetitors	PriceCompetitors	AnotherCPQ_PriceCompetitors
Price Parameter (PP)	CPQConfiguration	CPQConfiguration	AnotherCPQ_CPQConfiguration
	CPQStandardDiscount	CPQStandardDiscount	AnotherCPQ_CPQStandardDiscount
	CPQExceptionDiscount	CPQExceptionDiscount	AnotherCPQ_CPQExceptionDiscount
	CPQInputConfiguration	CPQInputConfiguration	AnotherCPQ_CPQInputConfiguration
	CPQOutputConfiguration	CPQOutputConfiguration	AnotherCPQ_CPQOutputConfiguration

Set a Quote Type

Before setting a quote type, you need to specify which quote type attribute will store the configuration. To do so, edit the column and name it "configSeparation".



From now on, you can configure which source you need to separate for the specific quote type:

- **Leave empty** - No need to separate, use the original tables.
- **All** - All tables, advanced configurations need to be cloned per a quote type. Add "QuoteTypeName_" as a prefix, e.g. AnotherCPQ_CPQConfiguration.

- **List of table names separated by comma** - Specify which default table / advanced configurations need to be cloned for each quote type.

Quote Type Management [7]

Name	Last Update	Pricing Logic	Header Logic	Workflow Logic	Configuration Separation
(Default)	20/05/2020	CPQ	CPQ_header	CPQ_Quote	
EVA	04/05/2020	ValueDriversQuote	QuoteHeaderDriver	EVA	
Value Estimator	21/08/2020	ValueEstimatorQuote	ValueEstimatorQuoteHeader	ValueEstimatorWorkflow	
AnotherCPQ	24/07/2020	CPQ	CPQ_header	CPQ_Quote	CPQConfiguration
TestQuoteType	24/07/2020	CPQ	CPQ_header	CPQ_Quote	CPQConfiguration, CPQExceptionDiscount, PriceByCountry, CustomerCashDiscount
TestQuoteTypeAll	24/07/2020	CPQ	CPQ_header	CPQ_Quote	All
QuoteType_SJ	29/07/2020	CPQ	CPQ_header	(Default Logic)	CPQInputConfiguration

Prepare Source

For each quote type, we need to clone the default configured tables / advanced configuration and derive from it our own configurations for the selected quote type. That way logics can read the correct configurations for the given quote type.

Price Parameters [12]

Name	Label	Valid After	Table Type	Value Type	Status	Simulation Set
CPQConfiguration	CPQ Logic Configuration	01/01/2018	SIMPLE	STRING	Active	
CPQExceptionDiscount	CPQ Exception Discount	01/01/2018	MATRIX	MATRIX2	Active	
CPQInputConfiguration	CPQ Inputs Configuration	01/01/2018	MATRIX	MATRIX	Active	
CPQStandardDiscount	CPQ Standard Discount	01/01/2018	MATRIX	MATRIX6	Active	
CPQOutputConfiguration	CPQ Outputs Configuration	01/01/2018	MATRIX	MATRIX	Active	
AnotherCPQ_CPQConfiguration	CPQ Logic Configuration_Test Quote Type	20/07/2020	SIMPLE	STRING	Active	
TestQuoteTypeAll_CPQExceptionDiscount	CPQ Exception Discount_Test Quote Type 01	20/07/2020	MATRIX	MATRIX2	Active	
TestQuoteTypeAll_CPQConfiguration	CPQ Logic Configuration_Test Quote Type 01	20/07/2020	SIMPLE	STRING	Active	
TestQuoteTypeAll_CPQInputConfiguration	CPQ Inputs Configuration	20/07/2020	MATRIX	MATRIX	Active	
TestQuoteTypeAll_CPQStandardDiscount	CPQ Standard Discount	20/07/2020	MATRIX	MATRIX6	Active	
TestQuoteTypeAll_CPQOutputConfiguration	CPQ Outputs Configuration	20/07/2020	MATRIX	MATRIX	Active	
QuoteType_SJ_CPQInputConfiguration	CPQ Inputs Configuration	29/07/2020	MATRIX	MATRIX	Active	

In the image above, we have the default and "AnotherCPQ" quote type.

