



# Accelerate Sales Compensation

Version 1.5,0

December 2022

# Accelerate Sales Compensation (SC)

The Sales Compensation Accelerator provides a solution for fast implementation of the company's sales compensation plan.

The package includes:

- Common compensation types
- Approval workflow for compensation agreements, agreement records and payout values
- Output templates for documents like Sales Compensation Agreement, configuration overview and progress reports
- Dashboards with information about progress towards the plan
- Dedicated views for compensation agreements and agreement records

In this section:

- [Product Info \(SC\)](#)
- [Get Started \(SC\)](#)
- [Working with Sales Compensation \(SC\)](#)
- [Reference \(SC\)](#)
- [Technical Information \(SC\)](#)
- [Release Notes \(SC\)](#)
- [Archive of Documentation \(SC\)](#)

Search for configuration, manuals or reference information.

## Product Info (SC)

The Sales Compensation Package provides a solution for the fast implementation of the company's sales compensation plan. Its purpose is to maintain, control and evaluate execution with the Pricefx application.

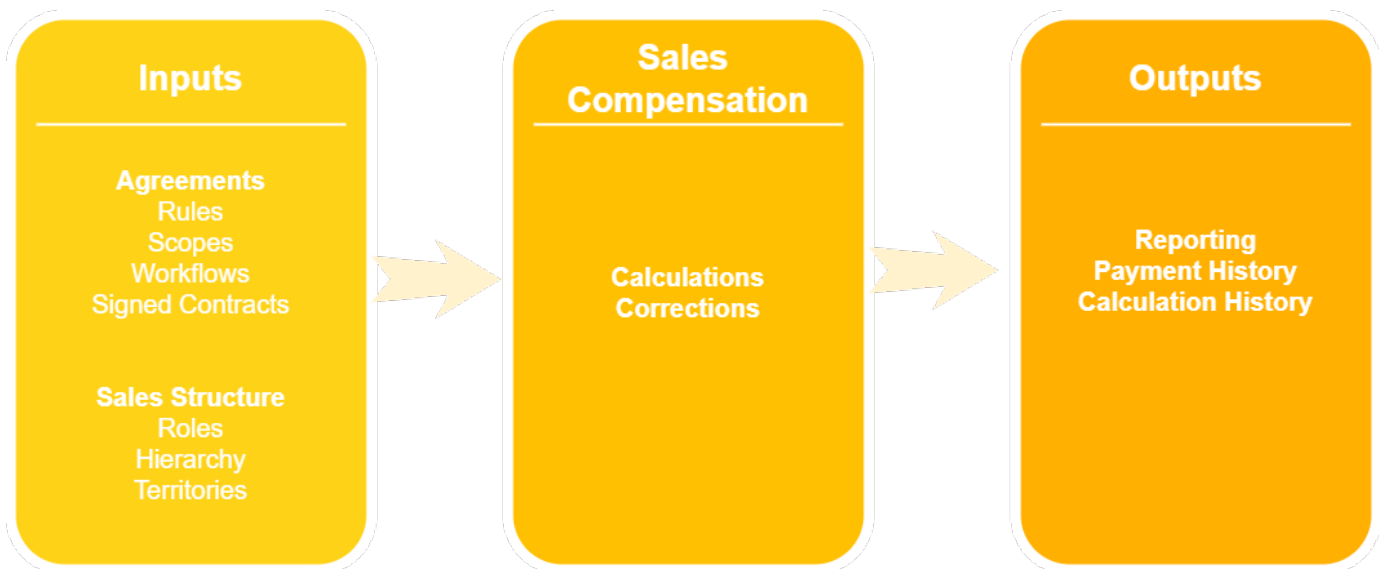
In this section you will find:

- [Sales Compensation Plan](#)
- [Business Roles](#)
  - [Sales Compensation Administrator](#)
  - [Sales Manager](#)
  - [Sales Agent](#)
- [Scope of Usage](#)
  - [Creating Compensation Agreement](#)
  - [Reporting Progress](#)
  - [Plan Maintenance](#)
- [Dashboards and Reporting](#)
  - [Compensation Dashboards](#)
  - [Compensation Plan Report](#)
  - [Overview Report](#)
  - [YTD Report](#)
  - [Transaction Report](#)

## Sales Compensation Plan

It outlines the sales agent's base salary as well as the company's commission and incentive program with strategy. Commission, bonus, and incentive structure incentivize sales force to reach their objectives in order to earn a deserved reward. Final pay mix together makes up On-Target Earnings. In other words, the amount of money a salesperson is paid in the end.

Sales compensations encourage positive behaviors in your sales teams that are necessary to achieve your overall organizational goals and results.



## Business Roles

The following business roles and individuals participate in the processes surrounding the Accelerator's usage activities.

### Sales Compensation Administrator

A person who is responsible for the smooth working of all elements. This person takes the plan from the company's representatives and ensures that individual Sales Managers can create agreements. The person makes sure all salespeople are in the system and individual compensation types and calculation logics that the company uses are present in the system. Sales team changes, adjustments or running plans, contract revisions for a new cycle, or global reporting are activities in their agenda.

Typically, this person is from Finance or SalesOps department closely cooperating with Sales Managers, HR, and legal team.

### Sales Manager

A person who is responsible for the actual creation of individual Compensation Agreements. The person enters conditions into agreements, negotiates details with Sales Agents, and is able to watch progress towards set goals. The Sales Manager also generates reports for those who are not able to access the system on their own. It could be an alternative name for Team Lead.

## **Sales Agent**

A person who actually sells and is also the main recipient of compensations. It could be an alternative name for a Sales Representative, SalesRep, or Salesperson.

## **Scope of Usage**

The Accelerator is a tool used in agendas of the above-mentioned roles. It helps them simplify, automatize, track or define compensation-related work tasks.

## **Creating Compensation Agreement**

Compensation Agreement is one unit of the main Sales Compensation Plan. It can encapsulate one or more rules, conditions, exclusions, and time spans that are used for the calculation of the final sum for the payee. This definition can represent a legal contract.

## **Reporting Progress**

Reporting is an activity that can be done on an agreement level or with a scope of several Sales Agents. Provided reports and dashboards are used for the analysis of the progress of the plan.

## **Plan Maintenance**

Corrections of wrongly accounted transactions, new agreement revisions, definitions for a new period of a year, or replacements in the sales team are common activities associated with a well-defined plan.

## **Dashboards and Reporting**

The ability to accurately track and compare progress towards defined goals is an important aspect of successful strategy execution. The package is distributed with a dashboard and set of reports supporting those processes.

## **Compensation Dashboards**

Interactive reporting in dashboards is used for a deeper understanding of compensation agreements.

Users are restricted to see data of their subordinates and their own. You can read more on [Dashboards Description \(SC\)](#).

## **Compensation Plan Report**

Here is an example of a basic agreement between the Manager and Sales Agent:



CO-551\_1670239872772.pdf

### Overview Report

A document summarizing definitions of an agreement with results for each agreement line and total compensation value. This is the one-pager for the agreement in numbers.

### YTD Report

It is a simple document showing agreement progress in the current calendar year. It goes into fewer details about each agreement line but focuses on results for the year.

### Transaction Report

This report is used for checking all transactions that are included in the calculation of the selected compensation agreement.

It can be used as an audit supporting tool or document generated when an agreement is fulfilled, see example [SC\\_Transaction\\_List\\_Report\\_SAMPLE.pdf](#).

## Get Started (SC)

- [Installation \(SC\)](#)
- [Configuration \(SC\)](#)

### Installation (SC)

This tutorial will guide you through the installation of the Sales Compensation Accelerator.

In this section:

- [Pre-requisites](#)
- [Installation Steps](#)
  - [Select Partition for Deployment](#)
- [Post-installation Steps](#)
  - [Add Indexing](#)
  - [Enable React UI](#)

- [Schedule Plan Calculation Task](#)
- [Schedule Record Calculation Task](#)
- [Assign Access to Users](#)

## Pre-requisites

Before you start, ensure that you have:

- Access to a partition on the Pricefx server (9.0 or newer). You will need:
  - Server URL
  - Partition name
  - Username and password for a partition user with sufficient rights for using the Accelerator
  - License on the partition must cover the Analytics and Sales Compensations modules
- Access to Pricefx PlatformManager
  - Username and password for PlatformManager user
  - The user must have the following permissions for your partition (to which you plan to deploy the Accelerator):
    - Permission *Marketplace Templates - deploy*
- Transaction data in the Datamart structure with required fields containing the following data:

Requirement description	Deployment label
The field name of a Value Base field is used to calculate the base value. In most cases, it will be a revenue, margin or gross margin field.	Baseline Value
The field name of the Margin field.	Margin
The field name contains an identification of a sales agent.	Seller Id
The field name contains a customer ID in the Datamart.	Customer Id
The field name contains a product ID in the Datamart.	Product Id
The field name contains a product name. It is used for the <i>Transaction List</i> report only.	Product Name
The field name contains a pricing date in the Datamart.	Pricing Date
The field contains the main Datamart currency.	Currency
The field name of the Invoice Price field	Invoice Price
The field name of the Quantity field	Quantity

The package contains several components and their technical description is provided on the [Components \(SC\)](#) page.

## Installation Steps

### Select Partition for Deployment

1. In PlatformManager, navigate to **Marketplace > Accelerator Packages**, find the *Sales Compensation*.
2. Click **Deploy** and select a partition to which you want to deploy the package.
3. Click **Deploy**.

4. A warning dialogue will appear. After you read the warning text and you agree with the conditions, you can click **Continue**.
  - ❗ If you need to leave the deployment process before it is finished, you can always come back later. The wizard will offer you to either start again, or continue in the previously started process.
5. Set up Datamart mapping of required fields from Datamart and default values for a few parameters.

## Settings

Source Type \*

Source Name

Baseline Value \*

Invoice Price \*

Quantity \*

Seller Id \*

Customer Id \*

Product Id \*

Product Name \*

Pricing Date \*

Margin \*

Currency \*

Currency ▼

**Payout Days**

4

**Payment Period**

Quarterly ▼

**Quota For**

Annual ▼

**Deposit Scheme**

Cumulative ▼

**Continue** **Cancel**

## 6. Upload the seller master data and define the mapping.

### Data Mapping

Sample from your uploaded DataExport-1656008082678.csv file. Your file contains 7 lines:

Seller Id String	Seller Name String	Reports To String	First Name String	Surname String	Pricefx User Account Id String	Active Boolean <span>▼</span>
SC-001	Seller 001		Grover	Grainger	jakub	false
SC-002	Seller 002	SC-001	Levy	Anje	giang1	false
SC-003	Seller 003	SC-002	Aniela	Bauer	michal	false

### Parsing Options

Separator\* ▼ ; (Semicolon)

Quote character "

Escape character \

Decimal Separator ▼ .

Date Format yyyy-MM-dd

**Continue** **Cancel**

- The deployment is complete. Go to your partition and continue with the manual steps required after deployment.

## Post-installation Steps

After the package is deployed to your partition and all automatic installation steps are done, you need to do a few manual tasks before you start with configuration and package adjustments to your specific business needs.

## Add Indexing

To avoid performance issues when running Calculation Tasks, indexing for COARs on the partition should be added, for these fields: `accrualType`, `attribute1`, `attribute20`.

Example command:

```
MariaDB [pricfx_templates_dev]> create index typeAttr1Attr20 on  
CompensationAccrualRecord(partition_id, accrualType, attribute1, attribute20)  
using BTREE;
```


## Enable React UI

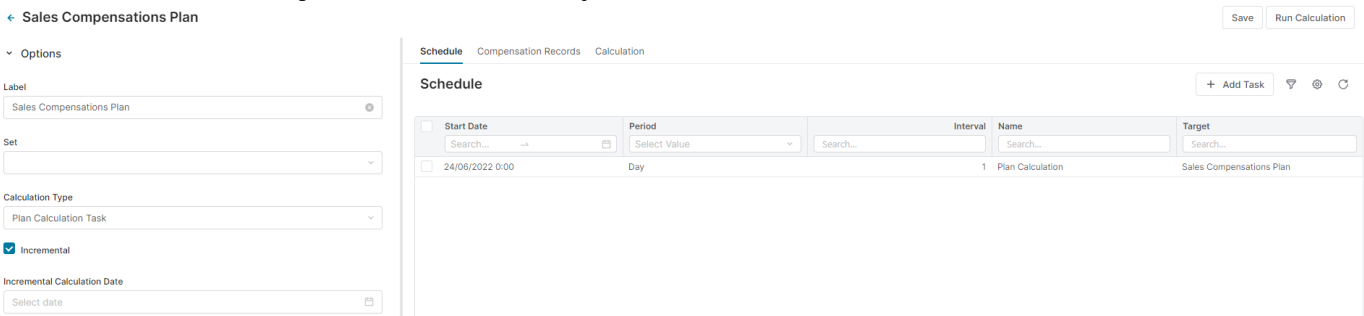
The package is designed to work with the latest UI engine in the Sales Compensations module. For proper functionality is necessary to enable it manually.

1. From the **Administration** menu go to the **Feature Flags** section.
2. Activate the following Feature Flags:  
`dashboard.useReactDashboard`  
`useReactFor.advancedFilter`
3. Log out and log back in for the changes to take effect.

## Schedule Plan Calculation Task

In this step, you will create a task for the plan calculation scheduler.

1. Go to **Sales Compensations > Calculations**.
2. Add a new calculation.
3. Enter a label, e.g. 'SC\_Calculation'.
4. Select a set 'Sales Compensations' and click the **Add** button.
5. Click the label of the newly added line.
6. Go to the Calculation tab.
7. As Logic/Formula choose '[SC] Sales Compensation'
8. As Feeder Formula choose '[SC] Compensation Record Calculation Feeder'.
9. Set the Calculation Type to Plan Calculation Task.
10. Optionally, enter StartDate and EndDate.  
 Leave empty unless you need to calculate some specific time frame.
11. Click **Save** in the left upper corner.
12. Enable **Incremental** calculation on the left side of your screen.
13. Schedule the calculating task according to your data, e.g. daily.
14. In the Overview, you can schedule this job to run at a suitable interval.



Start Date	Period	Interval	Name	Target
24/06/2022 0:00	Day	1	Plan Calculation	Sales Compensations Plan

## Schedule Record Calculation Task

The steps for creating the Record calculation plan are almost the same as for Plan calculation, the only difference is in step 9 where the Calculation Type should be set to Record Calculation Task.

## Assign Access to Users

The package comes with three predefined business roles which speed up configuration. You can adjust their default configuration and assign them to the users who have access to the system.

A detailed description of role definitions and their meaning is described in [Product Info \(SC\) | Business roles](#).

1. From the **Administration** menu go to the **Access Admin > Business Roles Admin** section.
2. Check if all roles with the '[SC]' prefix correspond with your expectations for minimal access and adjust them to your use cases.
3. Assign user accounts to their business roles from the Users tab of each role.

At this point, you completed the installation. Now, you should configure the package for specific business needs and get familiar with its use. The last thing to consider is the integration with a system that processes output compensations.

## Configuration (SC)

This tutorial will guide you through the configuration steps. After you have completed the installation, you can start with the configuration of the Sales Compensation (SC) Accelerator. You can customize the default configuration, extend types or remove those that are not needed.

In this section:

- [Global Package Configuration](#)
  - [Company Parameters](#)
- [Plan Header Types](#)
  - [How to Customize Your Types](#)
- [Condition Types](#)
  - [How to Customize Your Types](#)
  - [How to Add a New Type](#)
- [Define Seller Information in Compensation Header](#)
  - [How to Customize Your Seller Information Display](#)
- [Define Adjustment Wizard Transaction Field in Adjustment Form](#)
  - [How to Customize Your Adjustment Wizard Transaction Field Display](#)
- [Define Approval Workflows](#)
  - [Compensation Plan Approval Workflow](#)
  - [Compensation Record Approval Workflow](#)
  - [Adjustment Custom Form Workflow](#)

### Global Package Configuration

The global configuration data for the package are stored in an element of Advanced Configuration Options. Many of those parameters can be overridden on the Condition Type.

1. From the **Administration** menu go to the **System Configuration** section.
2. Switch to **Advanced Configuration Options**.
3. Search for the `SC_AdvancedConfiguration` element where you can adjust the parameters listed below.
4. Adjust configuration keys according to your specifications.

A list of all parameters, their descriptions, possible values, and other notes are in [Components \(SC\) | Advanced Configuration Options](#).

Sample of 'SC\_AdvancedConfiguration':

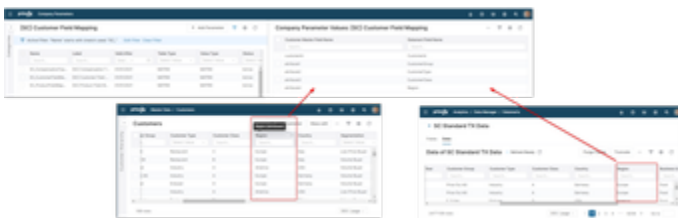
```
{
  "sourceType": "DM",
  "sourceName": "SC_TransactionData",
  "pricingDateFieldName": "PricingDate",
  "customerIDFieldName": "CustomerId",
  "productIDFieldName": "ProductId",
  "productNameFieldName": "ProductName",
  "baselineFieldName": "InvoicePrice",
  "invoicePriceFieldName": "InvoicePrice",
  "quantityFieldName": "Quantity",
  "marginFieldName": "GrossMargin",
  "sellerIDFieldName": "SellerId",
  "currencyFieldName": "Currency",
  "payoutDays": 2,
  "paymentPeriod": "Quarterly",
  "quotaFor": "Annual",
  "depositScheme": "Non-Cumulative"
}
```

### Company Parameters

To be able to apply a Customer filter input and Product filter input in a compensation plan, you need to update the mapping fields between Product/Customer master and Datamart fields.

Based on the Datamart structure and Product/Customer master in your partition, you need to update or add more mapping in two Company Parameters:

- SC\_CustomerFieldMapping
- SC\_ProductFieldMapping



### Plan Header Types

#### How to Customize Your Types

1. Go to **Sales Compensations > Compensation Types**.
2. Select the line you want to edit.
3. Click the Edit button to open the customization dialogue.

4. Adjust parameters according to your needs by replacing default values with custom logic.
5. Click Save Changes to save and confirm changes.

A list of all parameters, their descriptions, possible values, and other notes are in [Components \(SC\)](#).

Sample of the 'Plan Header Type' which is deployed with the package:

```
Name (uniqueName): Individual Compensation
Header Logic (headerFormulaName): Sales Compensation Header
Compensation Plan Workflow (workflowFormulaName): Sales Compensation
Plan
Compensation Record Workflow (scRecordWorkflowFormulaName):
Compensation Record
```


## Condition Types

This chapter describes procedures for manipulation with Lineltem elements which are primary blocks of calculation for compensations inside the individual plan.

### How to Customize Your Types

1. Go to **Sales Compensations > Condition Types**.
2. Look for items with the 'Sales Compensation' in the 'Pricing Logic (formulaName)' column.
3. Select the line you want to edit.
4. Click the Edit button to open the customization dialogue.
5. Adjust parameters according to your needs by replacing default values with custom ones.
6. Click Save Changes to confirm changes.

### How to Add a New Type

1. Go to **Sales Compensations > Condition Types**.
2. Click 'Add Condition Type' and a configuration dialogue will open.
3. Fill in fields you want to have specifically defined in the condition type.
  -  Be aware, parameters can inherit default values from Advanced Configuration Options. Empty field means the parameter inherits a value.
  - "Condition Type Name" shows a list of Sales Compensation. Options compatible with this package are listed in [Type Overview \(SC\) | Condition Types](#).
4. Click Add to save your new type.

A list of all parameters, their descriptions, inheritance, possible values, and other notes are in section [Components \(SC\) | Condition Type Attributes](#).

## Define Seller Information in Compensation Header

Add a new read-only input which will show the detailed information about selected Seller input on the Compensation Plan Header.

### How to Customize Your Seller Information Display

1. Go to Company Parameters.
2. Go to Pricefx Accelerators parent folder Sales Compensation folder.
3. Select SC\_HeaderSellerDetailsMapping Company Parameter name.

4. Add a new record or adjust the existing one to display which the seller information you want.

Sample of SC\_HeaderSellerDetailsMapping which is deployed with the package:

### Company Parameter Values: Header Seller Details Mapping



<input type="checkbox"/> Field Label	Seller Field
<input type="text" value="Search..."/>	<input type="text" value="Search..."/>
<input type="checkbox"/> Seller ID1	sellerId
<input type="checkbox"/> Seller Name	name
<input type="checkbox"/> Reports To	reportsTo
<input type="checkbox"/> First Name1	attribute1
<input type="checkbox"/> Last Name	attribute2
<input type="checkbox"/> Termination Date	attribute5

### Define Adjustment Wizard Transaction Field in Adjustment Form

You can display any transaction field in the Adjustment Form (Adjustments under the Sales Compensation module).

How to Customize Your Adjustment Wizard Transaction Field Display

1. Go to Company Parameters.
2. Go to Pricefx Accelerators parent folder Sales Compensation folder.
3. Select SC\_AdjustmentWizardTransactionFieldMapping Company Parameter name.
4. Add a new record or adjust the existing one to display which the Adjustment Field Name you want in the Adjustment Form.

Sample of SC\_AdjustmentWizardTransactionFieldMapping which is deployed with the package:

### Company Parameter Values: SC Adjustment Wizard Transaction Field Mapping

<input type="checkbox"/> Adjustment Field Name	Datamart Field Name
<input type="text" value="Search..."/>	<input type="text" value="Search..."/>
<input type="checkbox"/> Customer ID	-ADJUST_ME-
<input type="checkbox"/> Invoice Price	-ADJUST_ME-
<input type="checkbox"/> Margin	-ADJUST_ME-
<input type="checkbox"/> Pricing Date	-ADJUST_ME-
<input type="checkbox"/> Product ID	-ADJUST_ME-
<input type="checkbox"/> Quantity	-ADJUST_ME-
<input type="checkbox"/> Seller ID	-ADJUST_ME-

### Define Approval Workflows

Two types of approval workflows can be configured to support business conditions and flow inside an organization. The installation package already deployed and configured workflow types (SCA and SCR,

Adjustment) in **Administration > Logics > Workflow Logics > General Workflow Logics**. You just need to convert your business rules to workflow parameters in Company Parameters.

### Compensation Plan Approval Workflow

In this step, you should configure workflow conditions for the Compensation Plan. You will follow detailed steps from the manual for the [Accelerate Approval Workflow Package \(AWP\)](#) which is part of this package.

1. Go to **Company Parameters** in the top menu.
2. Locate PFXTemplate > Approval Workflow.
3. Define workflow steps in the 'ApprovalWorkflowsSetup' parameter.
  - 'Workflow Type' must be *SCA*.
4. Define conditions in the 'ApprovalConditions' parameter.
  - The list of variables is on the [Workflow Variables \(SC\) | Compensation Agreement approval workflows](#) page.
5. Define approvals in the 'Approvers' parameter.
6. Test your workflow.
  - You don't need to submit a plan to test it. Just recalculate and verify that the workflow is prepared to be triggered.

Example for a two-step workflow for Compensation Plan with conditions and approvals:

The screenshots show the configuration of a two-step workflow for Compensation Plan. The first screenshot shows the 'Approval Workflow Setup' parameter with a table of workflow steps:

Workflow Type	Step ID	Step Order	Step Label	Reason
SCA	1	1	Senior Sales Management	Management approval needed
SCA	2	2	VP of Sales	VP level needed
SCR	1	1	SalesOps	Compensation too high

The second screenshot shows the 'Approvers at each step' parameter with a table of approvers:

Workflow Type	Step ID	Approver ID	Approver Type	Approver Name	Skip
SCA	1	1	BUSINESS ROLE	DEMO_SC_SSM	No
SCA	2	2	USER	john.doe	No
SCR	1	1	BUSINESS ROLE	DEMO_SC_SSM	No

The third screenshot shows the 'Approval conditions at each step' parameter with a table of conditions:

Workflow Type	Step ID	Condition ID	Condition Description	Condition	Skip
SCA	1	1	Previous Compensation 10 000+	"line.PreviousCompensation" > 10000	No
SCA	2	1	Compensation is more than 15 %	"line.Compensation %" > 15.0	No
SCR	1	1	Compensation is more than 90	attribute17 > 90.0	No

### Compensation Record Approval Workflow

This workflow type configuration is identical to the one above. It uses **SCR** as the 'Workflow Type'.