We also separate data for both quote types in "CPQConfiguration" table.

Then we clone/copy this table and rename them with the "AnotherCPQ_" prefix.

Now, each quote type will have a different master and input configurations and the remaining configurations will read from the original source, e.g. "CPQStandardDiscount", "CPQOutputConfiguration", etc.

CPQ Element Names for Approval Workflow

[Accelerate Approval Workflow Package \(AWP\)](#) can be used with CPQ package.

To learn what elements can be used in workflow step conditions see [Element Name in Accelerators](#).

CPQ Package 2.1.0

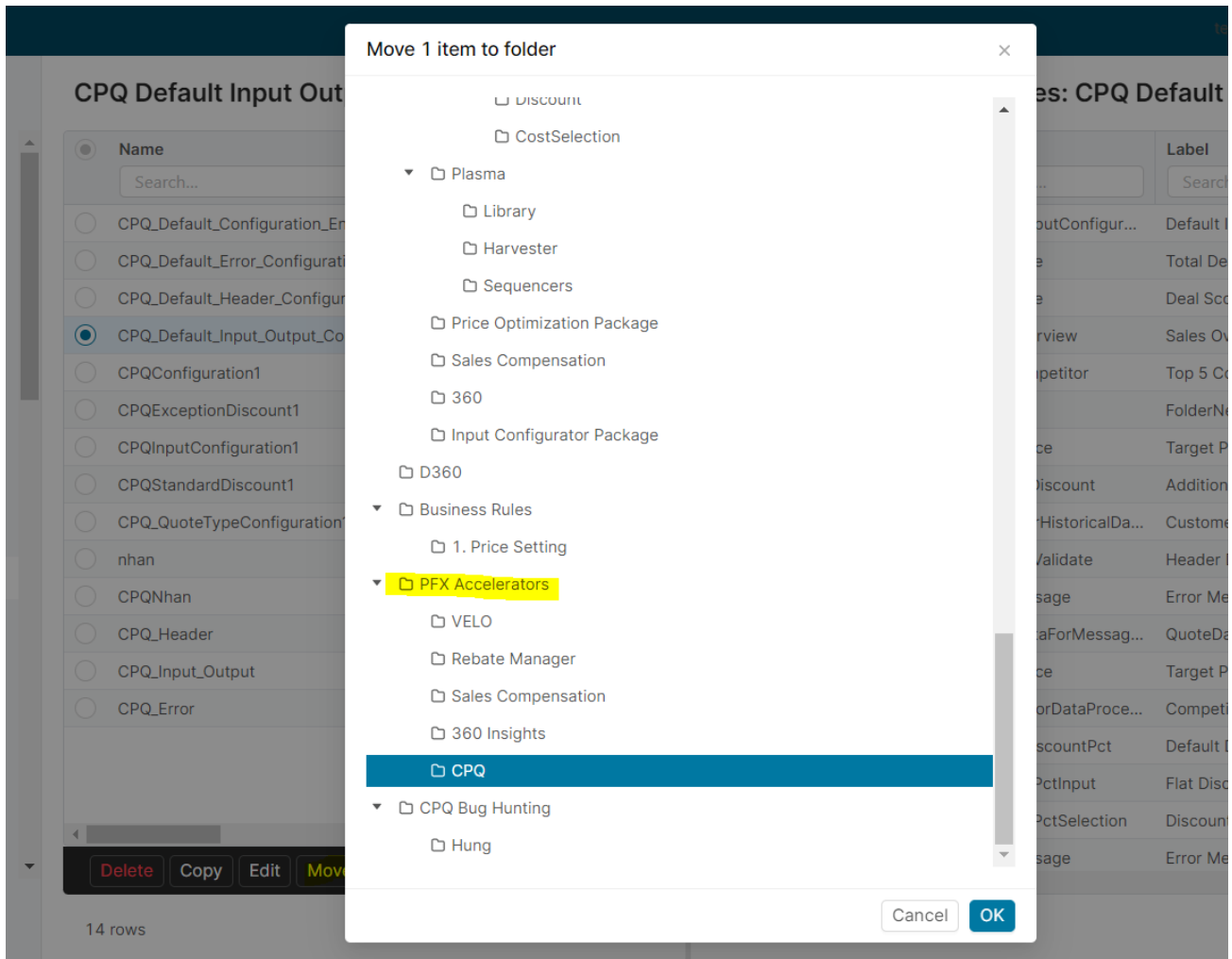
i Please back up your existing CPQ Company Parameters tables before upgrading to CPQ Package 2.1.0.