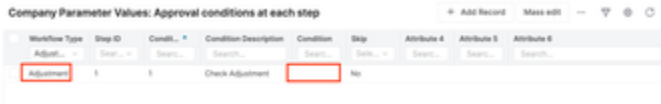
An example from the previous step shows one step workflow configuration. The list of variables is on the [Workflow Variables \(SC\) | Agreement Record approval workflows](#) page.

• To check the approvals for Compensation Records, you must approve the Compensation Plan first.

### Adjustment Custom Form Workflow

This workflow type configuration is identical to the one above. It uses **Adjustment** as the 'Workflow Type'. An example from the previous step shows one step workflow configuration.

In the current development, you only need to define the approval condition as empty to display the approval step:



## Working with Sales Compensation (SC)

- [Business Roles \(SC\)](#)
- [Advanced Configuration \(SC\)](#)
- [Sales Compensation Walkthrough \(SC\)](#)
- [Adjustments](#)
- [Dashboards Description \(SC\)](#)

### Business Roles (SC)

To use the Sales Compensation Accelerator, you need to have the following Pricefx [user roles](#):


Sales Agent	Sales Manager	Sales Admin
<ul style="list-style-type: none"> <li>• View Customers</li> <li>• View Products</li> <li>• View Product Details</li> <li>• View Custom Form</li> <li>• View Dashboards</li> <li>• Export Dashboards</li> <li>• Run Analytics Queries</li> </ul>	<ul style="list-style-type: none"> <li>• View Customers</li> <li>• View Products</li> <li>• View Product Details</li> <li>• View Sellers</li> <li>• View Seller Extensions</li> <li>• Manage Custom Form</li> <li>• Manage Compensation Plans</li> <li>• Manage Compensation Records</li> <li>• View Dashboards</li> <li>• Export Dashboards</li> <li>• Run Analytics Queries</li> </ul>	<ul style="list-style-type: none"> <li>• View Customers</li> <li>• View Products</li> <li>• View Product Details</li> <li>• Manage Sellers</li> <li>• Manage Seller Extensions</li> <li>• Manage Custom Form</li> <li>• Administer Compensation Plans</li> <li>• Administer Compensation Records</li> <li>• Administer Sales Compensations Module</li> <li>• Edit Global Preferences</li> <li>• Manage Workflows</li> <li>• View Dashboards</li> <li>• Export Dashboards</li> <li>• Run Analytics Queries</li> </ul>

### Advanced Configuration (SC)

The main place for package configuration is an Advanced Configuration option named [SC\\_AdvancedConfiguration](#). Option values are configured mostly during a package deployment process (refer to [installation](#)) and can be adjusted later here. Some options can be overridden on a Condition Type or Line Item level where the lower level has a priority.

The following table shows a list of all configuration parameters stored in SC\_AdvancedConfiguration:

Name	Description	Values	Condition Type overwritable
sourceType	Type of source data.	Select one of these options: <ul style="list-style-type: none"> <li>DM - Datamart</li> <li>DS - Data Source</li> </ul>	✓
sourceName	Name of the source used to query and allocate the compensation value.		✓
pricingDateFieldName	Contains a pricing date in the Source Name.		✓
customerIDFieldName	Contains a customer ID in the Source Name.		✓
productIDFieldName	Contains a product ID in the Source Name.		✓
productNameFieldName	Contains a product name. It is used for the <i>Transaction List</i> report only.		
baselineFieldName	The field name of a Value Base field is used to calculate the base value. In most cases, it will be a revenue or margin field.		✓
invoicePriceFieldName	Contains an Invoice Price in the Source Name. It is used for the Adjustments		✓
quantityFieldName	Contains a Quantity in the Source Name. It is used for the Adjustments		✓
marginFieldName	Contains a Margin in the Source Name. It is used for the Adjustments		✓
sellerIDFieldName	Contains an identification of a sales agent.		✓
currencyFieldName	Contains the main Source Name (DM or DS) currency.		
payoutDays	Number of days after the end date of the payment period when the payout happens.		✓
paymentPeriod	Sets a frequency of payouts. The user can specify it also on a Line Item level.	Select one of these options: <ul style="list-style-type: none"> <li>Monthly</li> <li>Quarterly</li> <li>Semi-Annually</li> <li>Annually</li> </ul>	✓
quotaFor	The target defined on the line item is evaluated for each payment period (see <i>paymentPeriod</i> ) or annually.	Select one of these options: <ul style="list-style-type: none"> <li><b>Payment Period</b> - Gets a base value for every period, then compares</li> </ul>	✓

		<p>and calculates the compensation.</p> <ul style="list-style-type: none"> <li>• <b>Annual</b> - Gets a base value for the whole year, then compares and calculates the compensation.</li> </ul>	
<b>depositScheme</b>	Parameter providing guidance on compensation accumulation across payment periods.	<p>If "quotaFor" is "Annual", you need to define how to calculate the compensation value:</p> <ul style="list-style-type: none"> <li>• <b>Non-Cumulative</b> - The compensation value is calculated for the current period based on a cumulation of the base value, then divided by the number of periods (12 months or 4 quarters, 2 for semi-annually, 1 year).</li> <li>• <b>Cumulative</b> - The same as with calculation of Non-Cumulative above, but the compensation value of the previous period is excluded.</li> </ul>	

Example of SC\_AdvancedConfiguration:

```

{
  "sourceType": "DM",
  "sourceName": "SC_TransactionData",
  "pricingDateFieldName": "PricingDate",
  "customerIDFieldName": "CustomerId",
  "productIDFieldName": "ProductId",
  "productNameFieldName": "ProductName",
  "baselineFieldName": "InvoicePrice",
  "invoicePriceFieldName": "InvoicePrice",
  "quantityFieldName": "Quantity",
  "marginFieldName": "GrossMargin",
  "sellerIDFieldName": "SellerId",
  "currencyFieldName": "Currency",
  "payoutDays": 2,
  "paymentPeriod": "Quarterly",
  "quotaFor": "Annual",
  "depositScheme": "Non-Cumulative"
}

```

Explanation of selected parts of the configuration above:

**Example 1:**

```
"payoutDays": 2,  
"paymentPeriod": "Quarterly",  
"quotaFor": "Payment Period",  
"depositScheme": ""
```

The layout on the line item of a Compensation Plan (see details in [Sales Compensation Walkthrough](#)) based on the configuration:



# Flat Bonus

## Input Parameters

Product(s)

Product Class: A



**Quota \***

100

**Target Incentive \***

15

**Payment Period \***

Quarterly



**Quota For**

Payment Period

**Deposit Scheme**

**Baseline Field ?**

Invoice Price

- Quota input (100) and Target Incentive (15) are understood as per Payment Period input (in this case, it is Quarterly by default, you can select another input such as Monthly, Semi-Annually, or Annually).
- Then, you compare those inputs with your baseline value in the Source Name, Source Type (e.g. SC\_TransactionData DM) for each payment period (Q1, Q2, Q3, Q4) and calculate the compensation based on the condition type you selected.
- End of each payment period (Q1, Q2, Q3, Q4), the payout date is planned to pay for the sales agent at the second day (2) later, excluding the weekends.

**Example 2:**

```
"payoutDays": 4,  
"paymentPeriod": "Quarterly",  
"quotaFor": "Annual",  
"depositScheme": "Non-Cumulative"
```

# Flat Bonus

## Input Parameters

Customer(s)

Select Customer(s) ▼



Product(s)

Product Class: A



Quota \*

130

Target Incentive \*

20

Payment Period \*

Quarterly ▼

Quota For

Annual

Deposit Scheme

## Non-Cumulative

- Quota For is Annual Quota input (130) and Target Incentive (20) are understood as a whole year (Annual), it should be the Start Date - End Date on the Compensation Plan (see details in [Sales Compensation Walkthrough](#)).
- You compare those inputs with your cumulated baseline value in the Source Name, Source Type (e.g. SC\_TransactionData DM) for payment period (Q1, Q2, Q3, Q4).
- For calculation details for this case see <https://pricefx.atlassian.net/wiki/spaces/ACCDEV/pages/3807543836/Type+Overview+SC#Agreement-Records-Calculation>.
- End of each payment period (Q1, Q2, Q3, Q4), the payout date is planned to pay for the sales agent at the fourth day (4) later, excluding the weekends.

### Summary

There are 3 different calculation ways:

1. Quota For: **Payment Period**
2. Quota For: **Annual** - Deposit Scheme: **Non-Cumulative**
3. Quota For: **Annual** - Deposit Scheme: **Cumulative**

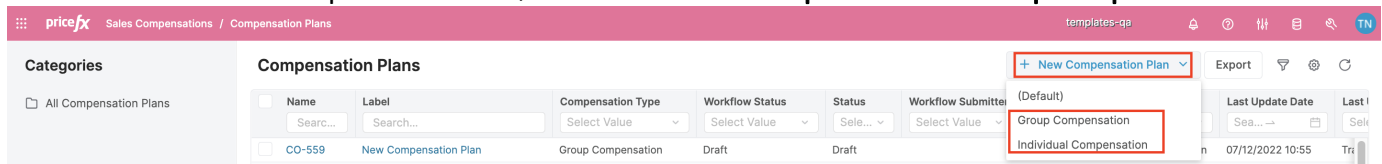
### Sales Compensation Walkthrough (SC)

This section helps you understand how a sales compensation flow is created and used in the Sales Compensation module.

- [A. Compensation Plan](#)
- [B. Calculations Job](#)
- [C. Compensation Records](#)

#### A. Compensation Plan

1. Go to **Sales Compensations > Compensation Plan**.
2. Create a New Compensation Plan, either **Individual Compensation** or **Group Compensation**.



The screenshot shows the 'Compensation Plans' section in the Pricefx application. A dropdown menu for '+ New Compensation Plan' is open, with 'Individual Compensation' selected. The table below shows a list of compensation plans.

Name	Label	Compensation Type	Workflow Status	Status	Workflow Submitter	Last Update Date	Last Update By
CO-559	New Compensation Plan	Group Compensation	Draft	Draft		07/12/2022 10:55	Tr

#### 1. Header Level View

On the header level, the package provides standard input entries, summarized line item compensation and forecast value, also grouped by Condition Type.

*User Inputs*

## Inputs

Start Date \*

06/12/2022



End Date \*

06/12/2022



External Reference

User Group (View Details)

Search...



User Group (Edit)

Search...



## Input Parameters

Customer(s)

Select Customer(s)



Payee \*

Select Seller



Payout Currency \*

EUR



- **Start Date**<sup>(\*)</sup> - Enter the start date of the period for the analysis.
- **End Date**<sup>(\*)</sup> - Enter the end date of the period for the analysis.
- **Customer(s)** - Allows you to choose one or more customers for the analysis.
- **Payee**<sup>(\*)</sup> - Allows you to select a seller if it is an individual compensation.
- **Team**<sup>(\*)</sup> - Allows you to select a team if it is a group compensation.
- **Payout Currency**<sup>(\*)</sup> - Allows you to select a payout currency and affects to all the calculation values. The currency list displays according to the condition: CcyFrom = DM Currency and CcyValidFrom <= startDate <= CcyValidTo  
The values are taken from:
  - Currencies DS - CcyFrom, CcyValidFrom, CcyValidTo
  - Compensation Plan - Start Date
- **General Filter** - Allows you to set up a generic transaction data filter.
- **Forecast Method**<sup>(\*)</sup> - Allows you to select a Forecast Method to calculate Compensation Forecast:
  - Seasonal Naive
  - Average

- **Sales Goal Increase %<sup>(\*)</sup>** - Input number

(\*) means this is a mandatory field

#### Calculated Results

▼ Compensation Detail	
Label	Calculation Result
Previous Compensation	870,791.53
Current Compensation	949,627.82

▼ Forecast	
Label	Calculation Result
Compensation Forecast	448,457.64
True-up	-26,356.27
Compensation Forecast Comparison Chart	<a href="#">Show</a>
This Year's Sales Forecast Table	<a href="#">Show</a>
This Year's Sales Forecast Chart	<a href="#">Show</a>

▼ Flat Bonus	
Label	Calculation Result
Current Compensation	949,627.82
Compensation Forecast	896,915.27

- Compensation Detail
  - Previous Compensation
  - Current Compensation
- Forecast
  - Compensation Forecast
  - True-up
  - Compensation Forecast Comparison Chart
  - This Year's Sales Forecast Table
  - This Year's Sales Forecast Chart
- <Grouped by Condition Type>
  - Current Compensation
  - Compensation Forecast

#### 2. Line Item Level

On the line item level, the package provides standard input entries, compensation and forecast value, charts, compensation type info.

1. Go to **Items** tab.
2. Click the **Add Items** button.

3. Choose the Condition Type you want to add and enter values at the Line Item Level. There are total 16 condition types, for details see [Type Overview \(SC\)](#).

**i** When any of the Sellers defined in the plan are missing a Salary (configured in the Seller Salaries SX) in the Payout Currency of the plan, the Variable Pay Linear Amount, Variable Pay Multi Target Amount, and Variable Pay Stepped Amount condition type are hidden from the items selection in the Compensation Plans.


You can set the following inputs:

*User Inputs*

# Flat Bonus

## Input Parameters

Customer(s)

Product(s)

Quota \*

Target Incentive \*

Payment Period \*

Quota For

Annual

Deposit Scheme

Non-Cumulative

## Baseline Field ?

### Invoice Price

- **Customer(s)** - Allows you to choose one or more customers for the analysis. It overrides the customer input on the header level.
- **Product(s)** - Allows you to choose one or more products for the analysis. The inputs are generated based on each condition type you selected on the line item.
- **Payment Period** - Allows you to choose a payment period. The options are:
  - Monthly
  - Quarterly
  - Semi-Annually
  - AnnuallyBy default, it is taken from [SC\\_AdvancedConfiguration](#).
- **Quota For** - Read-only, taken from [SC\\_AdvancedConfiguration](#).
- **Deposit Scheme** - Read-only, taken from [SC\\_AdvancedConfiguration](#).
- **Baseline Field** - Read-only, taken from [SC\\_AdvancedConfiguration](#).

**i** In case, you added a line item and then changed some values (Quota For, Deposit Scheme or Baseline Field) in [SC\\_AdvancedConfiguration](#), the read-only inputs are not updated accordingly. We suggest that you should NOT change this configuration after the deployment but if it is necessary, you need to delete an existing line item and add it again for the update.

#### *Calculated Results*

- **Compensation Detail**
  - Baseline Value Field Label - Shows the field label to use for calculating the Baseline Value.
  - Previous Baseline Value - Total sales for a validity period of the agreement - 12 months.
  - Current Baseline Value - Sum of the Current Baseline Value based on the Start Date - End Date.
  - Previous Compensation - Compensation value based on sales from the previous period (current contract validity - 12 months) and current quota input conditions.
  - Current Compensation - Sum of the current compensation from the current period and current quota input conditions.
- **Forecast**
  - Forecast Method
  - Sales Goal Increase %
  - Forecasted Baseline Value - Sum of the Forecasted Baseline Value
  - Compensation Forecast - Sum of the Compensation Forecast
  - True-up - Equals to Compensation Forecast - Current Compensation
  - Compensation Forecast Comparison Chart
  - This Year's Sales Forecast Table
  - This Year's Sales Forecast Chart
- **Compensation Type Info**
  - Quota For
  - Deposit Scheme

Calculations	
<ul style="list-style-type: none"> <li>Payment Period</li> <li>Payee Name(s)</li> <li>Customer(s)</li> <li>Product(s)</li> <li>General Filter</li> <li>Payout Currency</li> </ul>	
<ul style="list-style-type: none"> <li>Color coding: <ul style="list-style-type: none"> <li>Previous Compensation: <b>blue</b></li> <li>Current Compensation &gt; Previous Compensation: <b>green</b>, else <b>red</b></li> <li>Compensation Forecast &gt; Current Compensation: <b>green</b>, else <b>red</b></li> <li>If value does not change, it will be <b>orange</b>.</li> </ul> </li> </ul>	
<h3>3. Input Validation</h3> <p>The system will show a warning message if:</p> <ul style="list-style-type: none"> <li>Input for the required fields on the header/line level is missing.</li> <li>Input for the required field that is less than zero.</li> <li>Seller termination date (configured in the Sellers master data) is less than Start Date input on the Compensation Plan.</li> </ul> <p>Sample of input validation:</p> <div style="border: 1px solid orange; padding: 5px; background-color: #fff9e6;"> <p><b>!</b> InputValidation : INPUT VALIDATION:: Fla Quota.</p> <p>InputValidation : INPUT VALIDATION:: Single or equal to zero</p> <p>InputValidation : INPUT VALIDATION:: Single greater than or equal to zero</p> <p>InputValidation : INPUT VALIDATION:: Variab must be greater than zero</p> <p>Compensation plan calculated with errors</p> </div>	
<p>After the details on the header and line item level are filled in and the inputs are validated, the final step is to submit the Compensation Plan.</p>	
<h2>B. Calculations Job</h2> <ol style="list-style-type: none"> <li>Go to <b>Sales Compensation &gt; Calculations</b>.</li> </ol>	

### Calculations

#### Compensation Detail

Label	Calculation Result
Baseline Value Field Label	Invoice Price
Previous Baseline Value	3,627,661.13
Current Baseline Value	4,091,207.21
Previous Compensation	435,395.76
Current Compensation	474,813.91

#### Forecast

Label	Calculation Result
Forecast Method	Seasonal Naive
Sales Goal Increase %	3.00%
Forecasted Baseline Value	3,736,490.97
Compensation Forecast	448,457.64
True-up	-26,356.27
Compensation Forecast Comparis...	Show
This Year's Sales Forecast Table	Show
This Year's Sales Forecast Chart	Show

#### Compensation Type Info

Label	Calculation Result
Quota For	Annual
Deposit Scheme	Non-Cumulative
Payment Period	Quarterly
Payee Name(s)	Show
Customer(s)	('Region' = "America")
Product(s)	('ProductClass' = "A")
General Filter	'Region' = "Europe"
Payout Currency	EUR



Dashboards

> Pricefx Processes

> Analytics

> Price Setting

> Quoting

> Deal Planning

> Agreements & Promotions

> Rebates & Compensations

✓ Sales Compensations

Compensation Types

Condition Types

Compensation Plans

Calculations



Compensation Records

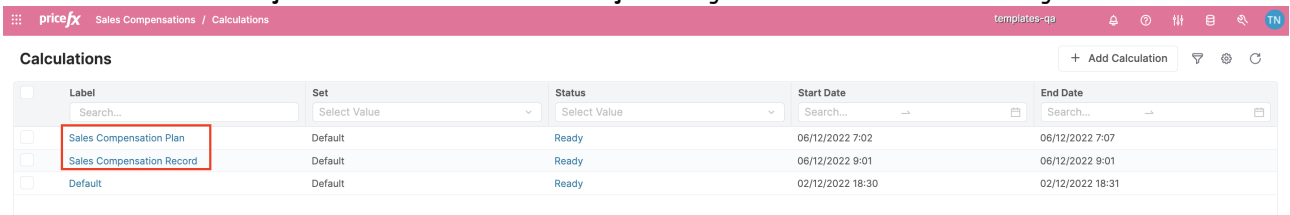
# Accrual Records

# Adjustments

> Channel Management

> Optimization

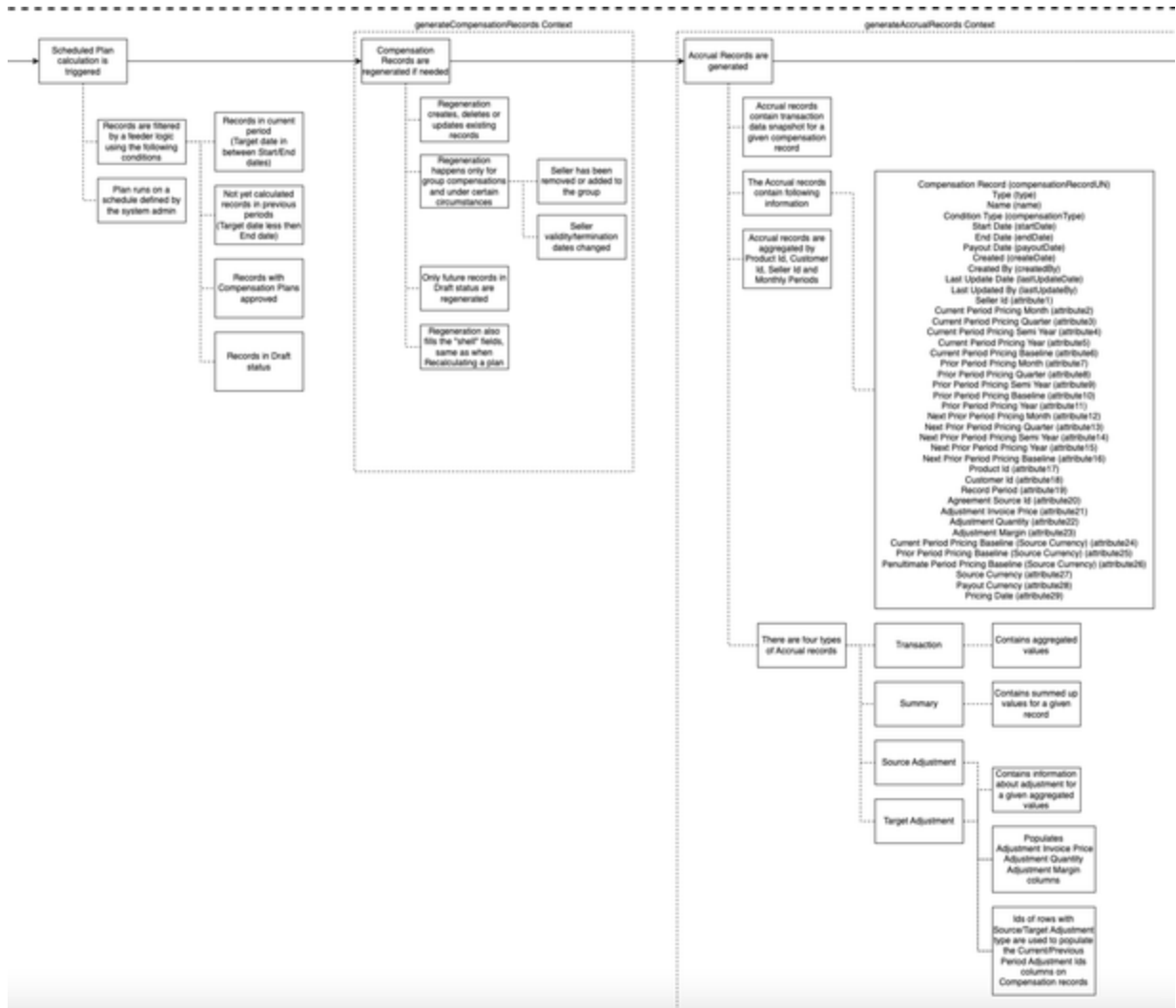
2. Run the below listed jobs . There are scheduled jobs or you can run them manually.



The screenshot shows the 'Calculations' section of the Pricefx interface. It features a table with columns for Label, Set, Status, Start Date, and End Date. The 'Sales Compensation Plan' and 'Sales Compensation Record' rows are highlighted with a red box. The 'Default' row is also visible.

Label	Set	Status	Start Date	End Date
Search...	Select Value	Select Value	Search...	Search...
Sales Compensation Plan	Default	Ready	06/12/2022 7:02	06/12/2022 7:07
Sales Compensation Record	Default	Ready	06/12/2022 9:01	06/12/2022 9:01
Default	Default	Ready	02/12/2022 18:30	02/12/2022 18:31

- **Sales Compensation Plan** - This job (re)generates Compensation Records and Accrual Records. The results in the Compensation Records are still empty.



- After the job finishes, you can go to the Accrual Records for checking:
  - Go to **Sales Compensation > Accrual Records**.

| Dashboards

- > Pricefx Processes
- > Analytics
- > Price Setting
- > Quoting
- > Deal Planning
- > Agreements & Promotions
- > Rebates & Compensations

∨ Sales Compensations

Compensation Types

Condition Types

Compensation Plans

Calculations

Compensation Records

---

# Accrual Records

## Adjustments

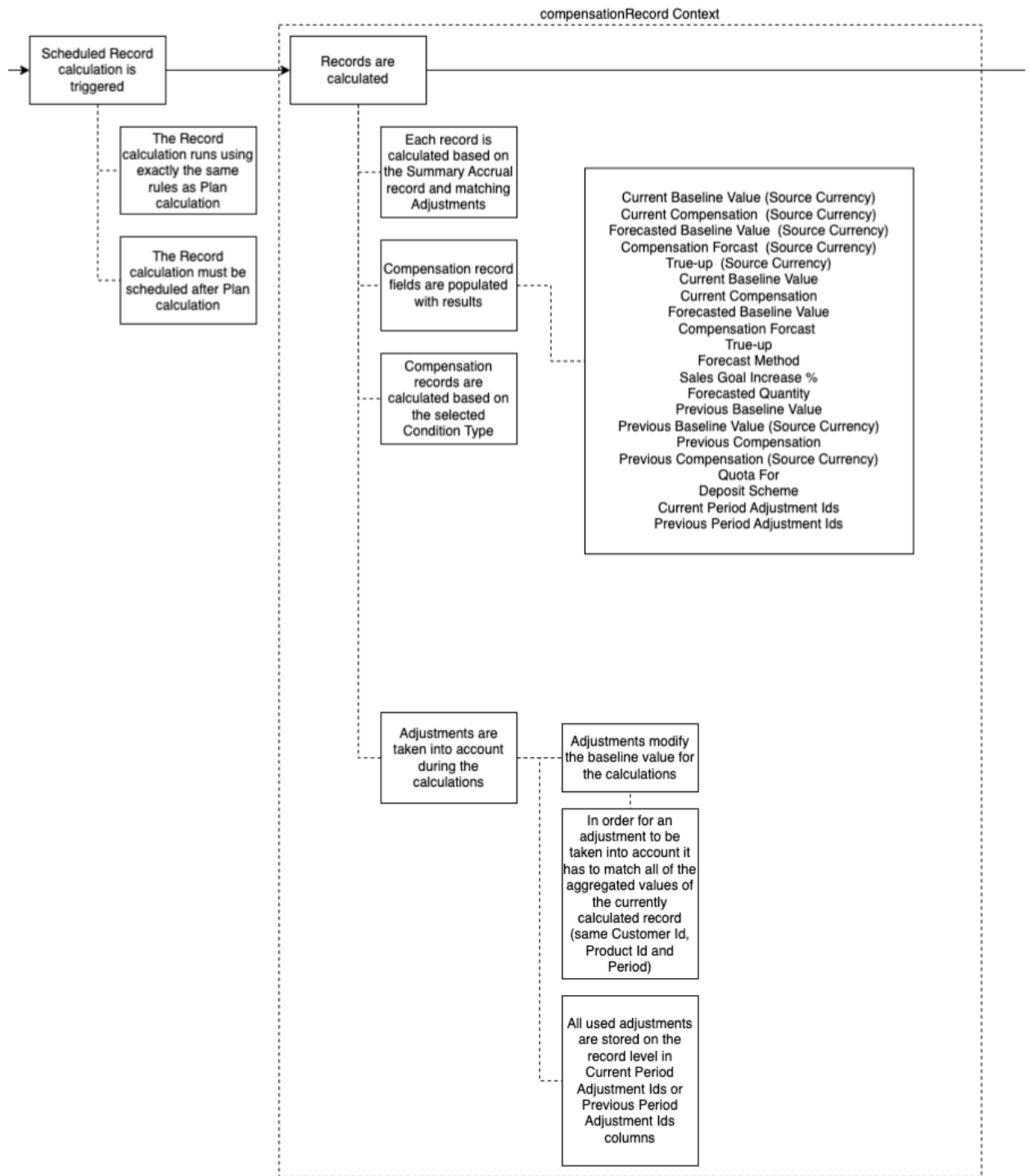
- > Channel Management
- > Optimization

- Filter Agreement Source Id == CO Id && Type == Transaction or Summary you want to check.

The screenshot shows the 'pricefx' interface for 'Sales Compensations / Accrual Records'. The table below is a filtered view of the data.

Compensation Record	Type	Agreement Source Id	Name	Condition Type	Status	Description	Start Date	End Date	Payout Date	Crez
<input type="checkbox"/>	Search...	Search...	CO-544	Search...	Search...	Search...	Search...	Search...	Search...	Search...
<input type="checkbox"/>	COR-8448	Transaction	CO-544	2022-Q3 Stepped Bo...	SC_SteppedBonus		01/07/2022	30/09/2022	03/10/2022	3
<input type="checkbox"/>	COR-8448	Transaction	CO-544	2022-Q3 Stepped Bo...	SC_SteppedBonus		01/07/2022	30/09/2022	03/10/2022	3
<input type="checkbox"/>	COR-8448	Transaction	CO-544	2022-Q3 Stepped Bo...	SC_SteppedBonus		01/07/2022	30/09/2022	03/10/2022	3
<input type="checkbox"/>	COR-8448	Transaction	CO-544	2022-Q3 Stepped Bo...	SC_SteppedBonus		01/07/2022	30/09/2022	03/10/2022	3
<input type="checkbox"/>	COR-8448	Transaction	CO-544	2022-Q3 Stepped Bo...	SC_SteppedBonus		01/07/2022	30/09/2022	03/10/2022	3
<input type="checkbox"/>	COR-8448	Transaction	CO-544	2022-Q3 Stepped Bo...	SC_SteppedBonus		01/07/2022	30/09/2022	03/10/2022	3
<input type="checkbox"/>	COR-8448	Transaction	CO-544	2022-Q3 Stepped Bo...	SC_SteppedBonus		01/07/2022	30/09/2022	03/10/2022	3

- **Sales Compensation Record** - This job calculates results including the adjustments (if any) in the Compensation Records.



## C. Compensation Records

1. Go to **Sales Compensations > Compensation Records**.



Dashboards

- > Pricefx Processes
- > Analytics
- > Price Setting
- > Quoting
- > Deal Planning
- > Agreements & Promotions
- > Rebates & Compensations

∨ **Sales Compensations**

Compensation Types

Condition Types

Compensation Plans

Calculations

**Compensation Records**

# Accrual Records

# Adjustments

- > Channel Management
- > Optimization

## 2. Select *Default* in **Set**.

pricefx Sales Compensations / Compensation Records

### Compensation Records

Set

Default

ID	Set	Source ID	Source ID w/o revision	Rank	Created
COR-6532	Default	CO-393	CO-393	0	13/09/2022 10:50
COR-6533	Default	CO-393	CO-393	0	13/09/2022 10:50

## 3. In the Compensation Records page, you can check the results in Payout Currency and Source Currency of a Compensation Plan that you have created, submitted and had calculated by the Calculations job.

pricefx Sales Compensations / Compensation Records

### Compensation Records

Set: Default

Compensation Type	Payee Name	Current Baseline Value	Current Compensation	Pay To ID	Current Baseline Quantity	Current Baseline Value (Source Currency)	Current Compensation (Source Currency)	Previous Baseline Value
Individual Compensation	Seller 003	197,415.06	329.03			1,562.18		2.60
Individual Compensation	Seller 003	2,013,852.55	3,685.45			15,935.96		29.16
Individual Compensation	Seller 003	1,901,251.04	6,854.20			15,044.92		54.24
Individual Compensation	Seller 003	0.00	6,854.20			0.00		54.24
Individual Compensation	Seller 003	0.00	6,854.20			0.00		54.24
Individual Compensation	Seller 003	2,696,809.77	11,348.88			21,340.31		89.81
Individual Compensation	Seller 003	561,749.39	12,285.13			4,445.22		97.21

For more details about the Calculation Flow see [Flow Diagrams \(SC\)](#).

## Adjustments

Adjustments allow you to modify the value of compensation that is going to be paid to a payee.

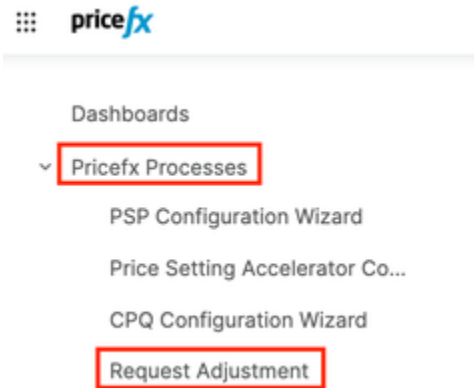
There are two places you can adjust:

- Request Adjustment
- Adjustment Form - only for the [SC] Administrator and [SC] Sales Manager roles

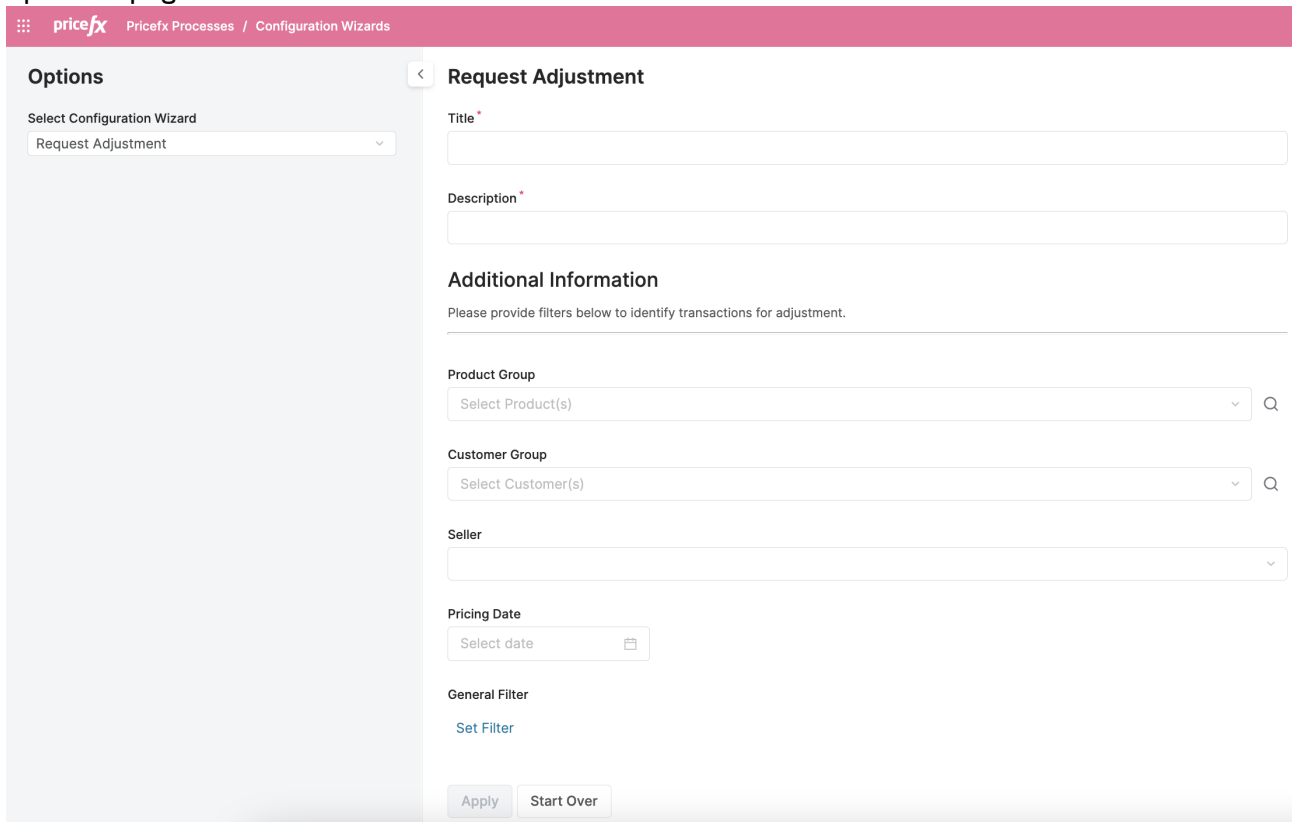
## Request Adjustment

This Request Adjustment can be viewed by any of these: Sales Agent, Sales Manager, Administrator.

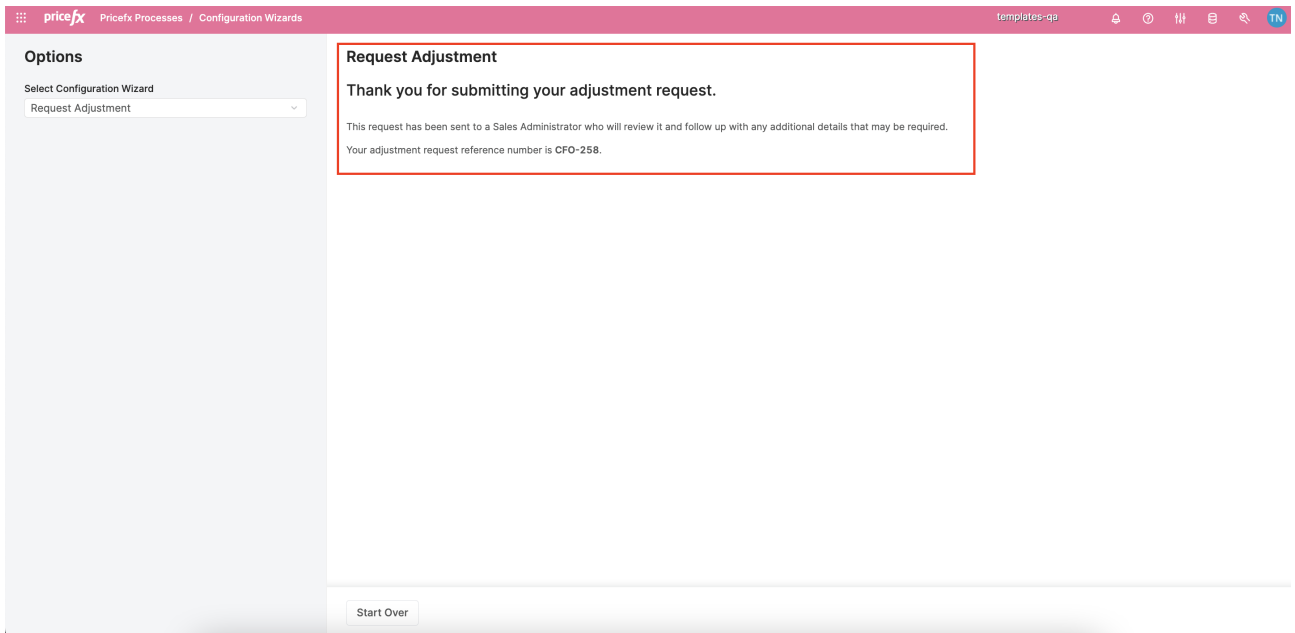
1. Go to **Pricefx Processes > Request Adjustment.**



2. Open the page.

A screenshot of the 'Request Adjustment' form in the Pricefx application. The form is displayed in a sidebar layout. On the left, under 'Options', there is a 'Select Configuration Wizard' dropdown menu with 'Request Adjustment' selected. The main form area is titled 'Request Adjustment' and contains several fields: 'Title' (required), 'Description' (required), 'Additional Information' (with a note: 'Please provide filters below to identify transactions for adjustment.'), 'Product Group' (dropdown), 'Customer Group' (dropdown), 'Seller' (dropdown), 'Pricing Date' (date picker), and 'General Filter' (with a 'Set Filter' link). At the bottom of the form are 'Apply' and 'Start Over' buttons.

3. You can request any transactions for the adjustment based on these filters. They provide affect the transactions that you want to adjust.
  - Product Group
  - Customer Group
  - Seller
  - Pricing Date
  - General Filter
4. After filling the required Title, Description, and filters, click the **Apply** button.
5. The submit message, including your adjustment request reference number, is displayed later.



## Adjustment Form

**i** Access to this Adjustment Form is restricted to users with the user group SC\_Administrator (label: [SC] Administrator), SC\_SalesManager (label: [SC] Sales Manager).

1. Go to **Sales Compensations > Adjustments**.



Dashboards

- > Pricefx Processes
- > Analytics
- > Price Setting
- > Quoting
- > Deal Planning
- > Agreements & Promotions
- > Rebates & Compensations
- ✓ **Sales Compensations**

Compensation Types

Condition Types

Compensation Plans

Calculations

Compensation Records

# Accrual Records

## Adjustments

- > Channel Management
- > Optimization

2. Create New Adjustments and fill in the Label and User Group (Edit) and User Group (View Details).



Attribute 1	Attribute 2	User Group (Edit)	User Group (View Details)	Created
Search...	Search...	Search... ▾	Search... ▾	Sea... → 📅

### Create New Adjustments ×

**Label** ?

**Attribute 1** ?

**Attribute 2** ?

**User Group (Edit)** ?  ▾

**User Group (View Details)** ?  ▾

18/10/2022 17:57

The Adjustments page has the following options:

pricefx Sales Compensations / Adjustments / Adjustment Transaction SC-001 templates-app

← Adjustment Transaction SC-001 ○ Draft - Draft Submit

Default

Query Details

- Payee Profile
- Transactions
- Messages
- Attachments
- Workflow

Recalculate

**Inputs**

Created By  
Tram Nguyen

Created On  
Wed Dec 07 07:06:08 UTC 2022

Title  
---

Status  
DRAFT

Description

**Find Transactions**

Use these filters to specify transactions which you want to adjust.

Product Group

Calculations

- **Inputs:**
  - Created By
  - Created On
  - Title - filled when you create from the Request Adjustment
  - Status
  - Description
- **Find Transactions** - Use these filters to specify transactions which you want to adjust:
  - Product Group
  - Customer Group
  - Seller
  - Pricing Date
  - General Filter
  - Filtered Transactions

Default

- Query Details
- Payee Profile
- Transactions
- Messages
- Attachments
- Workflow

### Find Transactions

Use these filters to specify transactions which you want to adjust.

**Product Group**

**Customer Group**

**Seller**

**Pricing Date**

**General Filter**

[Edit Filter](#)

**Warning:** For performance reasons only last 10 transactions found are shown below. Please make sure you want to adjust ALL transactions according to your filter or provide more specific filter. Keep in mind that adjustments will be applied to each transaction separately.

**Filtered Transactions**

Customer Id	Unique Id	Invoice Price	Gross Margin	Pricing Date
CD-00053	TXT_1579078529878...	300.47	128.64	2022-01-15
CD-00042	TXT_1579078529879...	48.31	19.93	2022-01-15
CD-00097	TXT_1579078529879...	204.93	63.28	2022-01-15
CD-00041	TXT_157907852988...	103.6	20.76	2022-01-15
CD-00007	TXT_157907852988...	200.35	54.94	2022-01-15

**Note:** For performance reasons only the last 10 transactions found are shown below. Make sure you want to adjust ALL transactions according to your filter or provide more specific filter. Keep in mind that adjustments will be applied to each transaction separately.

- **Define Adjustment** - Select Adjustment Type and specify what you want to change. Defined adjustment will be applied to *every selected transaction individually*.
  - If Adjustment Type == **Change Data** (this type allows you to change values, e.g., from Seller A to Seller B or from Date A to Date B)

- Query Details
- Payee Profile
- Transactions
- Messages
- Attachments
- Workflow

### Define Adjustment

Select adjustment type and specify what you want to change. Defined adjustment will be applied to **every selected transaction individually**.

Adjustment Type

Change Data

New Seller

New Product

Select Product

New Customer

Select Customer

New Date Value

Select date

Info

You can adjust any supported baseline field, but the adjustment will be applied to calculations only when given Condition Type uses selected field for compensation calculation.

New Invoice Price

New Quantity

New Gross Margin

- New Seller
- New Product
- New Customer
- New Date Value
- New Invoice Price
- New Quantity
- New Gross Margin
- If Adjustment Type == **Split Credit** (this type allows to split data among other sellers but you have to make sure the sum of split sellers equals to 100%)



## Define Adjustment

Select adjustment type and specify what you want to change. Defined adjustment will be applied to **every selected transaction individually**.

Adjustment Type

Split Credit

Seller Credits Split

<input type="checkbox"/> Seller Name	Percentage
<input type="checkbox"/> Seller 002 (SC-002)	45 
<input type="checkbox"/> Seller 003 (SC-003)	55 

2 rows

Add



If the sum of split sellers is not equal to 100%, there is a warning message displayed:

## Define Adjustment


Select adjustment type and specify what you want to change. Defined adjustment will be applied to **every selected transaction individually**.

Adjustment Type

Split Credit

**Warning: Credits split percentages sum isn't equal to 100%, adjust accordingly:**

Seller Credits Split

<input type="checkbox"/> Seller Name	Percentage
<input type="checkbox"/> Seller 002 (SC-002)	45 

1 row

Add



- You can go to other tabs such as Payee Profile, Transactions to check dashboard data.

pricefx Sales Compensations / Adjustments / Adjustment Transaction SC-001 templates-ga

← Adjustment Transaction SC-001 ○ Draft - Draft Submit

Default

Query Details

- Payee Profile
- Transactions
- Messages
- Attachments
- Workflow

**Dashboard Settings**

DATA FILTER

Seller(s)  
Select Sellers

Reporting Currency  
 Source Currency  
 Payout Currency

Payout Year\*  
2022

Payout Date\*  
02/12/2022

Apply Settings

**Admin Dashboard** + Add Portlet (13) Export to Excel

Payout Summary

Payouts 02/12/2022

Compensations YTD <b>€ 3,049.49</b>	Current Payments <b>€ 0.00</b>
--	-----------------------------------

Payout Table

Seller Id	YTD Sales	YTD Margin	YTD Margin %	YTD Compen...	Current Period Sales	Current Period Compensatio
No data						

- Lastly, you need to click on the Workflow tab and submit it. If not, there is no adjustment generated in the Accrual Records.
- You can check these generated adjustment in the Accrual Records as follows:
  1. Go to **Sales Compensations > Accrual Records**.