Release Highlights

- New Configuration wizard UIs
- Improved Optimization module integration
- Improved performance
- Added support for more input/output configurations
- More configuration mapping
- Flexible lookup filter configuration

Upgrade Notes

- The default PP table's folder has changed to "PFX Accelerators/CPQ". So, you need to move all of the existing default PP tables to the new location



- The predefined variables auto-injection mechanism is now configurable - users can disable the injection to improve performance. See [Predefined Variables](#) for the details.

Change Requests

- [PFPCS-5085](#) Data lookup using new BatchLookupUtils

Improvements

[PFPCS-5463](#) Historical Datamart name configuration should be moved to ConfigurationEntry table

[PFPCS-5436](#) Submit button is not greyed out and disabled on critical alert raising

[PFPCS-5435](#) CPQ Accelerator doesn't support calculation result override feature

[PFPCS-5366](#) Show Segment Id in Price Guidance calculation result in Quote

[PFPCS-5354](#) Add Currency field mapping in Price Competitors wizard

PFPCS-5352 CPQ Wizard: Put a short introduction for each configuration back
PFPCS-4851 Change default output Quantity to NUMERIC
PFPCS-3572 Add ability to store Price Guidance by Product and Customer (CPQ Accelerator)
PFPCS-5502 Potential "Too many new instances created"
PFPCS-5499 Deployment step - Add Product Cost mapping
PFPCS-5498 Introduce new quote type named CPQ in Quote Types configuration

Stories

PFPCS-5467 Check mandatory fields in CPQ and fix them
PFPCS-5272 PX price guidance configuration wizard
PFPCS-5251 Reduce Duplication In Wizard Screen Code
PFPCS-5235 Recreate Historical Data Screen
PFPCS-5201 Recreate Publishing Template Screen
PFPCS-5189 Recreate Other Screen
PFPCS-5187 Recreate Rebate Screen
PFPCS-5179 Create Customer Mapping Screen
PFPCS-5161 Update Main config and remove unnecessary constant
PFPCS-5160 Price Competitor is not support for configuration
PFPCS-5106 CPQ quote type configuration in PP table
PFPCS-5097 Configuration Wizard - Currency Configuration Screen
PFPCS-4924 V2.x Performance review & improvement
PFPCS-4918 Show results from Deal Guidance (PO) in CPQ Accelerator
PFPCS-4917 Query Deal Guidance from CPQ Accelerator
PFPCS-4909 Configuration Wizard- Price Guidance Configuration
PFPCS-4908 Configuration Wizard - Discount Configuration Screen
PFPCS-4907 Configuration Wizard - Price Competitor Screen
PFPCS-4906 Configuration Wizard - Price Configuration Screen
PFPCS-4905 Configuration Wizard - Cost Configuration Screen
PFPCS-4904 Configuration Wizard - Main Screen
PFPCS-4903 Configuration Wizard - Quote Type Configuration Screen
PFPCS-4902 Configuration Wizard - Layout update
PFPCS-3601 PO lookup configuration support for new Model.