Dashboards

> Pricefx Processes

---

> Analytics

---

> Price Setting

> Quoting

> Deal Planning

> Agreements & Promotions

> Rebates & Compensations

∨ **Sales Compensations**

Compensation Types

Condition Types

Compensation Plans

Calculations

Compensation Records

---

# Accrual Records

## Adjustments

- > Channel Management
- > Optimization

2. Filter Type == Adjustment && Compensation Record == your CFO ID you want to check.

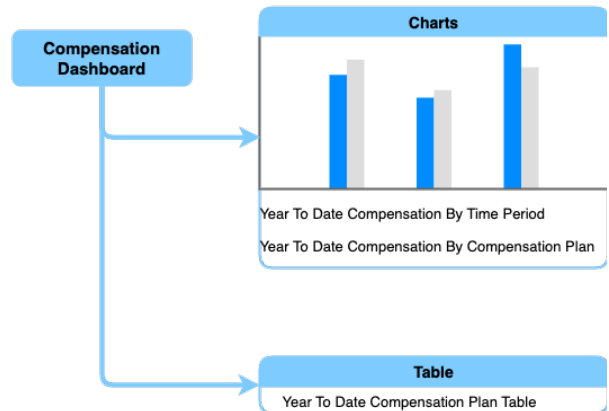
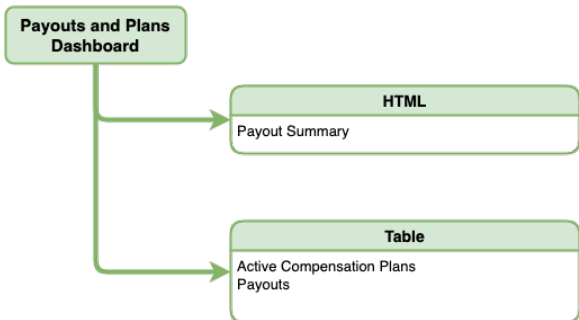
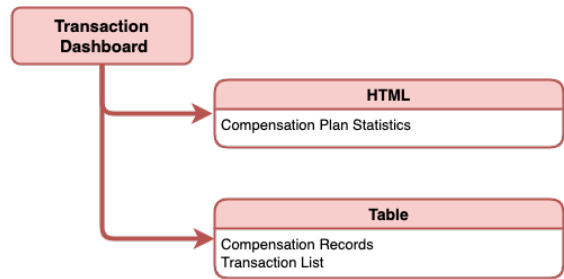
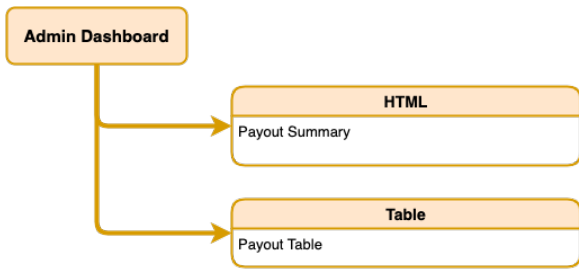
Compensation Record	Type	Name	Condition Type	Status	Description	Start Date	End Date	Payout Date	Created	Created By	Last Update Date	Last Up
221	Adjustment	Search...	Search...	Search...	Search...	S...>	S...>	S...>	Sea...>	Selec...>	Sea...>	Select
221	SourceAdjustment	221-Change Data-UrDFou2JScSMj3				01/09/2022	30/09/2022		21/10/2022 19:56	Tram Nguyen	21/10/2022 19:56	Tram Nc
221	TargetAdjustment	221-Change Data-0wHkOZjsh5IPkw				01/09/2022	30/09/2022		21/10/2022 19:56	Tram Nguyen	21/10/2022 19:56	Tram Nc

For more details about the Calculation Flow see [Flow Diagrams \(SC\)](#).

## Dashboards Description (SC)

Sales Compensations (SC) has four dashboards used for overview, analysis and tracking of compensations:

- [Admin Dashboard](#) - Overview dashboard for tracking all payouts for SC Administrator.
- [Payouts and Plans Dashboard](#) - Personal dashboard for sales agents, managers and administrators where they see planned and past payouts together with their plans.
- [Transaction Dashboard](#) - Analytical dashboard for sales agents, managers and administrators that can be used for drilling down to transaction level while respecting the individual permissions.
- [Compensation Dashboard](#) - Provides a progress overview for individual sales agents for the current year.



**i** The dashboard access is controlled by Sellers records from the Sellers master data. If you see the below message, it means your Pricefx user ID is not associated with any record:

Sales Agent: "There are no Compensation Records associated with your seller data (your user account: %s)"

Sales Manager: "There are no Compensation Records associated with your seller data or that of your subordinates (your user account: %s)"

## Dashboards Visibility

The dashboards are visible to the following user roles:

- **Admin Dashboard** - For administrators. No other limitations.
- **Payouts and Plans** - For administrators, managers and agents.
  - Admins can see everything.
  - Managers can see data related to them and their subordinates.
  - Agents can see the data related to them.
- **Transactions** - For administrators, managers and agents.
  1. Admins can see everything.
  2. Managers can see data related to them and their subordinates.
  3. Agents can see the data related to them.

## Admin Dashboard (SC)

This dashboard is designed for administrators who can see all payouts for sellers in a selected time period.

**i** Access to this dashboard is restricted to users with the user group SC\_Administrator (label: [SC] Administrator).

User Inputs

## Select Dashboard

Admin Dashboard



## DATA FILTER

### Seller(s)

Select Sellers



### Reporting Currency



Source Currency



Payout Currency

### Compensation Plan Currency \* ?

EUR



### Payout Year \*

2022



### Payout Date \*

02/12/2022



## Apply Settings

Input Filter Name	Values	Notes
Seller(s)	Seller Group picker	Optional. An empty value means the whole salesforce is selected.
Reporting Currency	Radio button: <ul style="list-style-type: none"> <li>Source Currency</li> <li>Payout Currency</li> </ul>	<ul style="list-style-type: none"> <li>If Source Currency is selected, the dashboard works with just pre-currency changes (no filtering, all results in DM currency).</li> <li>If Payout Currency is selected, an additional option input is displayed that allows to further down specify the currency. <ul style="list-style-type: none"> <li>The input is named: Compensation Plan Currency</li> <li>Mandatory input.</li> <li>Values: Shows only currencies selected as "Payout Currency" in plan(s) of the selected Seller(s)</li> </ul> </li> </ul>
Payout Year	Year drop-down picker	<p>Mandatory field. It shows only years where there were payouts for the selected Seller(s).</p> <p>By default, the current year should be selected.</p>
Payout Date	Date drop-down picker	<p>Mandatory field. It shows only dates of a selected Payout Year where there were payouts for the selected Seller(s).</p> <p>By default the payout date closest to the current date (in future) should be selected.</p>

### Portlets

- **Payout Summary** - Shows a total sum of all payouts from the filtered context.
  - **Compensations YTD** - Total sum of YTD Compensation for all payouts from the Payout table.

- **Current Payments** - Total sum of Current Period Compensation for all payouts from the Payout table.
- **Payout Table** - Provides a high level overview with a split between individual and group compensations. It has information on when the respective line was or will be paid. Numbers with a yellow background are incomplete summaries for the next month. Until their end date is finalized, their value can change.
  - **Seller Id** - ID of the Seller
  - **YTD Sales** - Sum of revenue from transaction data YTD
  - **YTD Margin** - Sum of margins from transaction data YTD
  - **YTD Margin %** - Calculated as YTD Margin / YTD Sales
  - **YTD Compensation** - Sum of compensations valid YTD
  - **Current Period Sales** - Sum of revenue from transaction data in a current date
  - **Current Period Compensation** - Sum of Individual Compensations and Group Compensations (fields below)
  - **Individual Compensation** - Sum of individual compensations valid in a selected payout date and source/payout currency (division by header type)
  - **Group Compensation** - Sum of group compensations valid in a selected payout date and source /payout currency (division by header type)
  - **Adjustment** - Link cell to Adjustments page

### Payouts and Plans Dashboard (SC)

This dashboard is a dedicated overview of Sales Agents' Compensation Plans and Payouts. The dashboard provides a list of Compensation Plans and their structure without having to access the Sales Compensations module. Additionally, users can also see their payouts (made or planned).

The dashboard is intended to be used by Sales Agents, Sales Managers as well as Sales Compensations administrators. Sales Agents can see only their own data, Sales Managers can see details for all their reports (based on the "Reports To" field in the Seller Master table) and SC Admins have unlimited selection.

User Inputs

## Select Dashboard

Payouts and Plans Dashboard



## DATA FILTER

### Seller(s)

Select Sellers



### Reporting Currency

Source Currency

Payout Currency

### Compensation Plan Currency \* ?

EUR



### Payout Year \*

2022



### Payout Date \*

02/12/2022



## Apply Settings

Input Filter Name	Values	Notes
Seller(s)	Seller Group picker	<ul style="list-style-type: none"> <li>• Optional</li> <li>• Pre-selected and read only for Sales Agents.</li> <li>• Only reporting agents can be selected by Sales Managers.</li> <li>• Empty value means the whole salesforce is selected (based on permissions).</li> </ul>
Reporting Currency	Radio button: <ul style="list-style-type: none"> <li>• Source Currency</li> <li>• Payout Currency</li> </ul>	<ul style="list-style-type: none"> <li>• If Source Currency is selected, the dashboard works with just pre-currency changes (no filtering, all results in DM currency)</li> <li>• If Payout Currency is selected, an additional option input is displayed that allows to further down specify the currency.               <ul style="list-style-type: none"> <li>• The input is named: Compensation Plan Currency</li> <li>• Mandatory input.</li> <li>• Values: Shows only currencies selected as "Payout Currency" in plan (s) of the selected Seller(s)</li> </ul> </li> </ul>
Payout Year	Year drop-down picker	Mandatory field. It shows only years where there were payouts for the selected Seller(s). By default the current year should be selected.
Payout Date	Date drop-down picker	Mandatory field. It shows only dates of a selected Payout Year where there were payouts for the selected Seller(s). By default the payout date closest to the current date (in future) should be selected.

Portlets

- **Payout Summary** - Quick summary of all payouts that have been paid in the selected period. Aside from the payout sum, there is also revenue, so the users can see this KPI on the side of their commission.
  - Summary for Payout Date, No. of Sellers, No. of Compensation Plans and No. of Payouts
  - Revenue - Sum of revenues from transaction data based on the selected payout date and currency input
- **Active Compensation Plans** - Table showing a list of all open plans. Clicking a Compensation Plan opens a detail. In the detail view, the user can see a simplified version of the configuration from the Compensation Plan stored in the Sales Compensations module. The view does not provide current progress but shows the definition on one screen when a line is clicked.
  - **Label** - Label of Compensation Plan
  - **Name** - ID of Compensation Plan
  - **Payout Date**
  - **Type** - Compensation Type
  - **Seller Id**
  - **Seller Name**
- **Payouts** - This table gives an overview of what has been paid and what is estimated for the next scheduled payment. Each Compensation Plan can have several payouts because they are linked to the conditions (items) of the plan. The user is able to see how much and when payouts occurred in past. Numbers with a yellow background are incomplete for the upcoming months. Until their due date is reached, their value can change.
  - **Label** - Label of Compensation Plan
  - **Source Id** - ID of Compensation Plan
  - **Header Type** - Compensation Type
  - **Payout Date** - Gets the Payout Date in Compensation Records based on the selected Payout Date input
  - **Amount** - Gets the Current Compensation in Compensation Records based on the source/payout currency input
  - **Currency** - Selected source/payout currency input
  - **Seller Id**
  - **Seller Name**

## Transaction Dashboard (SC)

This dashboard provides an overview of all transactions included in the calculation of the selected Compensation Plan with the possibility to further filter on specific compensation record or transaction date.

The dashboard is for Sales Agents, Sales Managers as well as Sales Compensations administrators. Sales Agents can see only their own data, Sales Managers can see details for all their reports (based on the "Reports To" field in the Seller Master table) and SC Admins have unlimited selection.

User Inputs

### Select Dashboard

Transaction Dashboard



### DATA FILTER

Please specify filters for transaction data.  
If no or too broad filters are selected the dashboard may take significant amount of time to load.

#### Seller(s)

Select Sellers



#### Compensation Plan Currency ?



#### Compensation Plan Type(s)



#### Compensation Plan Status(es)



#### Compensation Plan(s)



#### Compensation Record(s)

▼

**Date From**

Select date
📅

**Date To**

Select date
📅

**Apply Settings**

Input Filter Name	Values	Notes
Seller(s)	Seller Group picker	<ul style="list-style-type: none"> <li>• Optional</li> <li>• Pre-selected and read only for Sales Agents.</li> <li>• Only reporting agents can be selected by Sales Managers.</li> <li>• Empty value means the whole salesforce is selected (based on permissions).</li> </ul>
Compensation Plan Currency	The currency list in which plans were created	<ul style="list-style-type: none"> <li>• If a currency is selected, it limits the possible selection in the other inputs.</li> <li>• If nothing is selected in the input, the dashboard uses DM Source Currency.</li> </ul>
Compensation Plan Type(s)	Compensation Types	<ul style="list-style-type: none"> <li>• Optional</li> <li>• Multi select</li> <li>• Empty value means all Compensation Plan Types.</li> </ul>
Compensation Plan Status(s) (*see note below)	Compensation Plan Status(es)	<ul style="list-style-type: none"> <li>• Optional</li> <li>• Multi select</li> <li>• Empty value means all compensation plan statuses.</li> </ul>
Compensation Plan(s)	List of all Compensation Plans for a sales person of the selected compensation type	<ul style="list-style-type: none"> <li>• Optional</li> <li>• Multi select</li> <li>• Empty value means all Compensation Plans.</li> </ul>

Compensation Record(s)	List of all Compensation Records associated with a given Compensation Plan	<ul style="list-style-type: none"> <li>• Optional.</li> <li>• Multi select.</li> <li>• Empty value means all compensation records.</li> </ul>
Date From	Date	<ul style="list-style-type: none"> <li>• Empty value means no date filtering.</li> </ul>
Date To	Date	<ul style="list-style-type: none"> <li>• Empty value means no date filtering.</li> </ul>

**Note on Compensation Plan Status:**

**Sales Agent:**

- Can see ONLY statuses "Approved" and "Superseded" if those are applicable for "their" plans.
- The input is pre-populated to "Approved" by default.
- Cannot delete all possible values (Approved and Superseded if any).

**Sales Managers** (as they have Manage Plans User role) **and Administrators:**

- NOT limited to any selection and they can leave the field blank which acts as "no filter".
- The input is pre-populated to "Approved" by default.

**Portlets**

- **Compensation Plan Statistics** - Statistical overview of the selected scope from Compensation Plans in a form of several tiles. The following information is represented on tiles:
  - Number of Transactions
  - Number of Negative Baseline Transactions
  - Number of Sales Agents
  - Number of unique SKUs (Products IDs)
  - Number of unique Customers
  - Total Baseline
- **Compensation Records** - Provides information about attributes for compensation records. It can contain just one line if the analysis is focused on one specific condition from the Compensation Plan. The "Id" column is an active link leading directly to the full detail of the Compensation Record, similarly for "Compensation Plan Id" which leads to the full detail of the Compensation Plan. The SC Administrator can follow the link and see a context that was not available on the dashboard. An example is "Calculation Baseline" showing raw transaction data for the selected compensation record.
  - Id - ID of compensation record
  - Name - Name of compensation record
  - Label - Label of compensation record
  - Seller Id
  - Seller Name
  - Compensation Plan Id
  - Compensation Plan
  - Compensation Plan Type
- **Transaction List** - A list of selected columns from source transactions that were used as calculation baseline for the selected scope.
  - Date
  - Seller Id
  - Seller Name
  - Customer Id
  - Product Id

- Baseline

**i** Tip: Use Group By option on a column you would like to aggregate.

### Compensation Dashboard (SC)

This dashboard provides a progress overview for a particular Sales Agent for the current year. A central focus is to show income in a form of a table and bar charts for Compensation Plans or time periods (months, quarters, ...).

Its primary use is for Sales Agents but it will give valuable insights to Sales Managers or SC Administrators as a supportive analytical tool.

User Inputs

### Select Dashboard

Compensation Dashboard ▼

### DATA FILTER

Seller \*

Seller 002 (SC-002) ▼

Compensation Plan \*

▼

This is required

Time Period \*

Monthly ▼

**Apply Settings**

Input Filter Name	Values	Notes

Seller	One Sales Agent	Mandatory field. 'Compensation Plan' input is affected by the selected value.
Compensation Plan	Empty or one specific Compensation Plan	Mandatory field. Empty value compares all agreements that are assigned to the agent from the Seller input.
Time Period	Monthly, Quarterly, Semi-Annually, Annually	Sets an aggregation period for the 'Year To Date Compensation' portlet.

#### Portlets

- **Year To Date Compensation** - A bar chart aggregating all compensations for the selected Sales Agent in time periods. If no specific Compensation Plan was selected, one bar combines all Compensation Plans together giving a total compensation sum that was paid out to the Sales Agent. Be careful, 'Time Period' input provides an aggregation period, e. g. Quarterly would show increments only in 2 bars if the compensation had a semi-annual payment period.
- **Year To Date Compensation By Compensation Plan** - A bar chart showing the total compensations in one agreement as a bar on the X-axis.
- **Year To Date Compensation Plan Table** - A summary table showing the breakup of Compensation Plans through Line Items (conditions) down to Compensation Record in a single table. A user can see which lines or periods are outliers.

## Reference (SC)

- [Glossary \(SC\)](#)
- [Type Overview \(SC\)](#)
- [Components \(SC\)](#)
- [Workflow Variables \(SC\)](#)
- [Upgrade \(SC\)](#)
- [Troubleshooting \(SC\)](#)
- [Known Limitations \(SC\)](#)

## Glossary (SC)

This page lists specific terms, vocabulary, and definitions that are used in the context of the Sales Compensation package.

Term	Description
Compensation	A term used in the system as a final reward. In reality, it could be for some companies just a bonus, for other commission and in some cases incentives.
Incentive	An amount of money or non-monetary reward to motivate someone to achieve something.
Commission	Always in cash form, commission is an income payment. It could be a percentage of a product or service sold.

Sales Compensation Plan	It outlines sales agent' base salary as well as the company's commission and incentive program with strategy. Commission, bonus and incentive structure incentivize sales force to reach their objectives in order to earn a deserved reward.
Pay mix	The ratio of base salary to target compensations that make up On-Target Earnings (OTE).
On-Target Earnings (OTE)	Refers to an employee's pay mix made of basic salary and the additional variable component such as commission as their compensation.
SPIF	
Catch Up	True Up
Tollgate	Gate, Treshold
Clawback	
Attainment	
Cup	
Draw	
Quota	

 See also the general [glossary](#).

## Type Overview (SC)

This section provides an overview and calculation examples for types that are distributed in the package.

- [Compensation Types](#)
- [Condition Types](#)
- [Calculation Examples](#)
  - [Single Quota Amount](#)
  - [Single Quota Percent](#)
  - [Multi Quota Amount](#)
  - [Multi Quota Percent](#)
  - [Growth Absolute Amount](#)
  - [Growth Absolute Percent](#)
  - [Growth Percent Percent](#)
  - [Growth Percent Amount](#)
  - [Stepped Amount](#)
  - [Stepped Percent](#)
  - [Flat Bonus](#)
  - [Multi Target Bonus](#)
  - [Stepped Bonus](#)
  - [Variable Pay Linear Amount](#)
  - [Variable Pay Multi Target Amount](#)

- [Variable Pay Stepped Amount](#)
- [Agreement Records Calculation](#)
- [Payout Value Calculation](#)

## Compensation Types

List of available types with details.

Name	Header Logic	Input	Payout	Description
Individual Compensation	Sales Compensation Header	Seller (single person)	Seller (single person)	Compensation is tracked for one Sales Agent who receives all payout money.
Group Compensation	Sales Compensation Header	Team (multiple sellers)	Team (multiple sellers)	Compensation is tracked for all Sales Agents who are a part of a selected Team.

## Condition Types

List of available types with their default setup. Below you will find how to modify the setup.

Type	Name	Quota Type	Inputs (with units)
Conditional	Single Quota Amount	Single	Quota input: \$ Compensation input: \$
	Single Quota Percent	Single	Quota input: \$ Compensation input: %
Growth	Growth Absolute Amount	Multi	Quota input: \$ Compensation input: \$
	Growth Absolute Percent	Multi	Quota input: \$ Compensation input: %
	Growth Percent Amount	Multi	Quota input: % Compensation input: \$
	Growth Percent Percent	Multi	Quota input: % Compensation input: %
Stepped	Stepped Amount	Multi	Quota input: \$ Compensation input: \$
	Stepped Percent	Multi	Quota input: \$ Compensation input: %
Bonus	Flat Bonus	Single	Quota input: \$ Target Incentive input: \$

	Multi Target Bonus	Multi	Quota input: \$ Target Incentive input: \$ Multi Target Bonus: <ul style="list-style-type: none"> <li>• Target Achievement: %</li> <li>• Bracket Bonus Rate: %</li> </ul>
	Stepped Bonus	Multi	Quota input: \$ Target Incentive input: \$ Stepped Bonus input: <ul style="list-style-type: none"> <li>• Target Achievement input: %</li> <li>• Bracket Bonus Rate input: %</li> </ul>
Variable Pay	Variable Pay Linear Amount	Single	Quota input: \$ Variable Pay input: %
	Variable Pay Multi Target Amount	Multi	Quota input: \$ Variable Pay input: % Stepped Bonus input: <ul style="list-style-type: none"> <li>• Target Achievement input: %</li> <li>• Bracket Bonus Rate input: %</li> </ul>
	Variable Pay Stepped Amount	Multi	Quota input: \$ Variable Pay input: % Stepped Bonus input: <ul style="list-style-type: none"> <li>• Target Achievement input: %</li> <li>• Bracket Bonus Rate input: %</li> </ul>

**i** The dollar \$ sign represents a currency in general, not an actual USD.

## Calculation Examples

### Single Quota Amount

Inputs in the plan:

- Quota - \$
- Compensation - \$

Success formula: COMPENSATION\_INPUT

Example

Quota [input]	Compensation [input]	Baseline	Compensation [output]
\$100,000	\$1,000	\$110,000	\$1,000
\$100,000	\$1,000	\$90,000	\$0

### Single Quota Percent

Inputs in the plan:

- Quota - \$
- Compensation Value - %

Success formula:  $( \text{COMPENSATION\_INPUT} / 100 ) \times \text{BASELINE}$

Example

Quota [input]	Compensation Value [input]	Baseline	Compensation [output]
\$100,000	1%	\$110,000	\$1,100
\$100,000	1%	\$90,000	\$0

### Multi Quota Amount

Inputs in the plan:

- Quota - \$
- Compensation - \$

Success formula:  $\text{COMPENSATION\_INPUT}$

Example

Quota [tiered input]	Compensation [tiered input]
\$1000	\$100
\$1500	\$150

Result:

Baseline (\$)	Compensation [output]
\$100	\$0
\$1100	\$100
\$1600	\$150

### Multi Quota Percent

Inputs in the plan:

- Quota - \$

- Compensation - %

Success formula:  $( \text{COMPENSATION\_INPUT} / 100 ) \times \text{BASELINE}$

Example

Quota [tiered input]	Compensation [tiered input]
\$1000	1%
\$1500	10%

Result:

Baseline (\$)	Compensation [output]
\$100	\$0
\$1100	\$11
\$1600	\$160

### Growth Absolute Amount

Comparison to some previous period - Month, Quarter, Semi-Year, Year.

Inputs in the plan:

- Quota(s) - \$
- Compensation Value(s) - \$

Success formula:  $\text{COMPENSATION\_INPUT\_FOR\_TIER}$

Example

Definition:

Quota [input] - Growth Tier	Compensation Value [input]
\$10,000	\$100
\$25,000	\$300
\$100,000	\$10,000

Result:

Growth Baseline (Current - Previous)	Compensation [output]
\$5,000	\$0
\$30,000	\$300
\$150,000	\$10,000

### Growth Absolute Percent

Comparison to some previous period - Month, Quarter, Semi-Year, Year.

Inputs in the plan:

- Quota(s) - \$
- Compensation Value(s) - %

Success formula:  $(\text{COMPENSATION\_INPUT\_FOR\_TIER} / 100) \times \text{BASELINE}$

Example

Definition:

Quota [input] - Growth Tier	Compensation [input]
\$10,000	1%
\$25,000	2%
\$100,000	5%

Result:

Baseline	Compensation [output]
\$5,000	\$0
\$25,000	\$500
\$150,000	\$7,500

### Growth Percent Percent

Comparison to some previous period - Month, Quarter, Semi-Year, Year.

Inputs in the plan:

- Quota(s) - %
- Compensation Value(s) - %

Success formula:  $(\text{COMPENSATION\_INPUT\_FOR\_TIER} / 100) \times \text{BASELINE}$

Example

Definition:

Quota [input] - Growth Tier %	Compensation Value [input]
2%	1%
5%	3%
10%	5%

Result:

Baseline	Calculation Baseline	Compensation [output]
1%	\$10,000	\$0
2%	\$100,000	\$1,000
11%	\$100,000	\$5,000

### Growth Percent Amount

Comparison with some previous period - Month, Quarter, Semi-Year, Year.

Inputs in the plan:

- Quota(s) - %
- Compensation Value(s) - \$

Success formula: COMPENSATION\_INPUT\_FOR\_TIER

Example

Definition:

Quota [input] - Growth Tier %	Compensation Value [input]
2%	\$1,000
5%	\$10,000
10%	\$25,000

Result:

Baseline	Compensation [output]
1%	\$0
2%	\$1,000
11%	\$25,000

### Stepped Amount

Inputs in the plan:

- Quota(s) - \$
- Compensation Value(s) - \$

Success formula for step 1: COMPENSATION\_INPUT\_FOR\_STEP1

Success formula for step 2: COMPENSATION\_INPUT\_FOR\_STEP1 + COMPENSATION\_INPUT\_FOR\_STEP2

Example

Definition:

Quota [input] - Step	Compensation Value [input]
\$10,000	\$100
\$50,000	\$500
\$100,000	\$5,000

Result:

Baseline	Compensation [output]
\$5,000	\$0
\$15,000	\$100
\$110,000	$\$100 + \$500 + \$5,000 = \$5,600$

### Stepped Percent

Inputs in the plan:

- Quota(s) - \$
- Compensation Value(s) - %

Success formula for step 1:  $(\text{COMPENSATION\_INPUT\_FOR\_STEP1} / 100) \times (\text{BASELINE} - \text{TARGET\_STEP1})$

Success formula for step 2:  $((\text{COMPENSATION\_INPUT\_FOR\_STEP1} / 100) \times \text{TARGET\_STEP1}) + ((\text{COMPENSATION\_INPUT\_FOR\_STEP2} / 100) \times (\text{BASELINE} - \text{TARGET\_STEP2}))$

Example

Definition:

Quota [input] - Step	Compensation Value [input]
\$10,000	1%
\$50,000	3%
\$100,000	10%

Result:

Baseline	Compensation [output]
\$5,000	\$0
\$15,000	$(\$5,000 * 1\%) = \$50$
\$110,000	$(\$40,000 * 1\%) + (\$50,000 * 3\%) + (\$10,000 * 10\%) = \$400 + \$1,500 + \$1,000 = \$2,900$

### Flat Bonus

Inputs in the plan:

- Quota - \$
- Target Incentive - \$

Success formula:  $(\text{BASELINE} / \text{QUOTA}) \times \text{TARGET\_INCENTIVE}$

Example

Quota [input]	Target Incentive [input]	Baseline	Compensation [output]
\$100,000	\$100	\$110,000	\$110
\$100,000	\$1,000	\$90,000	\$900

### Multi Target Bonus

Inputs in the plan:

- Quota - \$
- Target Incentive - \$
- Target Achievement - %
- Bracket Bonus Rate - %

Success formula for baseline under tier 1:  $(\text{BASELINE} / \text{QUOTA}) \times \text{TARGET\_INCENTIVE} \times \text{BRACKET\_BONUS\_RATE\_FOR\_TIER\_1} / 100$

Success formula for baseline under tier 2:  $((\text{BRACKET\_BONUS\_RATE\_FOR\_TIER\_1} / 100 \times \text{QUOTA}) / \text{QUOTA}) \times \text{TARGET\_INCENTIVE} \times \text{BRACKET\_BONUS\_RATE\_FOR\_TIER\_1} / 100 + ((\text{BASELINE} - (\text{BRACKET\_BONUS\_RATE\_FOR\_TIER\_1} / 100 \times \text{QUOTA})) / \text{QUOTA}) \times \text{TARGET\_INCENTIVE} \times \text{BRACKET\_BONUS\_RATE\_FOR\_TIER\_2} / 100$

Example

Quota [input] - \$1,000

Target Incentive [input] - \$100

Tiers definition:

Target Achievement [tiered input]	Bracket Bonus Rate [tiered input]
120%	3%
140%	5%
250%	10%

Result:

Baseline (\$)	Target Achievement (%)	Bracket Bonus Rate (%)	Compensation [output]
\$1,000	120%	3%	\$3

\$1,300	140%	5%	\$4.1
\$2,000	250%	10%	\$10.6
\$3,000	250%	10%	\$20.6

### Stepped Bonus

Inputs in the plan:

- Quota - \$
- Target Incentive - \$
- Target Achievement - %
- Bracket Bonus Rate - %

Success formula for baseline under tier 1:  $(\text{BASELINE} / \text{QUOTA}) \times \text{TARGET\_INCENTIVE} \times \text{BRACKET\_BONUS\_RATE\_FOR\_TIER\_1} / 100$

Success formula for baseline under tier 2:  $(\text{BASELINE} / \text{QUOTA}) \times \text{TARGET\_INCENTIVE} \times \text{BRACKET\_BONUS\_RATE\_FOR\_TIER\_2} / 100$

Example

Quota [input] - \$1,000


Target Incentive [input] - \$100

Tiers definition:

Target Achievement [tiered input]	Bracket Bonus Rate [tiered input]
120%	3%
140%	5%
250%	10%

Result:

Baseline (\$)	Target Achievement (%)	Bracket Bonus Rate (%)	Compensation [output]
\$1,000	120%	3%	\$3
\$1,300	140%	5%	\$5
\$2,000	250%	10%	\$10
\$4,000	above 250%	0%	\$0

 Please note that the Compensation drops to zero when the tier is not defined (in the example it is the case of "above 250%" Target Achievement).

### Variable Pay Linear Amount

Inputs in the plan:

- Quota - \$
- Variable Pay - %

Success formula:  $(\text{BASELINE} / \text{QUOTA}) \times \text{VARIABLE\_PAY} / 100 \times \text{SELLER\_SALARY}$

Example

Quota [input]	Variable Pay [input]	Baseline	Seller Salary (Annual)	Compensation [output]
\$1,000,000	10%	\$300,000	\$60,000	\$1,800
\$100,000	2%	\$90,000	\$80,000	\$1,440

**i** In case of Group compensations the salary taken to calculations is the average of all included seller salaries.

### Variable Pay Multi Target Amount

Inputs in the plan:

- Quota - \$
- Variable Pay - %
- Target Achievement - %
- Bracket Bonus Rate - %

Success formula for baseline under tier 1:  $(\text{BASELINE} / \text{QUOTA}) \times \text{VARIABLE\_PAY} / 100 \times \text{SELLER\_SALARY} \times \text{BRACKET\_BONUS\_RATE\_FOR\_TIER\_1} / 100$

Success formula for baseline under tier 2:  $((\text{BRACKET\_BONUS\_RATE\_FOR\_TIER\_1} / 100 \times \text{QUOTA}) / \text{QUOTA}) \times \text{VARIABLE\_PAY} / 100 \times \text{SELLER\_SALARY} \times \text{BRACKET\_BONUS\_RATE\_FOR\_TIER\_1} / 100 + ((\text{BASELINE} - (\text{BRACKET\_BONUS\_RATE\_FOR\_TIER\_1} / 100 \times \text{QUOTA})) / \text{QUOTA}) \times \text{VARIABLE\_PAY} / 100 \times \text{SELLER\_SALARY} \times \text{BRACKET\_BONUS\_RATE\_FOR\_TIER\_2} / 100$

Example

Quota [input] - \$1,000

Variable Pay [input] - 10%

*Tiers definition:*

Target Achievement [tiered input]	Bracket Bonus Rate [tiered input]
120%	3%
140%	5%
250%	10%

*Result:*

Baseline (\$)	Target Achievement (%)	Bracket Bonus Rate (%)	Seller Salary (\$)	Compensation [output]
\$1,000	120%	3%	\$10,000	\$3,000

\$1,300	140%	5%	\$10,000	\$4,100
\$2,000	250%	10%	\$10,000	\$10,600
\$3,000	250%	10%	\$10,000	\$20,600

**i** In case of Group compensations the salary taken to calculations is the average of all included seller salaries.

### Variable Pay Stepped Amount

Inputs in the plan:

- Quota - \$
- Variable Pay - %
- Target Achievement - %
- Bracket Bonus Rate - %

Success formula for baseline under tier 1:  $(\text{BASELINE} / \text{QUOTA}) \times \text{VARIABLE\_PAY} / 100 \times \text{SELLER\_SALARY} \times \text{BRACKET\_BONUS\_RATE\_FOR\_TIER\_1} / 100$

Success formula for baseline under tier 2:  $(\text{BASELINE} / \text{QUOTA}) \times \text{VARIABLE\_PAY} / 100 \times \text{SELLER\_SALARY} \times \text{BRACKET\_BONUS\_RATE\_FOR\_TIER\_2} / 100$

Example

Quota [input] - \$1,000

Target Incentive [input] - \$100

*Tiers definition:*

Target Achievement [tiered input]	Bracket Bonus Rate [tiered input]
120%	3%
140%	5%
250%	10%

*Result:*

Baseline (\$)	Target Achievement (%)	Bracket Bonus Rate (%)	Seller Salary (\$)	Compensation [output]
\$1,000	120%	3%	\$10,000	\$3,000
\$1,300	140%	5%	\$10,000	\$5,000
\$2,000	250%	10%	\$10,000	\$10,000
\$4,000	above 250%	0%	\$10,000	\$0

**i** In case of Group compensations the salary taken to calculations is the average of all included seller salaries.

⚠ Please note that the Compensation drops to zero when the tier is not defined (in the example it is the case of "above 250%" Target Achievement).

## Agreement Records Calculation

### Payout Value Calculation

The "Annual" target needs a Deposit Scheme. The scheme decides if the calculation is split by the number of periods in the target time span.

Input information:

- Condition Type: [SC] Single Quota Amount
- Payment Period: Quarterly
- Target For: **Annual**

Target input:

- Target Input: 100,000.0
- Compensation Input: 1,000.0

As the payment period is each quarter, we have 4 agreement records with the following data:

Agreement (Rebate) Record	Period	Period Sale	Cumulative Sale
RR01	Q1	90,000	90,000
RR02	Q2	11,000	101,000
RR03	Q3	49,000	150,000
RR04	Q4	50,000	200,000

### Non-Cumulative Example

The calculation is driven by following an execution pattern.

```

If ( Cumulative_Sale > Target_Input ) then
    Compensation_Value = Compensation_Input / 4
end

```

📌 Note: Number 4 stands for a number of agreement records (Rebate Records without internalization applied).

Agreement (Rebate) Record	Period	Compensation Value
RR01	Q1	0
RR02	Q2	250
RR03	Q3	250

### Cumulative Example

The calculation is driven by following an execution pattern.

```

If ( Cumulative Sale > Target_Input ) then
    Compensation_Value = ( Compensation_Input / 4 ) *
Current_Period_Index - Accrual
end

```

#### Notes:

- Number 4 stands for a number of agreement records (Rebate Records without application text changes).
- Current Period Index starts with 1.
- Accrual is the cumulative compensation.

Agreement Record	Period	Current Period Index	Accrual	Compensation Value
RR01	Q1	1	0	0
RR02	Q2	2	0	500
RR03	Q3	3	500	250
RR04	Q4	4	750	250

### Components (SC)

On this page, you will find a technical description and a list of components used in the package.

- [Groovy Overview](#)
  - [Logics](#)
  - [Libraries](#)
- [Advanced Configuration Options](#)
- [Company Parameters](#)
  - [RM\\_SC\\_ConditionTypes](#)
  - [SC\\_CustomerFieldMapping](#)
  - [SC\\_ProductFieldMapping](#)
  - [Dependency Parameters](#)
- [Attributes](#)
  - [Agreement Header Type Attributes](#)
  - [Condition Type Attributes](#)
  - [Agreement Record Attributes](#)
  - [Sellers Attributes](#)
- [Publishing Templates](#)
- [Dashboards](#)

- [Access Management](#)
- [Other Components](#)
  - [Approval Workflows](#)

## Groovy Overview

### Logics

Logic Name	Default Label	Description
SC_Compensation	Sales Compensation	This logic processes all compensation types. If you need to add more input/output elements for the compensation, add them in this logic.
SC_CompensationHeader	Sales Compensation Header	Header logic for a personal compensations.
SC_CompensationRecordCalculationFeeder	Compensation Record Calculation Feeder	Finds and calculates agreement records.
SC_Dashboard_Compensations		Input generation, portlet generation, data fetching, etc.
SC_Dashboard_Compensations_Configurator	Compensation Dashboard Configurator	Builds a Seller input for dashboards.
SC_OverviewReport	Overview Report	Logic supporting report of the same name.
SC_SalesCompensationAgreementReport	Sales Compensation Agreement Report	Logic supporting report of the same name.
SC_TransactionsReport	Transactions Report	Logic supporting report of the same name.
SC_YTDReport	Year To Date Report	Logic supporting report of the same name.

### Libraries


Library Name	Default Label	Description
SC_CompensationProcessingLib	Sales Compensation Processing Library	Provides common functions used in the compensation package such as utilities for dates manipulation, inputs, record fetching, etc.
SC_CompensationTypesLib	Condition Type Library	Defines all compensation types. When you have a new condition type, add a new element to this library to define functions for this condition type.

SC_Compensation DashboardsLib	Sales Compensation Dashboard Library	Provides common functions used in dashboards of the compensation package.
SharedLib		Provides common functions used in the compensation package. Provided by Shared Groovy Library dependency.
ApprovalWorkflow		Provides function for approval workflow logics. Provided by Approval Workflow Library dependency.
FormulaEvaluator		Used in Approval Workflow. Provided by Formula Evaluator Library dependency.
HighchartLibrary		Provides "Highcharts" functions used in dashboards. Provided by Dashboards Accelerator Library dependency.

### Advanced Configuration Options

Package configuration is done from this central point in the system. Option values are configured mostly during a package deployment process and can be adjusted later. Some options can be overridden on a Condition Type or Line Item level where the lower level has a priority.

The following table shows a list of all configuration parameters stored in `SC_AcceleratorConfiguration`.

Name	Description	Condition Type overwritable
<b>datamartName</b>	Name of Datamart used to query and allocate the compensation value	
<b>datamartBaselineFieldName</b>	The field name of a Value Base field is used to calculate the base value. In most cases, it will be a revenue or margin field.	
<b>datamartSellerIDFieldName</b>	The field name contains an identification of a sales agent.	
<b>datamartCustomerIDFieldName</b>	The field name contains a customer ID in the Datamart.	
<b>datamartProductIDFieldName</b>	The field name contains a product ID in the Datamart.	
<b>datamartProductNameFieldName</b>	The field name contains a product name. It is used for the <i>Transaction List</i> report only.	
<b>datamartPricingDateFieldName</b>	The field name contains a pricing date in the Datamart.	
<b>datamartCurrencyFieldName</b>	The field contains the main Datamart currency.	
<b>payoutDays</b>	A number of days after the end date of the payment period when the payout happens. The parameter defines	

	the default "Payout Days" for Condition Types that do not have it filled in directly.	
<b>targetFor</b>	<p>The target defined on the line item is evaluated for each payment period (see <i>paymentPeriod</i>) or annually. The parameter defines the default "Target For" for Condition Types that do not have it filled in directly.</p> <p>Select one of these options:</p> <ul style="list-style-type: none"> <li>• <b>Payment Period</b> - Gets a base value for every period, then compares and calculates the compensation.</li> <li>• <b>Annual</b> - Gets a base value for the whole year, then compares and calculates the compensation.</li> </ul>	✓
<b>paymentPeriod</b>	<p>A parameter setting a frequency of payouts. The parameter defines the default "Payment Period" for Condition Types that do not have it filled in directly. The user can specify it also on a Line Item level.</p> <p>Values: "Monthly", "Quarterly", "Semi-Annually", "Annually"</p>	✓
<b>depositScheme</b>	<p>A parameter providing guidance on compensation accumulation across payment periods.</p> <p>If "targetFor" is "Annual", you need to define how to calculate the compensation value:</p> <ul style="list-style-type: none"> <li>• <b>Non-Cumulative</b> - The compensation value is calculated for the current period based on a cumulation of the base value, then divided by the number of periods (12 months or 4 quarters, 2 for semi-annually, 1 year).</li> <li>• <b>Cumulative</b> - The same as with calculation of Non-Cumulative above, but the compensation value of the previous period is excluded.</li> </ul>	✓

## Company Parameters

The following supporting configuration parameters are deployed with the package and are used for configuration.

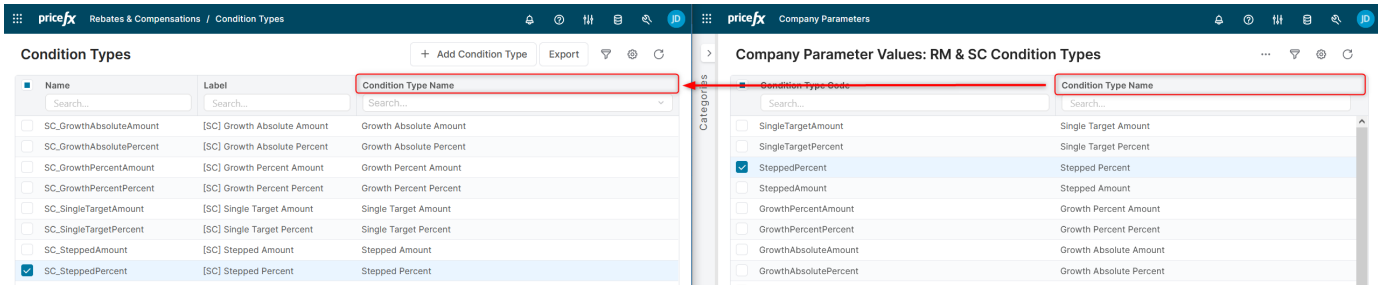
### RM\_SC\_ConditionTypes

This parameter saves the condition type code and condition type name. If the package supports a new condition type, you need to add a new row in this table.

The value of the "Condition Type Name" column will be shown in the "Condition Type Name" (attribute4) column from the Condition Types table from the Rebates module.

Column Name	Description
Condition Type Code	Must match the element name in Groovy Library "SC_CompensationTypesLib" (and "RebateTypesLib" from Rebate Manager if deployed together).

Condition Type Name Will be shown in "Condition Type Name".



### SC\_CustomerFieldMapping

This parameter maps the **Customers** master table attributes (source fields) with the datamart field names. You should check it and update based on the master table structure in your partition as explained in the configuration documentation.

Column Name	Column Label	Description
name	Customer Master Field Name	Attribute name (source field) in Customers
attribute1	Datamart Field Name	Map with a field name in Datamart

### SC\_ProductFieldMapping

This parameter maps the **Products** master table attributes (source fields) with the Datamart field names. You should check it and update based on the master table structure in your partition as explained in the configuration documentation.

Column Name	Column Label	Description
name	Product Master Field Name	Attribute name (source field) in Products
attribute1	Datamart Field Name	Map with a Field Name in Datamart

### Dependency Parameters

List of parameters that come from dependencies and have an effect on the package behavior.

- CurrencySymbols
- ApprovalCondition
- ApprovalWorkflowSetup
- Approvers
- WFExpressionVariableConfiguration
- WFLookupFilterConfiguration

### Attributes

This section enlists attributes for various elements used for the package functionality.




## Agreement Header Type Attributes


No extra attributes are used for Agreement Header Types.

## Condition Type Attributes

Condition type attributes are needed for the sales compensation logic to calculate the compensation value and agreement records.


**i** The package deploys also attributes used for functions in the Rebate Manager Package. Those are shown in light grey color.

Attribute	Attribute Name	Default Label	Type	Description	Advanced Configuration override
formulaName	formulaName	Pricing logic	Drop-down list	Sets a logic to run. The value should be "Compensation Logic".	
attribute1	CustomerSelectionLevel	Customer Selection Level	Drop-down list		
attribute2	CustomerFilterLogic	Customer Filter Logic	Text field		
attribute3	PaymentPeriod	Payment Period	Drop-down list	Defines the payment period for a condition type. If users do not set a value for this attribute, there is the "Payment Period" input in the compensation agreement with the same values.  Values: "Monthly", "Quarterly", "Semi-Annually", "Annually"	
attribute4	ConditionTypeName	Condition Type Name	Drop-down list	Sets a condition type template for the current condition type. The value is set to "Condition Type Name" in Company Parameters.	
attribute5	PayoutDays	Payout Days	Number	Enter the number of days after the end date of the payment period when the user wants the payout.	
attribute6	BaseSource	Rebate Source Name	Text field		
attribute7	BaseFieldValue	Rebate Base Field Value	Text field		
attribute8	BaseFieldDate	Rebate Base Field Date	Text field		
attribute9	TargetFor	Target For	Drop-down list	Defines the target input for a period or annually.  Values: "Payment Period", "Annual"	
attribute10	SourceType	Source Type			

			Drop-down list		
attribute11	DepositScheme	Deposit Scheme	Drop-down list	Defines the calculation type for a compensation value. If Target For = "Payment Period", users do not need to set a value for this attribute.  Values: "Cumulative", "Non-Cumulative"	
attribute12	BaseFieldCustomer	Rebate Base Field Customer ID	Text field		
attribute13	BaseFieldProduct	Rebate Base Filed Product ID	Text field		
attribute14	BaseFieldQuantity	Rebate Base Field Quantity	Text field		
attribute15	ProductFilterLogic	Product Filter Logic	Text field		

### Agreement Record Attributes

The following extra information is provided.

 The package deploys also attributes used for functions in the Rebate Manager Package. Those are shown in light grey color.

Attribute	Attribute Name	Default Label	Description	Accelerator
attribute1	CurrentBaselineValue	Current Baseline Value	Baseline value (revenue or margin ...) in this period	SC, RM
attribute2	CurrentRebate	Current Rebate		RM
attribute3	PayToID	Pay To ID	The ID of a Sales Agent who will be paid	SC, RM
attribute4	CurrentBaselineQuantity	Current Baseline Quantity		RM
attribute5	ForecastBaselineValue	Forecast Baseline Value		RM
attribute6	ForecastQuantity	Forecast Quantity		RM
attribute7	Forecast	Forecast		RM
attribute8	AccrualForecastBaselineValue	Accrual Forecast Baseline Value		RM
attribute9	AccrualForecast	Accrual Forecast		RM
attribute10	TruesUp	Trues Up		RM
attribute11	AccrualMethod	Accrual Method		RM

attribute12	SalesGoalIncreasePct	Sales Goal Increase %		RM
attribute13	ForecastType	Forecast Type		RM
attribute14	AccrualForecastQuantity	Accrual Forecast Quantity		RM
attribute15	GeneralFilter	General Filter	Filter which was used as Compensation Agreement input.	SC
attribute16	Currency	Currency	Compensation Currency	SC
attribute17	CurrentCompensation	Current Compensation	Amount of money which will be paid	SC

### Sellers Attributes

The following extra information is expected to be in the Sellers table.

Attribute	Label	Type	Is Mandatory?
sellerId	Seller ID	String	✓
name	Seller Name	String	✓
attribute1	First Name	String	✗
attribute2	Surname	String	✗
attribute3	Reports To	String	✓
attribute4	Pricefx User Account Id	String	✓
attribute5	Active	String	✗

### Publishing Templates

The following templates are provided.

Looks like the page 'Installation (SC)' does not contain Multiple Excerpts (excerpt with key) or the given key 'sc\_components\_publishing\_templates' is wrong. If problem persists, you can contact support at support@bitwelt.atlassian.net.

### Dashboards

Table of dashboards included in the package.

Name	Logic	Description
Compensation Dashboard	SC_Dashboard_Compensations	Shows a compensations value of each compensation type from the beginning of the year to the current date.

### Access Management

A list of predefined business roles and user groups with minimal access rights.

### Business Roles

Business Role	Assigned User Roles
SC_Administrator	DASHBOARDADMIN, I18NADMIN, PA_DATAANALYZER, PA_DATAMANAGER, PA_SCHEMAEDITOR, PB_CUSTOMERS_RO, PB_PARAMETERS, PB_PRODUCTS_RO, RM_REBATEAGREEMENTS_ADMIN, RM_REBATEMANAGER, RM_REBATERECORDS_ADMIN, RM_SELLERSMANAGER, WF_ADMIN, WF_BUILDER
SC_SalesManager	PB_CUSTOMERS_RO, PB_PRODUCTS_RO, RM_REBATEAGREEMENTS, RM_REBATERECORDS
SC_SalesAgent	PB_CUSTOMERS_RO, PB_PRODUCTS_RO, RM_REBATEAGREEMENTS_RO, RM_REBATERECORDS_RO

 For more details see [User Roles](#) and <https://pricefx.atlassian.net/wiki/spaces/ACCDEV/pages/3807805649/Product+Info+SC#Business-Roles>.

### User Groups

- SC\_SalesManager

### Other Components

### Approval Workflows

The Approval Workflow Library is part of the package as a dependency. It is extended with new types for sales compensation context. For details see [Accelerate Approval Workflow Package \(AWP\)](#).