Bugs

- PFPCS-5496 Library runner binding config - "No" value is not disabled the feature
- PFPCS-5495 Restructure CPQ folder
- PFPCS-5489 Error thrown in quotes when not mapping Datamart
- PFPCS-5482 Configuration per quote type: Incorrect Historical Data values
- PFPCS-5481 Configuration per quote type: Source (Segment Id/ Price Guidance PX table name) is not displayed
- PFPCS-5480 Missing pre-defined values (Critical, Yellow, Red) in Alert Type
- PFPCS-5479 CPQ Input Output wizard: Missing Overridable, Overridden, Alert Type, Alert Message
- PFPCS-5470 Missing table name configuration for error, header processing, input output
- PFPCS-5469 Configuration per quote type: Competitor does not get default entry table when having incorrect entry table structure
- PFPCS-5465 Unused Names in CPQ_Default_Configuration_EntryPP
- PFPCS-5464 Page Not Found when clicking a link in wizard
- PFPCS-5462 PX Price Guidance shows 0 when no mapping Currency in wizard
- PFPCS-5460 Price Guidance PO gets defaultCurrency configuration instead of base currency in DM
- PFPCS-5457 Price Guidance is not converted correctly when looking up from PO
- PFPCS-5455 Missing required inputs in publishing templates
- PFPCS-5454 Issue when adding subfolder in quote
- PFPCS-5449 All steps are optional during PlatformManager deployment
- PFPCS-5447 Rebate shows data while no rebate calculation plan is configured
- PFPCS-5443 Missing 0 before decimal places in publishing templates
- PFPCS-5441 null is displayed in Currency input when default currency is not defined
- PFPCS-5439 Price Guidance wizard: Able to go next step while required inputs are not filled
- PFPCS-5438 PO Price Guidance wizard: No data is displayed in Guidance Mapping when leaving required inputs empty
- PFPCS-5420 Existing PX table configured before is not displayed when going to wizard
- PFPCS-5416 Missing Back button in Rebate wizard
- PFPCS-5381 Cost: ValidTo in Cost table does not work if expiryDate in Quote < currentDate (calculationDate)
- PFPCS-5380 PX Price Guidance filter config does not work with CX
- PFPCS-5379 Customer Assigned PL priority not considered when performing Product Price Lookup
- PFPCS-5378 Target Date in Price PX lookup does not work

PFPCS-5374 List Price gets from PX does not work

PFPCS-5370 Wizard: Both name and attribute are displaying together in dropdown list

PFPCS-5369 Wizard - Filter config screen: Error when change existing filter type config from P/C to PX/CX

PFPCS-5367 Incorrect price guidance returned when looking up from PO

PFPCS-5365 PriceGuidance PO wizard: Missing required input

PFPCS-5353 Error message: name is being displayed instead of Message in quote

PFPCS-5349 Historical Data wizard: Missing Revenue Filter Type & Revenue Filter Value

PFPCS-5348 Previous Price result: Showing Invalid Date in Date

PFPCS-5329 Configuration per quote type - default input configuration does not work for specific quote type

PFPCS-5318 Lookup config - add default table structure config

PFPCS-5304 PX Price Guidance lookup - data is not shown

PFPCS-5302 PX price guidance configuration wizard - cannot change PX table

PFPCS-5301 Customer cash discount table structure documentation - add target date

PFPCS-5299 Customer cash discount lookup configuration - Currency config is unnecessary

PFPCS-5297 CPQ_QuoteType configuration should be documented

PFPCS-5296 Documentation - default configurations should be mentioned in the Confluence document

PFPCS-5295 Configuration wizard: Filter builder is not initialized error when apply empty filter config

PFPCS-5294 Lookup Util - groupingProperties not work with generic value

PFPCS-5292 standard discount unused configurations in default data

PFPCS-5291 CPQ Configuration wizard - label for key column is not loaded for selection

PFPCS-5270 Cannot show all PP field in configuration wizard

PFPCS-5188 Old configuration do not removed when using cpq configuration wizard

PFPCS-4879 Lookup keys are not converted to technical name

PFPCS-4878 Configuration Wizard - PX field is not loaded to select when its columns is not renamed

PFPCS-4877 Price Guidance from PO failed to load when using configuration wizard to configure

PFPCS-4833 CPQ wizard - Duplicate configuration for Cost, Competitor table

PFPCS-4815 CPQ Configuration wizard - missing title and description

PFPCS-4814 CPQ Configuration wizard typo

PFPCS-3491 Fix documentation about PO configuration

PFPCS-2910 Configuration wizard - Not showing all columns of table in dropdown