Workflow Type	Workflow Name	Logic Name
SCA	Compensation <b>Agreement</b> Approval Workflow	SC_SalesCompensationAgreement
SCR	Agreement <b>Record</b> Approval Workflow	SC_AgreementRecord
Adjustment		

### Workflow Variables (SC)

This page provides a list of variables used for building conditions for approval workflows inside the package. It provides support information for [Configuration \(SC\)](#).

- [Compensation Agreement Approval Workflows](#)
- [Agreement Record Approval Workflows](#)

### Compensation Agreement Approval Workflows

SCA is the name of Workflow Type that must be used for workflows you build on top of any Compensation Agreement. SCA must be used in ApprovalWorkflowSetup, ApprovalCondition and Approvers tables inside Company Parameters.

A list of all possible variables is available also in [Element Name in Accelerators](#) from Approval Workflow Package.

Header - Label	Element Name	Example
Start Date	startDate	startDate == "2021-01-31"
End Date	endDate	endDate == "2021-01-31"
Payout Date	payoutDate	payoutDate == "2021-01-31"
Seller	seller	Seller == "SC-001"
Currency	currency	header.currency == "EUR"
<b>Header - Compensation Detail</b>		
Previous Compensation	previousCompensation	previousCompensation >= 10000
Current Compensation	currentCompensation	currentCompensation >= 50000
<b>Line Item - Label</b>		
Condition Type	rebateType	rebateType = "SC_SingleTargerPercent"
<b>Input</b>		
Target	line.Target	"line.Target" > 100
Compensation	line.Compensation	"line.Compensation" == 2.0
Compensation %	line.Compensation %	"line.Compensation %" > 10
Payment Period	line.Payment Period	"line.Payment Period" == "Annually"
<b>Line Item - Compensation Detail</b>		
Previous Baseline Value	line.PreviousBaselineValue	"line.PreviousBaselineValue" > 127
Current Baseline Value	line.CurrentBaselineValue	"line.CurrentBaselineValue" > 1200
Previous Compensation	line.PreviousCompensation	"line.PreviousCompensation" > 1000
Current Compensation	line.CurrentCompensation	"line.CurrentCompensation" >= 1200
<b>Line Item - Compensation Type Info</b>		

Target For	line.TargetFor	"line.TargetFor" == "Annual"
Deposit Scheme	line.DepositScheme	"line.DepositScheme" == "Non-Cumulative"
Payment Period	line.PaymentPeriod	"line.PaymentPeriod" == "Monthly"
Seller Name	line.SellerName	"line.SellerName" == "John Doe"
General Filter	<not supported>	
Customer(s)	<not supported>	
Product(s)	<not supported>	

### Agreement Record Approval Workflows

SCR is the name of Workflow Type that must be used for workflows you build on top of any Agreement Record. SCR must be used in ApprovalWorkflowSetup, ApprovalCondition and Approvers tables inside Company Parameters.

A list of all possible variables is available also in [Element Name in Accelerators](#) from Approval Workflow Package.

When the condition is set up, use the field name and do not add any prefixes; see the table below with examples having specific meaning for this package.

Label	Element Name	Example
Current Baseline Value	attribute1	attribute3 > 1000
Pay To Id	attribute3	attribute3 == "SC-001"
Currency	attribute16	attribute16 == "EUR"
Current Compensation	attribute17	attribute17 >= 1500

### Upgrade (SC)

This tutorial will guide you through the upgrade of the Sales Compensation Accelerator.

In this section:

- [Pre-requisites](#)
- [Upgrade Steps](#)

#### Pre-requisites

Before you start, ensure that you have:

- Access to a partition on the Pricefx server (8.0 or newer). You will need:
  - Server URL
  - Partition name

- Username and password for a partition user with sufficient rights for using the Accelerator
- Access to Pricefx PlatformManager
  - Username and password for PlatformManager user
  - The user must have the following permissions for your partition (to which you plan to deploy the Accelerator):
    - Permission *Marketplace Templates - deploy*
- Familiarize yourself with the steps required after the upgrade - [Manual Upgrade Steps \(SC\)](#).
- Optionally, read about changes in [Release Notes \(SC\)](#).

## Upgrade Steps

Select Partition for Deployment and upgrade logics.

1. In PlatformManager, navigate to **Marketplace** > **Accelerator Packages**, find the *Sales Compensation - Upgrade*.
2. Click **Deploy** and select a partition to which you want to upgrade.
  - Only logics are deployed. The configuration remains without changes.
3. Click **Deploy**.
4. A warning dialogue will appear. After you read the warning text and you agree with the conditions, you can click **Continue**.
5. The first part of the upgrade is complete. Go to your partition and continue with the manual steps required after the upgrade.
 

The exact process depends on the original version of the package, you will find all details in [Manual Upgrade Steps \(SC\)](#).

## Troubleshooting (SC)

In this section, you can find troubleshooting tips that will help you during the support procedure.

### Where do I find a package version?

For efficient communication with the support team, you should know which package version is used.

1. Go to **Company Parameters**.
2. Search for the 'deployedAccelerators' parameter.
3. Find the key 'sales-compensation', its value contains 'templateVersion'.
4. Version is a value stored in 'templateVersion'.

## Known Limitations (SC)

### Adjustments Limits

In versions 1.4.0, 1.5.0, when you submit the Adjustment, with filters on Transactions data returning data exceeding 25 000 rows, you will often get timeout. This issue will be resolved once the support for mass addOrUpdate for COARs is added: [PFUN-17298](#) - Support COAR type in mass api.addOrUpdate() operation **RESOLVED** .

### Warnings in Compensation Plan

In versions 1.4.0, 1.5.0, warnings on the Compensation Plan level are not displayed properly. There is a product enhancement request which is approved and scheduled to be tackled in 2023:

[PM-2369](#) - (BE) api.addWarning doesn't work in Sales Compensation and Rebate Manager **APPROVED** .

These warnings should be displayed if:

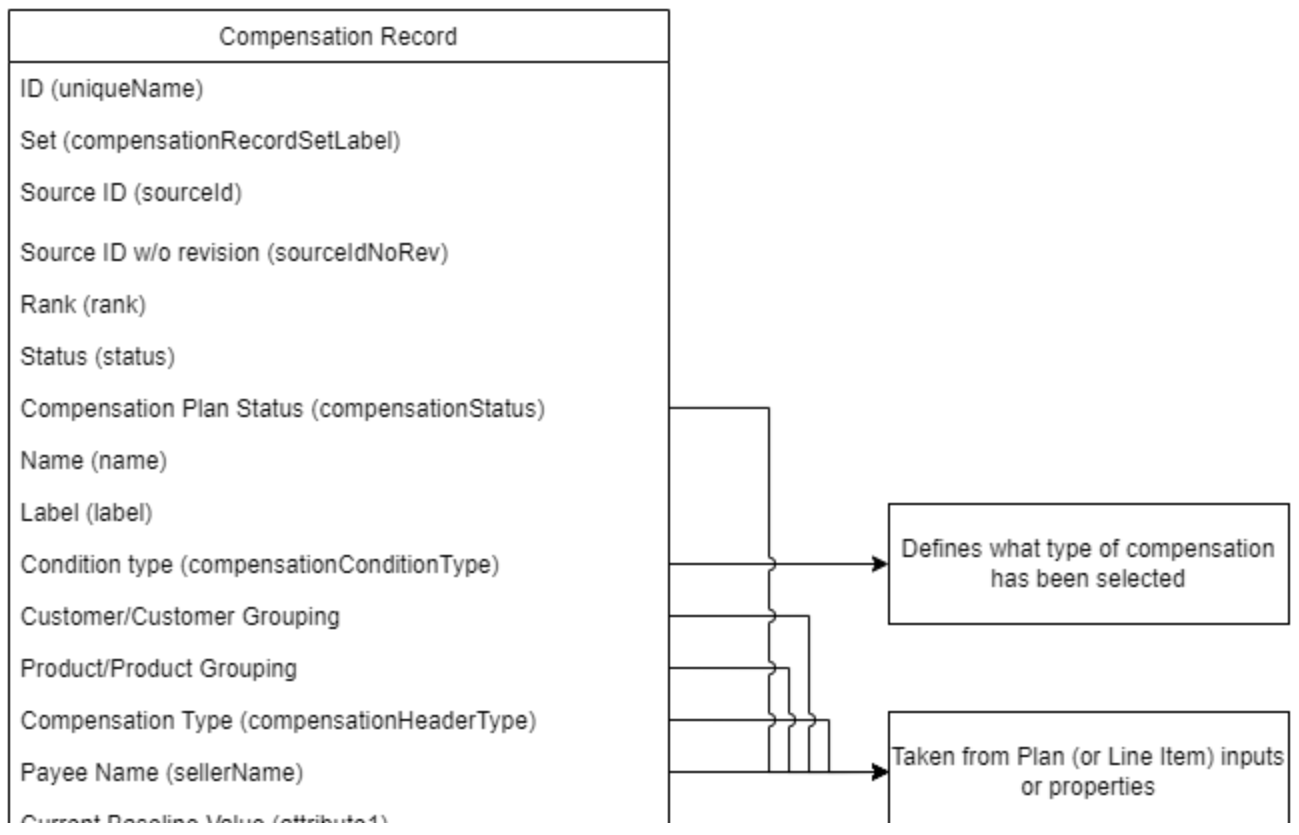
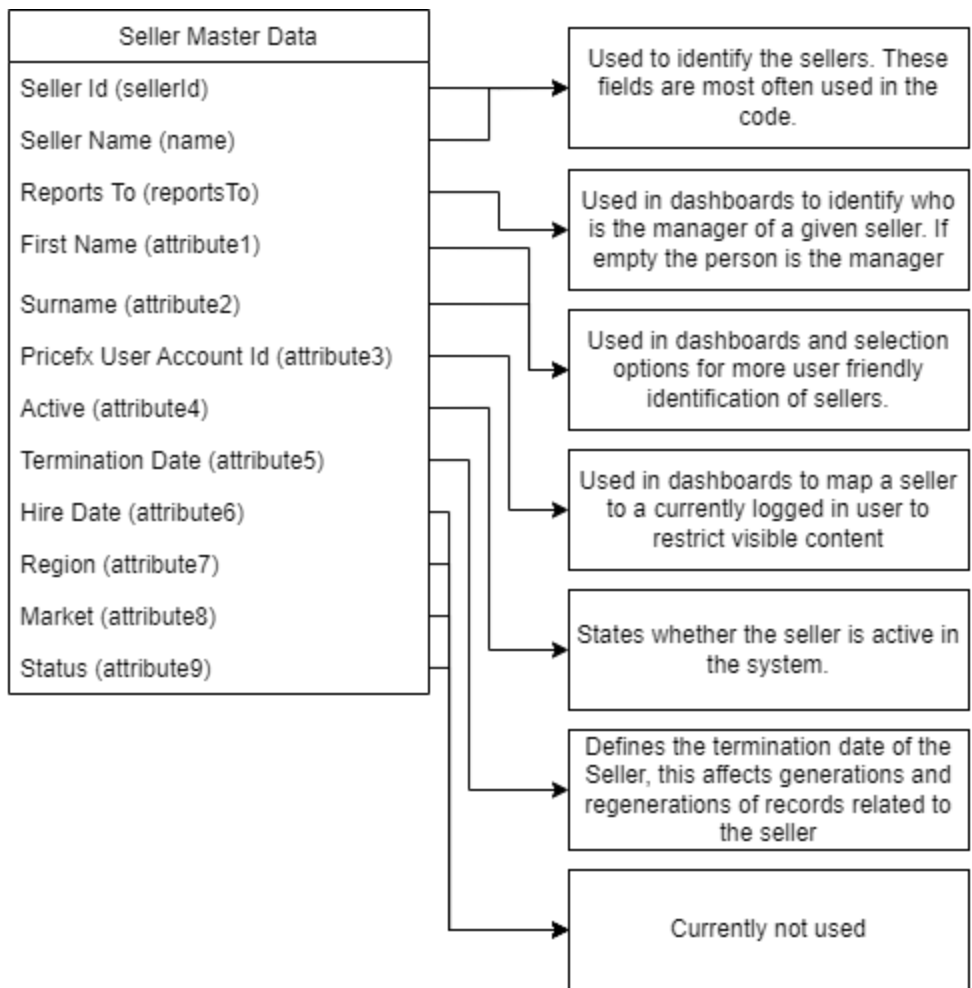
- Any of the sellers selected in the Seller input are terminated.
- Any of the sellers selected in the Seller input have a missing salary.
- Any of the sellers selected in the Team input have been skipped because of invalid validity dates.
- Any of the sellers selected in the Team input have been skipped because validity dates contradict planned start/end date settings.

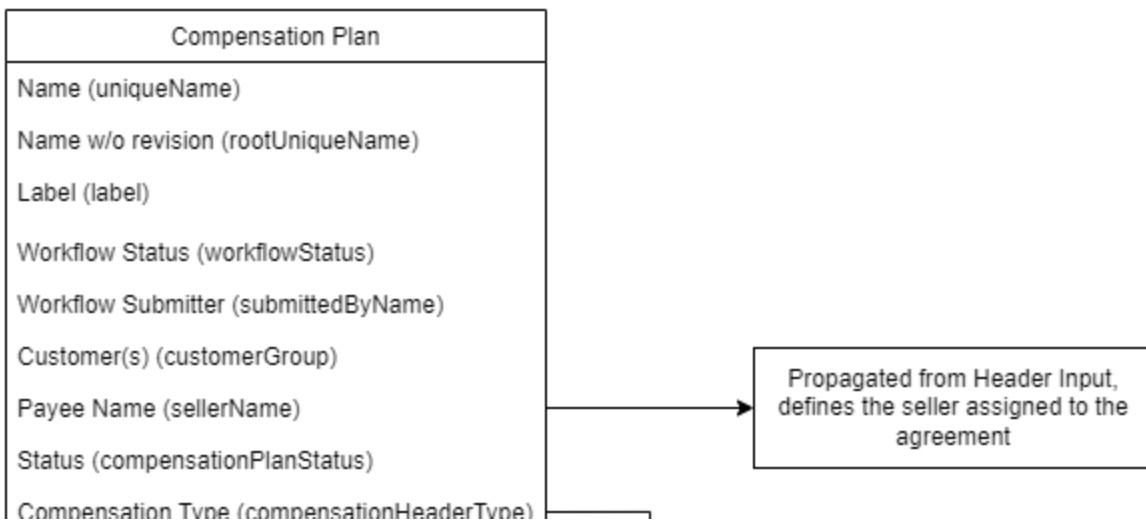
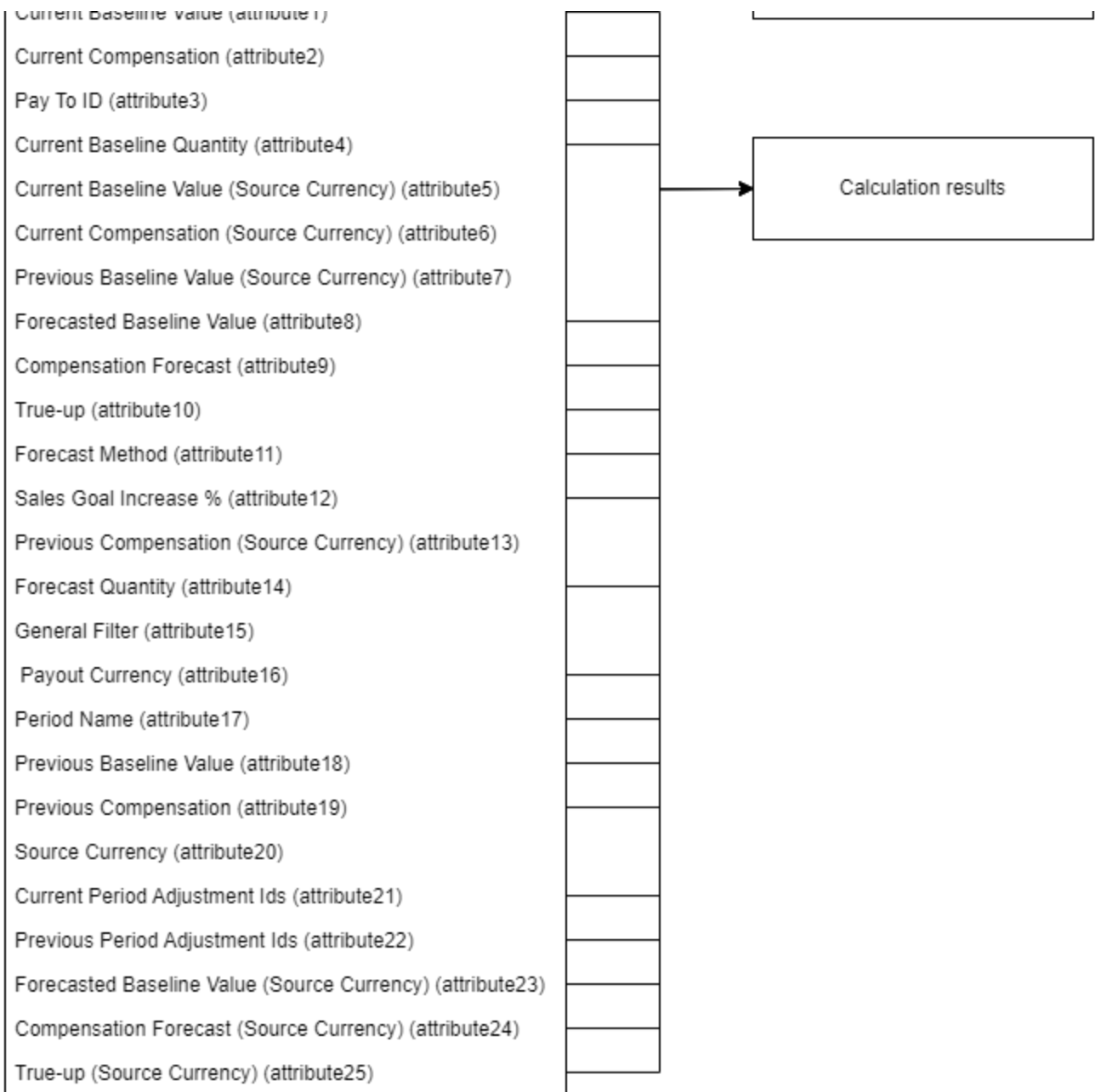
## Technical Information (SC)

- [Architecture Documentation \(SC\)](#)
- [Logic Documentation \(SC\)](#)
- [Flow Diagrams \(SC\)](#)

## Architecture Documentation (SC)

### System Objects





-----

Defines the type of the agreement.  
Used for filtering out non-SC related agreements, for example in dashboards

Compensation Type
Name (uniqueName)
Header Logic (headerFormulaName)
Compensation Plan Workflow (workflowFormulaName)
Compensation Record Workflow (scRecordWorkflowFormulaName)
Filtering logic (scTypeFilterFormulaName)

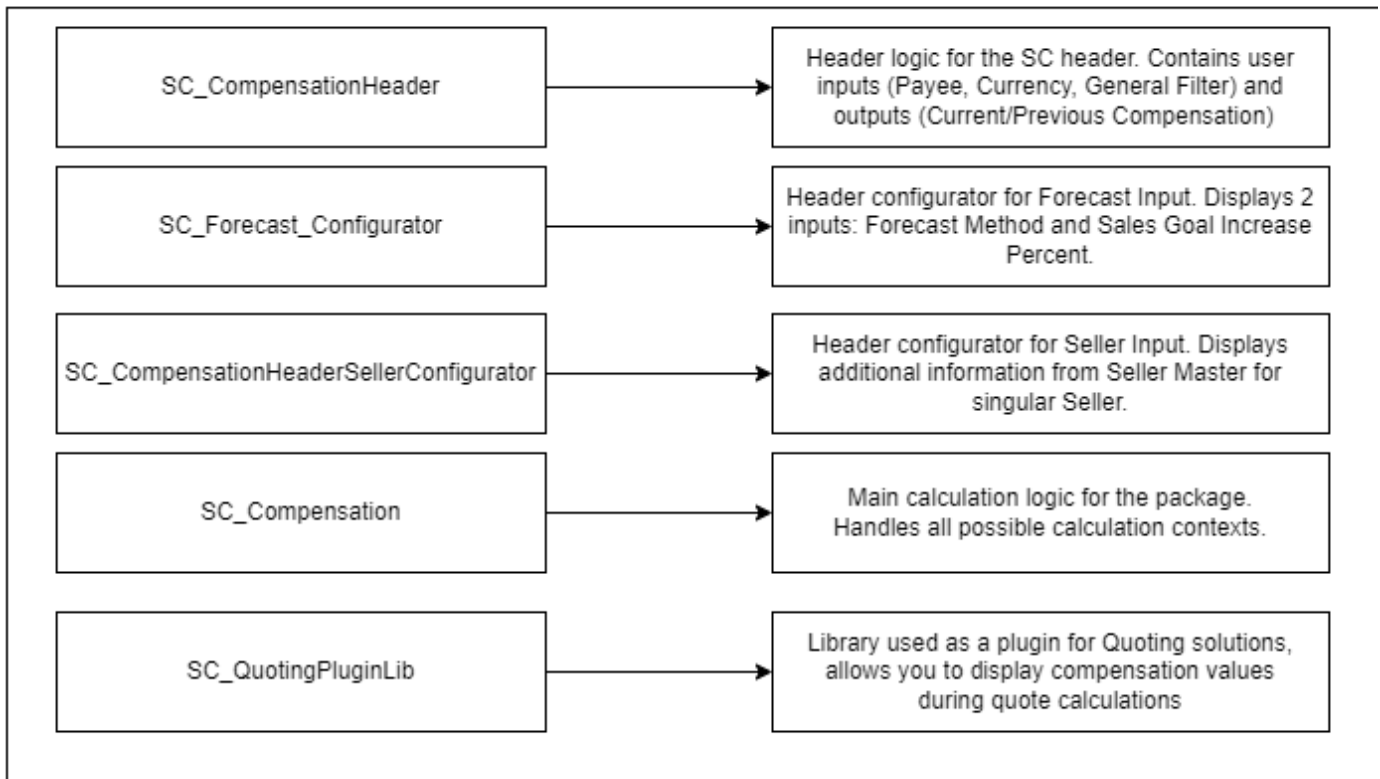
Defines the logic used for all header inputs and output handling

Defines the logics used for workflow handling on the agreement and record level

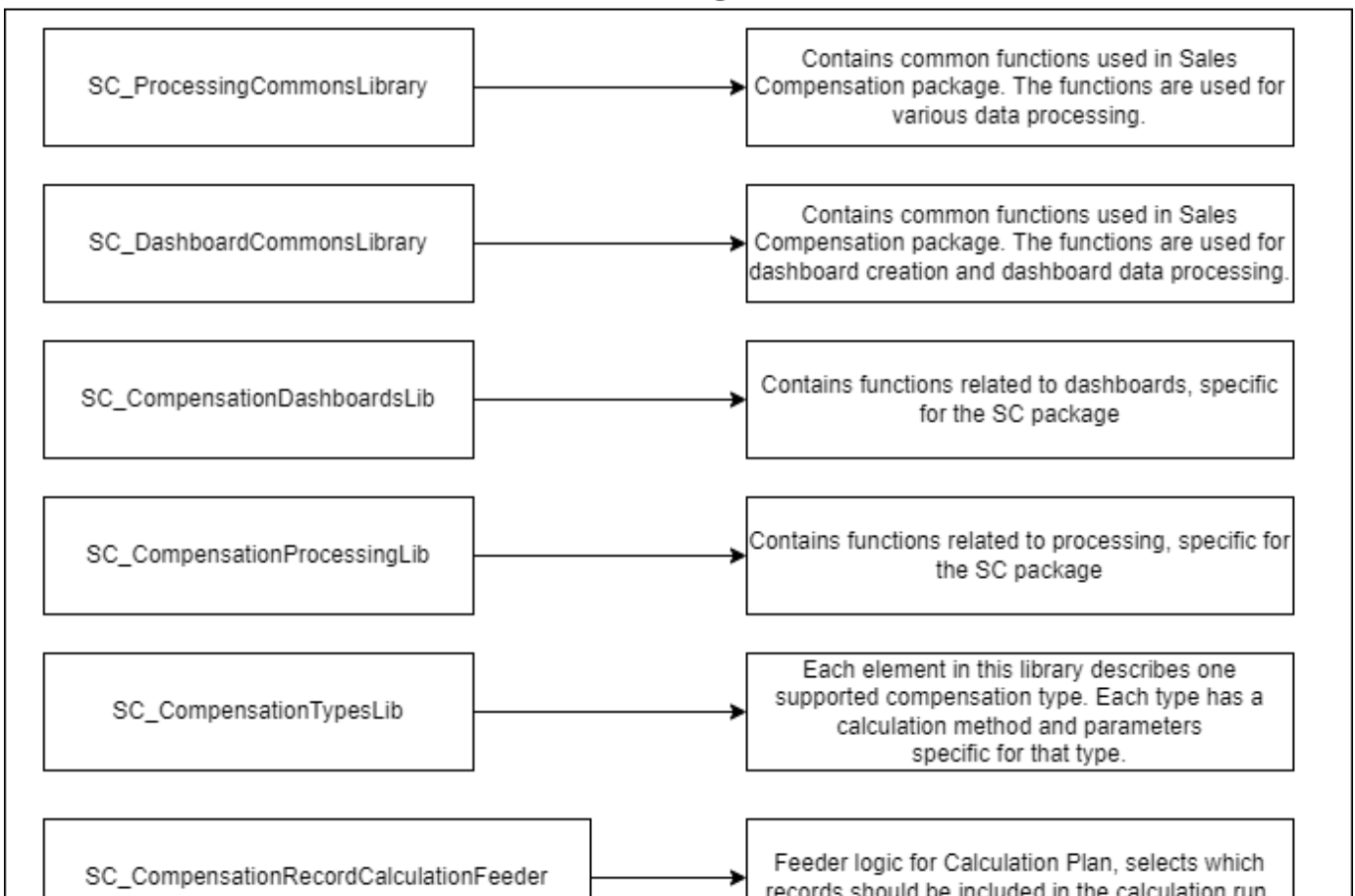
Defines the logics used for filtering available line items to be selected in agreements

## Logics

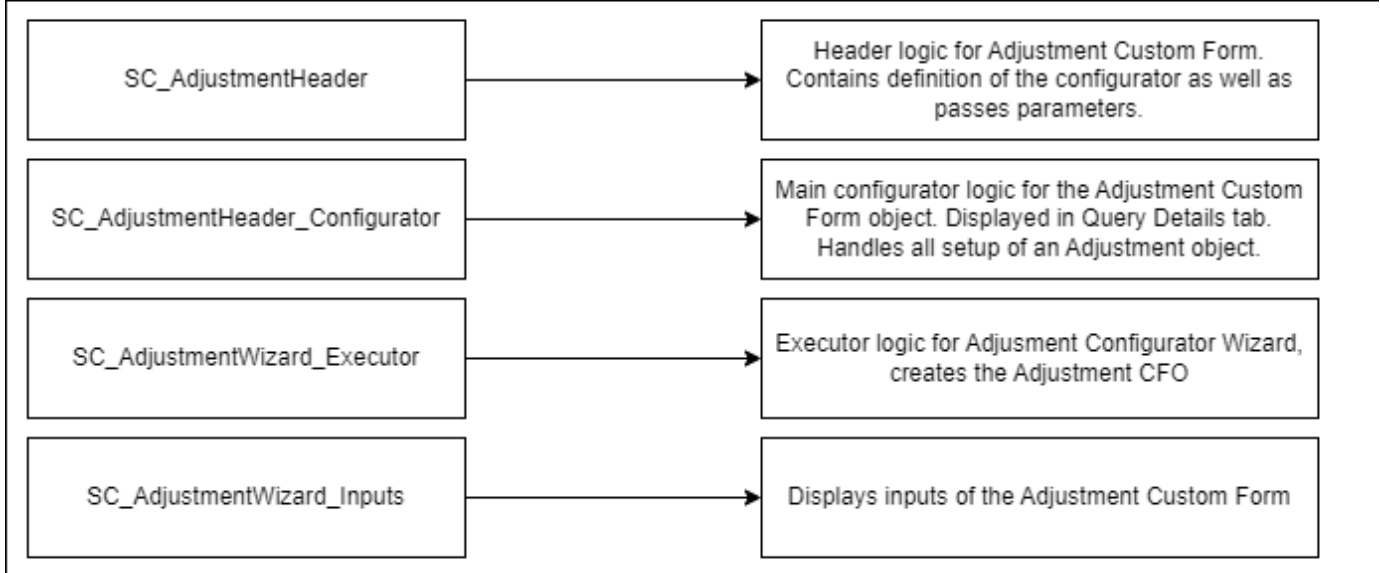
### Core logics



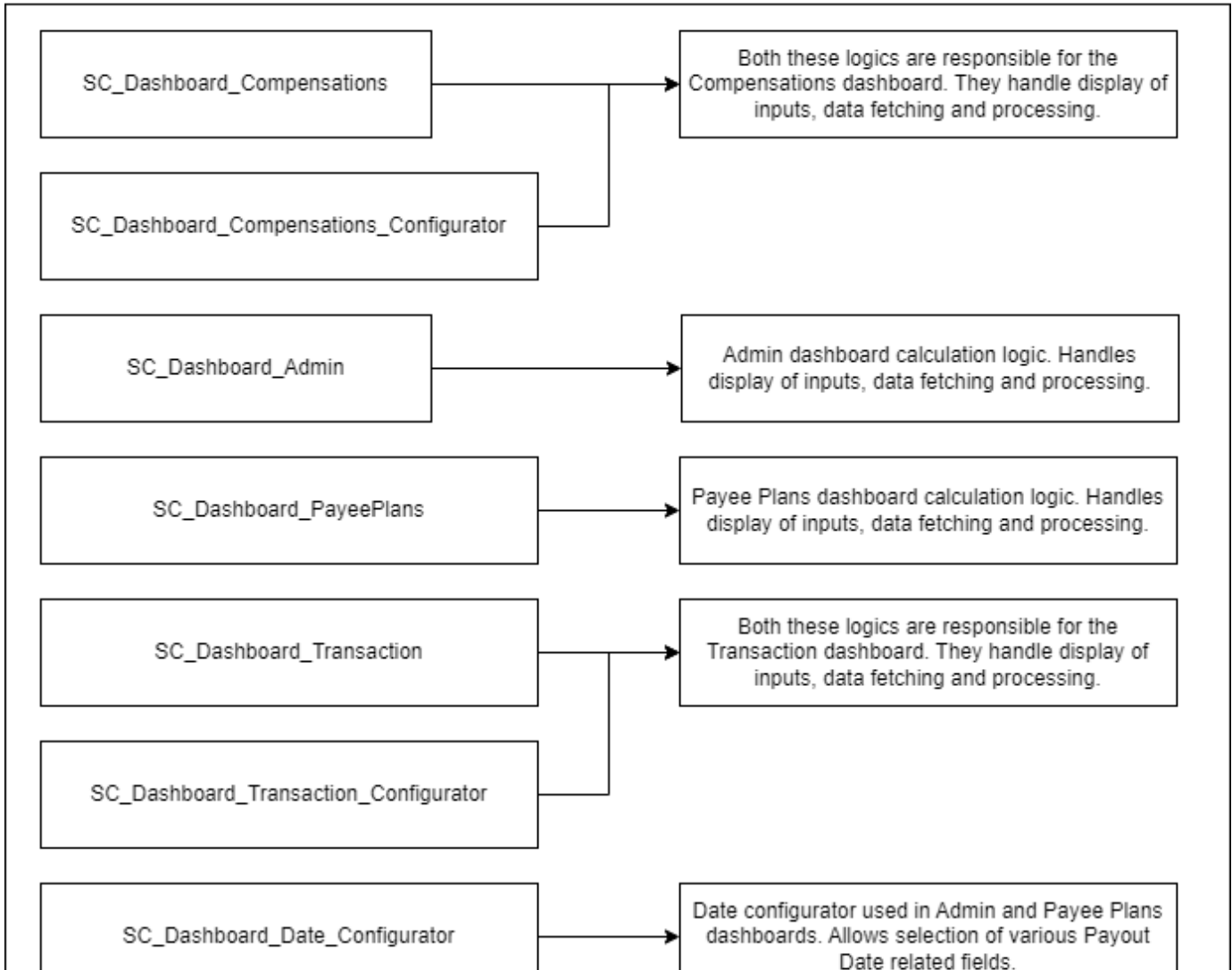
### Backend logics

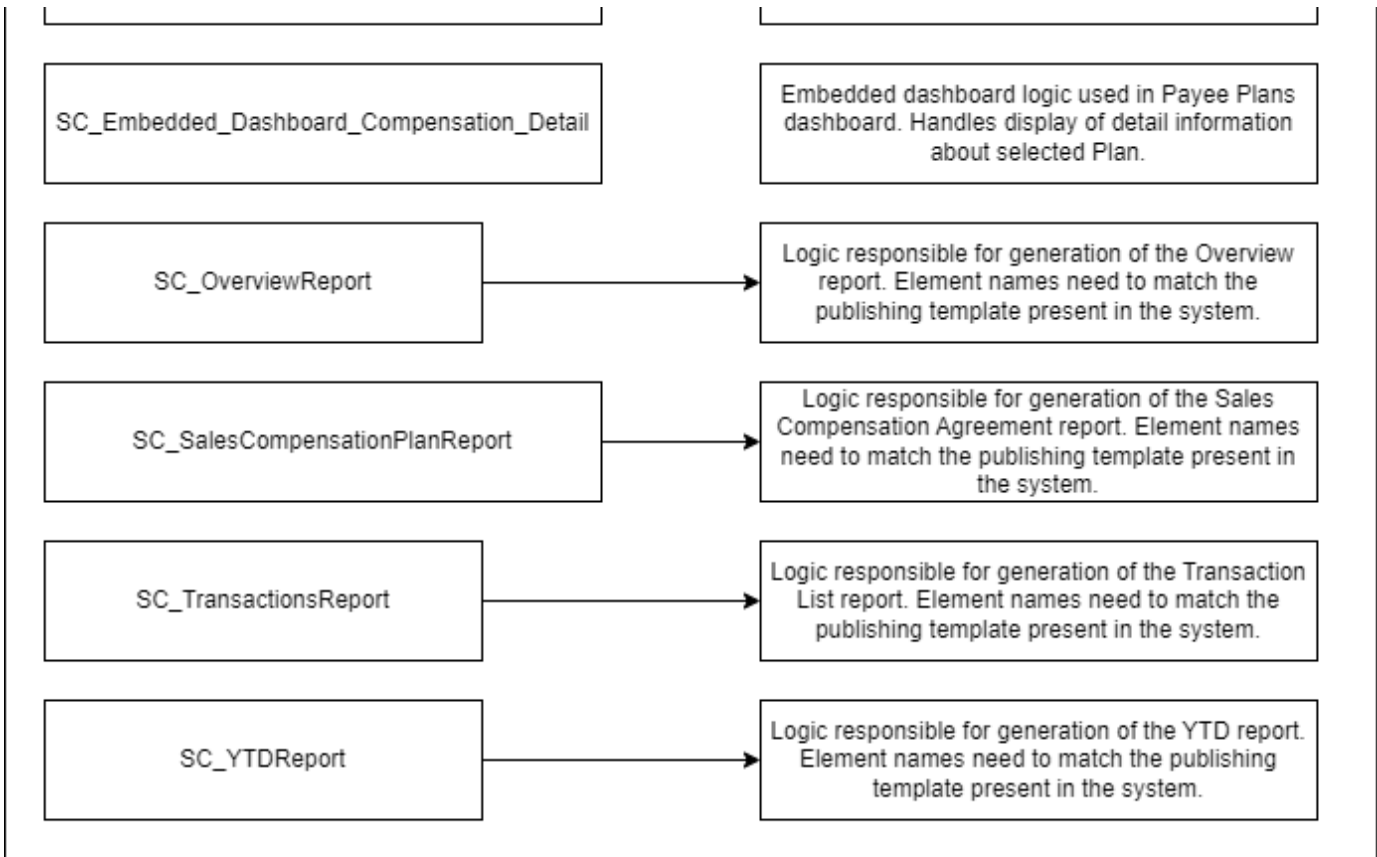


### Adjustment logics

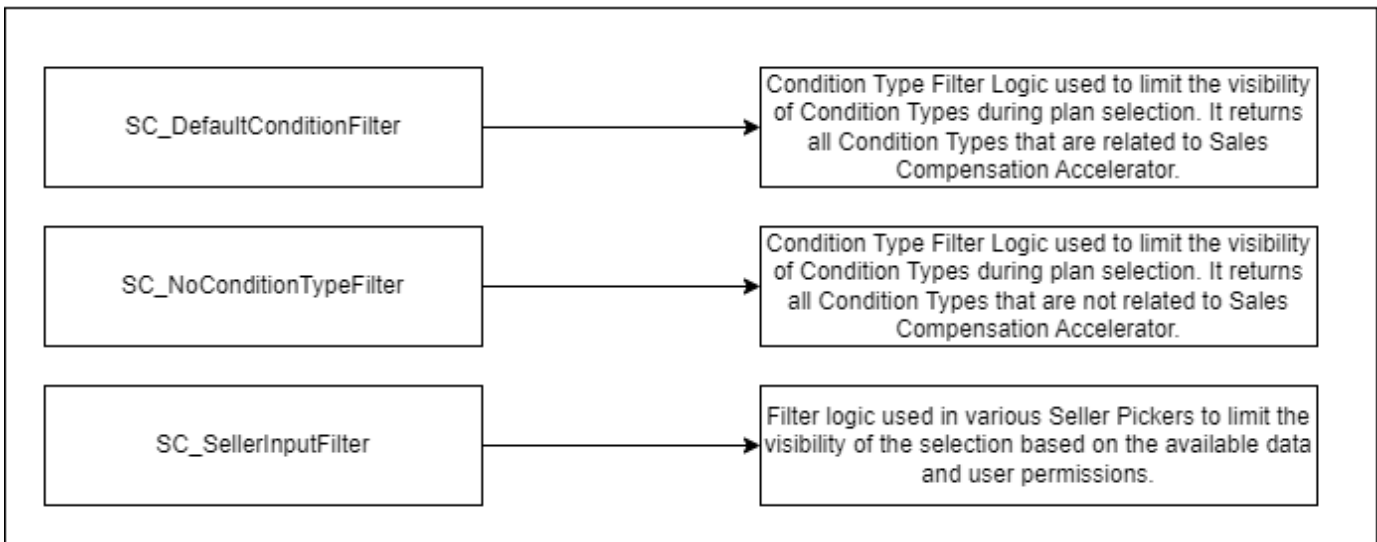


### Output logics

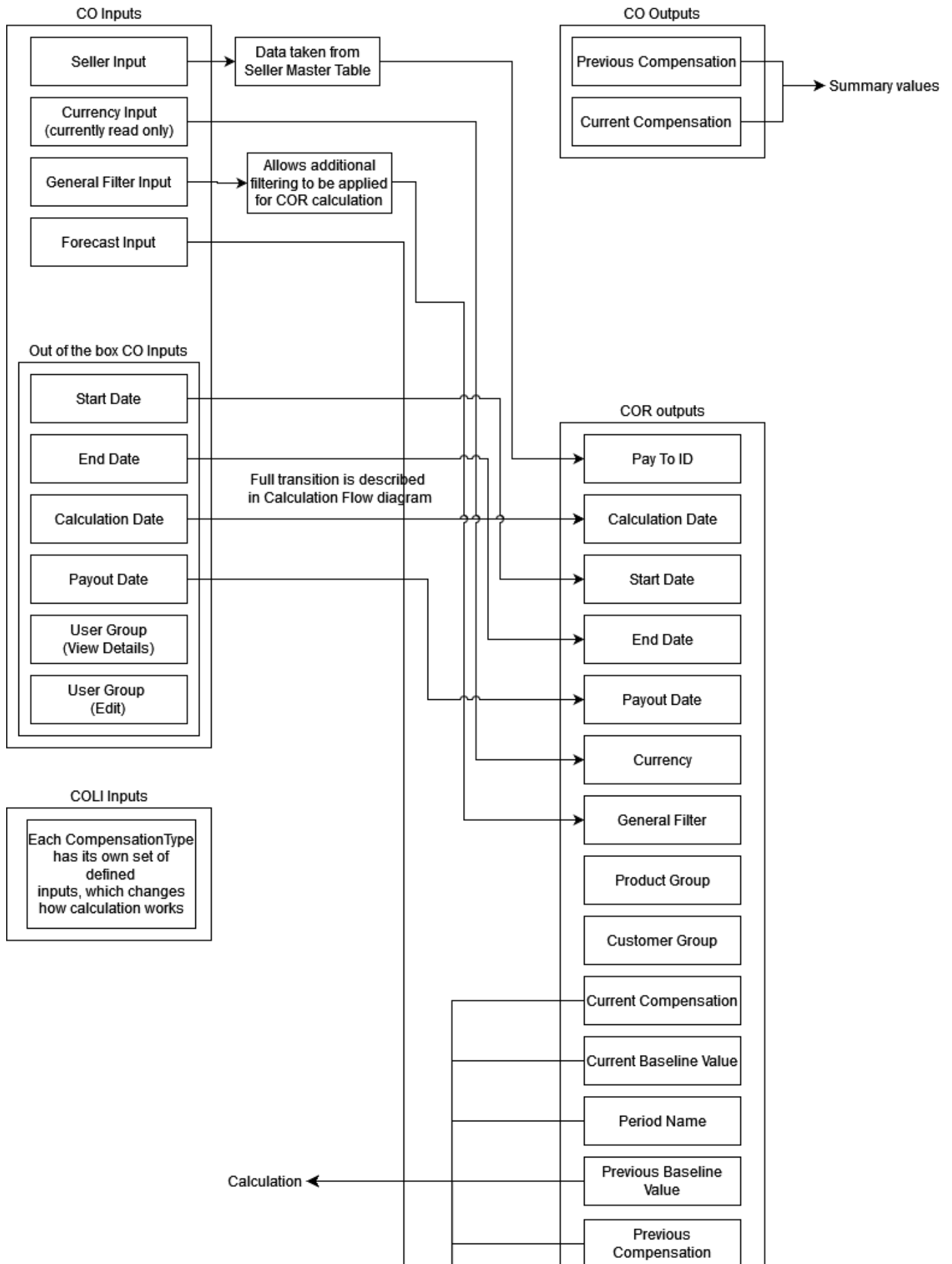


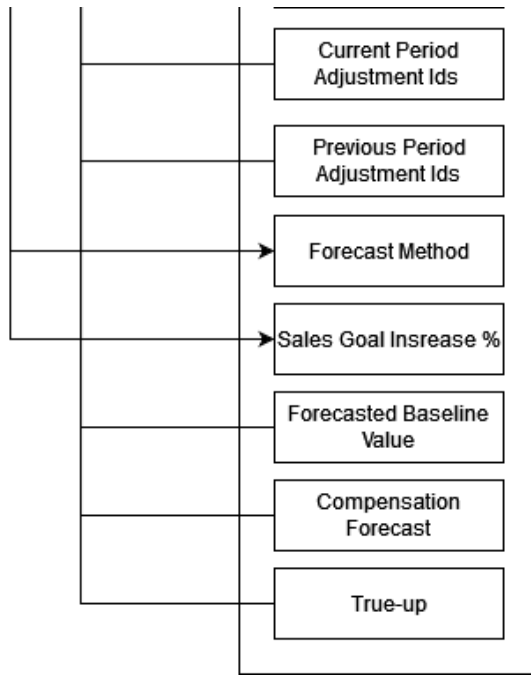


Filter logics

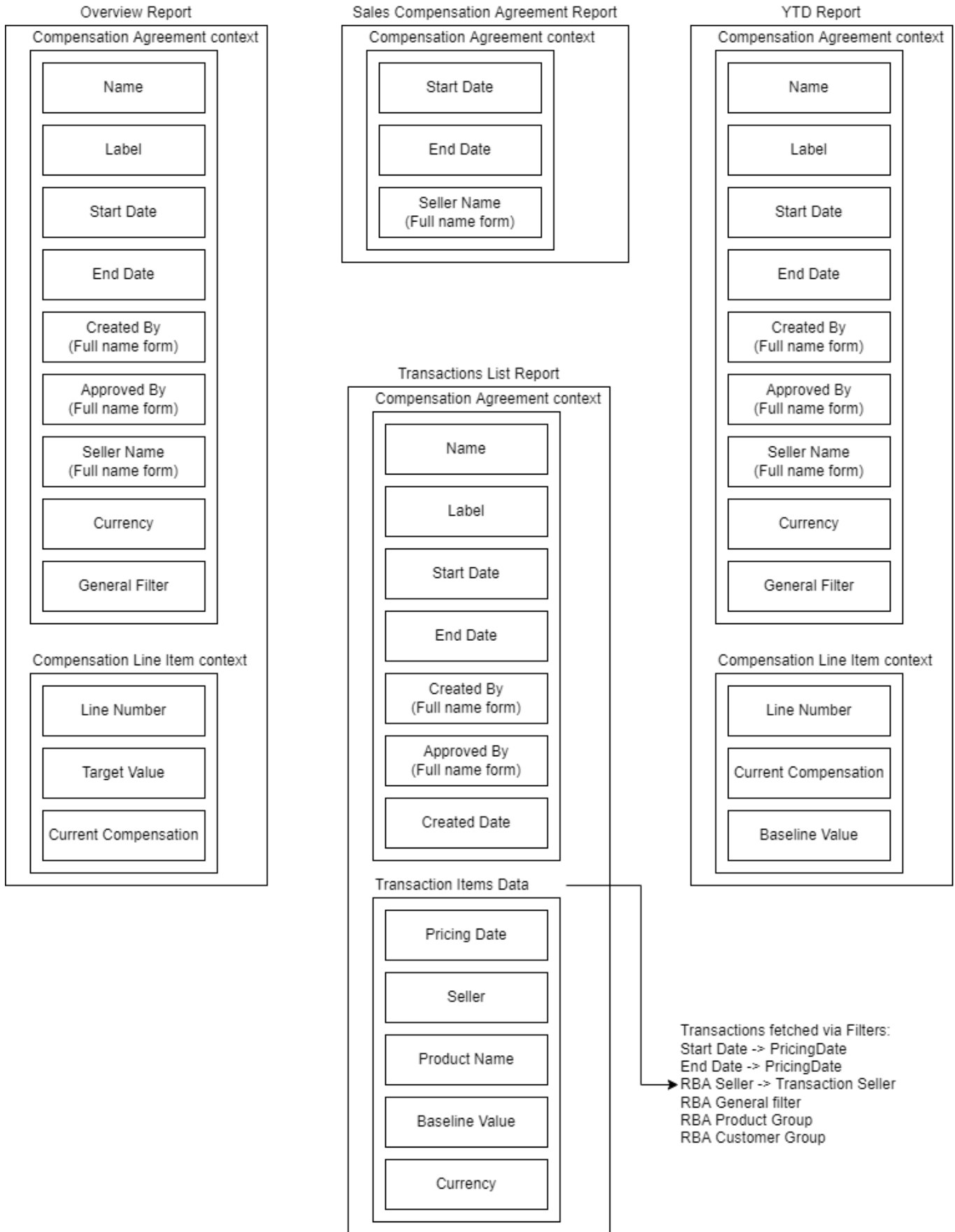


## Simplified Compensation Plan Compensation Record Data Flow





# Reports Data Flow



## Logic Documentation (SC)

- [SC\\_CompensationDashboardsLib](#)
- [SC\\_CompensationProcessingLib](#)
- [SC\\_CompensationTypesLib](#)
- [SC\\_CompensationHeader](#)
- [SC\\_CompensationHeaderSellerConfigurator](#)
- [SC\\_Forecast\\_Configurator](#)
- [SC\\_Compensation](#)
- [SC\\_Dashboard\\_Compensations\\_Configurator](#)
- [SC\\_Dashboard\\_Compensations](#)
- [SC\\_Dashboard\\_Transactions\\_Configurator](#)
- [SC\\_Dashboard\\_Transaction](#)
- [SC\\_Dashboard\\_Date\\_Configurator](#)
- [SC\\_Dashboard\\_Admin](#)
- [SC\\_Dashboard\\_PayeePlans](#)
- [SC\\_EmbeddedCompensationDetailDashboard](#)
- [SC\\_Compensation\\_RecordCalculationFeeder](#)
- [SC\\_AdjustmentHeader](#)
- [SC\\_AdjustmentHeaderConfigurator](#)
- [SC\\_AdjustmentWizardExecutor](#)
- [SC\\_AdjustmentWizardInputs](#)
- [SC\\_QuotingPluginLib](#)
- [SC\\_OverviewReport](#)
- [SC\\_SalesCompensationPlanReport](#)
- [SC\\_YTDReport](#)
- [SC\\_TransactionsReport](#)

### SC\_CompensationDashboardsLib

This logic contains common utility functions that are used by the Sales Compensation package. The library focuses on dashboards and operations related to them - mostly data fetching and processing.

See also the [architecture documentation](#).

#### Elements Description

- **ConstConfig** - Contains definitions of global constant variables that are used throughout the logic. Any configuration-like constants should be stored here to avoid hard-coding and to allow for better maintainability of the solution.
- **ConfigurationUtils** - Extension of the common element from SC\_CompensationDashboardsLib, contains SC package specific configuration structures and loading operations. This is the place where all Advanced Configuration setup fetching and storage takes place.
- **CompensationTypeUtils** - Contains methods related to fetching and accessing data from the Compensation Type object (COCT).
- **CompensationRecordUtils** - Contains methods related to fetching and accessing data from the Compensation Record object (COR). This util also defines how the Compensation Record data are structured.
- **CompensationAgreementUtils** - Contains methods related to fetching data from the Compensation Plan object (CO).

- **CompensationAgreementLineItemUtils** - Defines methods for accessing and building the structures for Compensation Plan Line Items ("COLI" currently). The data is extracted from the already fetched and processed Compensation Plan objects.
- **ConfiguratorUtils** - Contains methods used in various configurators for dashboards. These are mostly input generation methods.
- **DateUtils** - Contains methods related to date in any way, such as filter generation for years or start /end dates.
- **InputUtils** - Contains methods related to common user dashboard inputs.
- **InputValidationUtils** - Contains methods used for input validation in dashboards - such as checking if input is not empty or has valid values.
- **FormatUtils** - Contains methods used for formatting output values to appropriate money, percentage etc. formats. Currency symbol generation methods are also stored here.
- **PortletUtils** - Contains methods that generate ready to use portlets, such as summary portlets.
- **QueryUtils** - Contains methods related to any type of Datamart / Data Source lookups and queries.
- **CurrencyUtils** - Contains methods related to currency (formats, fetching or label generation).
- **PaybackTypeUtils** - Contains methods related to fetching and processing system data for various types of compensation types.
- **PaybackAgreementUtils** - Contains everything related to compensation plans and also contains methods for generation of data structures as well as for fetching the data and displaying it.
- **UserUtils** - Contains methods related to users, such as checking if a user belongs to a given user group.
- **SellerUtils** - Contains methods related to Seller master data, such as filtering based on user groups.

## SC\_CompensationProcessingLib

This logic contains common utility functions that are used by the Sales Compensation package. The library focuses on data processing related to the core of this package - that means calculation of plans and records mostly.

See also the [architecture documentation](#).

### Elements Description

- **ConstConfig** - Contains definitions of global constant variables that are used throughout the logic. Any configuration-like constants should be stored here to avoid hard-coding and to allow for better maintainability of the solution.
- **ConfigManager** - Strictly defined structure that allows management of solution configuration. The benefit is that the whole configuration is contained within one "class", it cannot be modified and is cached.
- **CompensationInputParameter** - "Class" like element that provides a creation method for a structure called InputParameter which stores all user inputs for further processing or access.
- **InputUtils** - Extension of SC\_CompensationProcessingLib InputUtils element with Sales Compensation related methods. Contains mostly methods for creating inputs on the line item level.
- **InputValidationUtils** - Extension of SC\_CompensationProcessingLib InputValidationUtils, contains checks for Sales Compensation specific input types and their validation.
- **QueryUtils** - Contains methods related to Datamart queries, mostly for transaction type data.
- **FormatUtils** - Extension of SC\_CompensationProcessingLib FormatUtils, extends the formatting by coloring options.
- **CalculationUtils** - Contains methods for final steps of the core logic, creation of Compensation Records and enrichment of Compensation Record data based on the calculation base.
- **CacheDataManagementUtils** - Contains methods for various caching purposes.
- **CompensationDataCalculator** - Contains methods related to main calculations, that is for example calculation of the current and previous compensation in various contexts. Calculations related to transaction data can also be found here.

- **CompensationAgreementUtils** - Extension of SC\_CompensationProcessingLib PaybackAgreementUtils with Sales Compensation related methods. The main addition is the focus on Compensation Agreement objects as well as fetching and processing data coming from those objects.
- **CompensationRecordUtils** - Extension of SC\_CompensationProcessingLib PaybackRecordUtils with Sales Compensation related methods. Defines a way of fetching the CompensationRecords as well as provides a way of building filters for the fetch based on the provided data.
- **BasicCompensationValuesDataUtils** - Helper util element that supports compensation calculations, summaries etc.
- **CompensationCalculationParameter** - "Class" like element that provides a creation method for a structure called CalculationParameter which stores all calculation related closures for further processing or access.
- **CompensationValueCalculator** - Contains methods that describe the calculation methods for various types of compensations: target, stepped, growth or multi.
- **FilterUtils** - Extension of SC\_CompensationProcessingLib FilterUtils with Sales Compensation related methods. Mostly related to building a product/customer filter structure.
- **ReportUtils** - Contains utils that are used in all reports in the package. This util includes data fetches for reports as well as generation of report specific labels.
- **SellerUtils** - Contains methods that are used for all Seller related data manipulation - that is fetching, retrieval based on some specific fields etc.
- **AdjustmentUtils** - Contains methods that are related to Adjustment Custom Form, their creation, processing and access of data.
- **AdjustmentConfiguratorUtils** - Contains methods used in Adjustment related configurators (Adjustment Form and Adjustment Creation Wizard) and mainly consists of input generation methods.
- **AdjustmentValidationUtils** - Contains methods related to validation of Adjustment inputs.
- **CompensationTypeUtils** - Contains methods related to CompensationTypes. Provides methods for easy access to all necessary information of a given CompensationType, as well as processing methods for its data such as reading Target or Compensation values from line items.
- **AccrualForecastUtils** - Contains calculation methods related to Forecasts.
- **ChartUtils** - Contains methods that generate charts. These charts are displayed in the outputs of the Compensation Plan, hence they have been separated from the SC\_CompensationDashboardsLib.
- **DateUtils** - Contains methods related mostly to date and period definition. Reuses and extends a lot of methods from SharedLib.
- **PeriodDataUtils** - Related to DateUtils, this util handles the Period structure creation and data access to it.
- **MathUtils** - Contains math related methods, such as percentage calculations.

## SC\_CompensationTypesLib

The library stores all types of currently supported Compensation Types that can be added as line items in Compensation Agreements. Each type has a strictly defined structure: Calculation Type, Target Type, Target Value Type, Compensation Value Type as well as a way to calculate the compensation. The element names match 1-1 the Condition Types defined in the system and should be carefully maintained if any changes arise.

See also the [architecture documentation](#).

### Elements Description

- SingleTargetAmount
- SingleTargetPercent
- MultiTargetAmount
- MultiTargetPercent
- SteppedAmount

- SteppedPercent
- GrowthAbsoluteAmount
- GrowthAbsolutePercent
- GrowthPercentAmount
- GrowthPercentPercent
- FlatBonus
- MultiTargetBonus
- SteppedBonus
- VariablePayLinearAmount
- VariablePayMultiTargetAmount
- VariablePaySteppedAmount

## SC\_CompensationHeader

This is currently the only header logic that defines the header behavior of Compensation Plan. The logic defines the inputs and outputs from the header and further operations that happen in the main core logic.

See also the [architecture documentation](#).

### Elements Description

- **ConfigManager** - Initialization of a ConfigManager structure defined in SC\_CompensationProcessingLib.
- **HeaderUtils** - Contains methods for creating header inputs and outputs, main codebase of the logic. Calculation methods for Previous and Current compensation can also be found here.
- **SyntaxCheckAbort**
- **HeaderInputs** - Contains initialization of all header inputs using HeaderUtils (even the hidden ones, such as the type of the header).
- **HeaderOutputs** - Adds outputs to the header using HeaderUtils.

## SC\_CompensationHeaderSellerConfigurator

This is the configurator for Seller selection on the header level. It is used only in header types that contain a singular Seller. It displays additional information that is configured to be displayed in SC\_HeaderSellerDetailsMapping Company Parameter table.

See also the [architecture documentation](#).

### Elements Description

- **SellerInput** - Returns a user input that allows a Seller selection.
- **SellerDetailsInput** - Returns a read-only user input that, based on selection in SellerInput, displays additional Seller information provided in the SC\_HeaderSellerDetailsMapping Company Parameter table.

## SC\_Forecast\_Configurator

This is the configurator for the Forecast input on the header level. It is used to display forecast inputs for the user.

See also the [architecture documentation](#).

### Elements Description

- **ForecastMethod** - Generates the Forecast Method options selection input that allows the user to provide forecast method.
- **SalesGoalIncreasePercent** - Presents sales goal increase % value user input.

- **PreviousForecastMethod** - Stores the old value of ForecastMethod. It is used to check the current value of ForecastMethod to reset value for SalesGoalIncreasePercent.

## SC\_Compensation

This is the core logic for the whole package. The logic works in four contexts:

1. *Agreement* context (line items) which as a result outputs a set of Compensation Records with some data prefilled.
2. *Generate Compensation Record* context which regenerates records if any of the main source data changed.
3. *Generate Accrual Records* context which constructs the snapshot of the current transaction data and stores them in Accrual Records to be used in calculations later on.
4. *Compensation Record* which does the actual calculation that is run via the Sales Compensation Calculation object on a set schedule.

This logic - due to its size - has "context separators" which make it easier to see which elements run in which context. The separators are empty elements with the following name pattern: `___SOME_NAME_CONTEXT1_CONTEXT2___`

- `SOME_NAME` describes what is in the section.
- `CONTEXT1, CONTEXT2` define the contexts this section runs in.

See also the [architecture documentation](#).

## Elements Description

- **CompensationConfig** - Initialization of a ConfigManager structure defined in SC\_CompensationProcessingLib.
- **CompensationTypeCode** - Retrieves the type name of the currently processed Compensation Line Item. This is the Name column value from ConditionTypes. This code is used to access the appropriate element from the CompensationTypesLib later on.
- **CalculationType** - Defines the calculation type of a currently processed Compensation Type. Can be one of these values: Conditional, Growth, Stepped.
- **QuotaType** - Defines the quota type of a currently processed Compensation Type. Can be one of these values: None, Single, Multi.
- **QuotaValueType** - Defines the target value type of a currently processed Compensation Type. Can be one of these values: Amount, Percent.
- **CompensationValueType** - Defines the compensation value type of a currently processed Compensation Type. Can be one of these values: Amount, Amount Per Unit, Percent, None.
- **DatamartCurrency** - Defines the currency used by the main Datamart. The Datamart configuration is stored in SC\_AdvancedConfiguration.
- **CustomerGroupInput** - User input for Customer Group, used for filtering later on.
- **ProductGroupInput** - User input for Product Group, used for filtering later on.
- **SingleQuotaInput** - Returns a user input for a particular Compensation Type that has the quota defined as Single. Allows input of a quota factor.
- **SingleCompensationInput** - Returns a user input for a particular Compensation Type that has the target defined as Single. Allows input of a compensation value.
- **TargetIncentiveInput** - Returns a user input for a particular Compensation Type that has the calculation type defined as Bonus. Allows input of a target incentive value.
- **VariablePayPercentageInput** - Returns a user input for a particular Compensation Type that has the target defined as Variable Pay or Stepped Variable Pay. Allows input of a variable pay percentage value.

- **MultiQuotalInput** - Returns a user input for a particular Compensation Type that has the quota defined as Multi and Calculation Type as one of Conditional, Growth or Stepped. Allows input of a combination of both quota factor and compensation value in a multitiered fashion.
- **MultiTargetBonusInput** - Returns a user input for a particular Compensation Type that has the quota defined as Multi and Calculation Type as Bonus. Allows input of a combination of both quota factor and bonus payout rate factor in a multitiered fashion.
- **SteppedBonusInput** - Returns a user input for a particular Compensation Type that has the quota defined as Multi and Calculation Type as Stepped Bonus or Stepped Variable Pay. Allows input of a combination of both quota factor and bonus payout rate factor in a multitiered fashion.
- **PaymentPeriodInput** - Displays a used input for a Payment Period selection as an option. Available values are Monthly, Quarterly, Semi-Annually, Annually.
- **QuotaForInput** - Displays a read-only user input containing the QuotaFor value stored in Condition Types table. If the value is not present, the value from Advanced Configuration SC\_AdvancedConfiguration is taken instead.
- **DepositSchemeInput** - Displays a read-only user input containing the DepositScheme value stored in Condition Types table. If the value is not present, the value from Advanced Configuration SC\_AdvancedConfiguration is taken instead.
- **BaselineFieldNameInput** - Displays a read-only user input containing the BaselineFieldName value stored in Condition Types table. If the value is not present, the value from Advanced Configuration SC\_AdvancedConfiguration is taken instead.
- **SyntaxCheckAbort**
- **HeaderSellerGroupInput** - Retrieves the Seller group input value from the header level.
- **HeaderCurrencyInput** - Retrieves the currency value from the header level.
- **HeaderTypeInput** - Retrieves the type of the header.
- **ObjectValidation** - Performs validation of the state of a currently processed item in order to assess whether the logic execution should continue.
- **SellerIdsInput** - Returns the Seller group converted into a list of individual Seller IDs.
- **CompensationInputDefinition** - Returns CompensationInputParameter that stores all the inputs taken from user/header etc. This parameter is used later on for other processing/access operations.
- **InputValidation** - Performs validation on all inputs that were provided. If any input is marked as invalid, an exception is thrown with a proper message to the user.
- **PreviousPeriodAdjustmentData** - Returns the previous period adjustment data used later on in calculations.
- **CurrentPeriodAdjustmentData** - Returns the current period adjustment data used later on in calculations.
- **PreviousPeriodData** - Returns a value of the previous period compensation based on the current context.
- **CurrentPeriodData** - Returns a value of the current period compensation based on the current context.
- **BaselineValueFieldLabel** - Returns a field in a Datamart that will be used for baseline calculations. The field is defined in the Datamart Advanced Options configuration.
- **PreviousBaselineValue** - Extracts the baseline value from the previous period data fetched in one of the previous elements.
- **CurrentBaselineValue** - Extracts the baseline value from the current period data fetched in one of the previous elements.
- **PreviousCompensation** - Extracts the compensation value from the previous period data fetched in one of the previous elements. It colors the result according to the value.
- **CurrentCompensation** - Extracts the baseline value from the current period data fetched in one of the previous elements. It colors the result according to the value.
- **ForecastMethod** - Passes the value of the Forecast Method input.
- **SalesGoalIncreasePct** - Passes the value of Sales Goal Increase % input.
- **ForecastBaselineValue** - Extracts the forecasted baseline value from the current period data fetched in one of the previous elements.

- **ForecastCompensation** - Extracts the forecasted compensation value from the current period data fetched in one of the previous elements.
- **TrueUp** - Forecast Compensation minus Current Compensation.
- **QuotaFor** - Retrieves the quota for a value stored in the configuration of a currently processed Compensation Type of Advanced Configuration of the package (default). Can be either Payment Period or Annual.
- **DepositScheme** - Retrieves the deposit scheme value stored in the configuration of a currently processed Compensation Type of Advanced Configuration of the package (default). Can be either Cumulative or Non-Cumulative.
- **PaymentPeriod** - Passes the value of the Payment Period input.
- **SellerNamesTable** - Passes the value of the Seller input, stores the name of the Sellers.
- **CustomerGroup** - Passes the value of the CustomerGroup input.
- **ProductGroup** - Passes the value of the ProductGroup input.
- **GeneralFilter** - Passes the value of the GeneralFilter input.
- **PayoutCurrency** - Passes the value of the Payout Currency input.
- **CurrentPeriods** - Returns a list of periods based on the line item payment period input and CO plan startDate/endDate.
- **PreviousPeriods** - Returns a list of previous periods based on the line item payment period input and CO plan startDate/endDate.
- **PreviousCompensationDataOnPeriods** - Returns data (adjustment data & Datamart data) of the previous periods. The period is based on the payment period input, used later on in calculations.
- **CurrentCompensationDataOnPeriods** - Returns data (adjustment data & Datamart data) of the current periods. The period is based on the payment period input, used later on in calculations.
- **PreviousMonthlyPeriodsData** - Returns data (adjustment data & Datamart data) of the previous monthly periods, used to aggregate data in the header.
- **CurrentMonthlyPeriodsData** - Returns data (adjustment data & Datamart data) of the current monthly periods, used to aggregate data in the header.
- **AddingMonthlyPeriodsDataToCache** - Adds previous monthly periods data and current monthly periods data to the cache, used to pass data to the header (post phase) for aggregation header data.
- **ForecastCompensationComparisonChart** - Bar chart displaying Previous Compensation vs. Current Compensation vs. Forecasted Compensation.
- **ForecastBaselineValueTable** - Table displaying this year's sales forecast by Payment Period of the particular line item.
- **ForecastBaselineValueChart** - Bar chart displaying this year's sales forecast by Payment Period of the particular line item.
- **PreviousPeriodAdjustmentIds** - Returns the list of IDs of Adjustments that were taken into account during calculations. These Adjustments are taken from the previous period.
- **CurrentPeriodAdjustmentIds** - Returns the list of IDs of Adjustments that were taken into account during calculations. These Adjustments are taken from the current period.
- **SourceCurrencyExchangeRate** - Retrieves the currency exchange rate based on the selected Payout Currency and the Datamart Currency. The Exchange rate is based on the payout date of the record.
- **SourceCurrencyPreviousBaselineValue** - Extracts the baseline value from the previous period data fetched in one of the previous elements, but in source currency.
- **SourceCurrencyCurrentBaselineValue** - Extracts the baseline value from the current period data fetched in one of the previous elements, but in source currency.
- **SourceCurrencyPreviousCompensation** - Extracts the compensation value from the previous period data fetched in one of the previous elements, but in source currency. It colors the result according to the value.
- **SourceCurrencyCurrentCompensation** - Extracts the baseline value from the current period data fetched in one of the previous elements, but in source currency. It colors the result according to the value.

- **SourceCurrencyForecastBaselineValue** - Extracts the forecasted baseline value from the current period data fetched in one of the previous elements, but in source currency.
- **SourceCurrencyForecastCompensation** - Extracts the forecasted compensation value from the current period data fetched in one of the previous elements, but in source currency.
- **SourceCurrencyTrueUp** - Forecast Compensation minus Current Compensation in source currency.
- **SourceCurrency** - Defines the Source currency of the source data, the same as DatamartCurrency.
- **CompensationRecords** - Generates the Compensation Record shells according to the processed information. This element is only run in the Plan context.
- **RegenerateCompensationRecords** - Re-generates the Compensation Records if any of the re-generation conditions were met. This element is run only in the Generate Compensation Record context.
- **GenerateAccrualRecords** - Generates the main calculation data snapshot in the Accrual Records table. This element is run only in the Generate Accrual Record context.
- **CalculationBase** - Returns the calculationBase for the Compensation Record.

### SC\_Dashboard\_Compensations\_Configurator

This is the configurator logic for the Compensations Dashboard. Its main purpose is to allow selection of different Compensation Agreements based on the selected Seller.

See also the [architecture documentation](#).

#### Elements Description

- **SellerInput** - Presents the user with a Seller input if the user is a Sales Manager (allows to select one of the Sellers out of all who report to that person) or if the user is not a Sales Manager (a text with what Seller is being used for further processing).
- **CompensationInput** - Provides a Compensation Agreement selection option based on the Seller input defined above. The agreements are fetched in YTD period and only approved ones are considered.

### SC\_Dashboard\_Compensations

This is a logic for the Compensation Dashboard. It handles display of all inputs, validation of the inputs, data fetching as well as processing of the obtained data. The dashboard displays three portlets: one data table and two charts.

See also the [architecture documentation](#).

#### Elements Description

- **Configuration** - Fetches the Advanced Configuration options for the package. The most important part for the dashboard is the Datamart configuration.
- **DashboardUtils** - Contains all methods related to creation of portlets and data processing related to them. The utils are for all three currently visible portlets.
- **StartDate** - Stores the start date of the current year for further processing.
- **EndDate** - Stores the end date of the current month for further processing.
- **Configurator** - Displays the inline configurator defined by SC\_Dashboard\_Compensations\_Configurator logic.
- **TimePeriod** - Allows selection of time period that modifies the results returned by the data fetch.
- **AbortIfIsSyntaxCheck**
- **InputValidation** - Validates any user input. If it is incorrect, an exception is thrown. Currently only Seller input is validated.
- **PlanCurrency** - Returns the currency in which the currently processed plan is being payed out.

- **CompensationAgreementData** - Main data fetch element for the dashboard, it retrieves Datamart data based on the provided filters such as time period, Seller etc. The data is returned in structures defined by the PaybackAgreementUtils element in the SC\_CompensationDashboardsLib library.
- **CompensationAgreementTable** - Performs processing operations on the data returned from the eCompensationAgreementData element in order to display the defined data table portlet. The data table rows/columns are created in this element.
- **CompensationAgreementByUniqueNameChart** - Gathers information for the display and displays the CompensationAgreementByUniqueNameChart portlet.
- **CompensationAgreementByTimePeriodChart** - Gathers information for the display and displays the CompensationAgreementByTimePeriodChart portlet.
- **MixpanelTracking** - Gathers information about the usage of the dashboard for Mixpanel.

## SC\_Dashboard\_Transactions\_Configurator

This is the configurator logic for the Transaction Dashboard. Its main purpose is to allow filtering the transaction list present in the Dashboard, currently the supported inputs are: SellerGroup, Plan Currency, Compensation Plan Type, Compensation Plan Status, Compensation Plan, Compensation Record, Start Date, End Date.

See also the [architecture documentation](#).

### Elements Description

- **ConfiguratorCommonUtils** - Provides util methods to read values from configurator entries or create them.
- **WarningElement** - Dummy element returning an empty ConfiguratorEntry, used to set warning messages in case of no provided inputs.
- **IsInitialRenderElement** - Creates a Hidden entry that defines whether this or any other configurator refresh is initial.
- **IsInitialRenderValue** - Reads the value that states whether this pass is an initial render, used to set up default values.
- **SellerGroupInput** - Generates the Seller Group picker input. The input is read only for Sales Agents.
- **ShouldAbortCalculation** - Creates a Hidden entry that is used to pass information to dashboard logic whether the calculation of the dashboard should be aborted.
- **AbortInputGeneration** - Aborts the generation of inputs if the condition to display them is not fulfilled.
- **PlanCurrencyInput** - Generates the Plan Payout Currency option input. This input is influenced by all inputs above.
- **CompensationPlanTypeInput** - Generates the Plan Type options input. This input is influenced by all inputs above.
- **CompensationPlanStatusInput** - Generates the Plan Status options input. This input is influenced by all inputs above.
- **CompensationPlanInput** - Generates the Plan Unique Name options input. This input is influenced by all inputs above.
- **CompensationRecordInput** - Generates the Compensation Record Unique Name options input. This input is influenced by all inputs above.
- **StartDateInput** - Generates the Start Date date entry input.
- **EndDateInput** - Generates the End Date date entry input.
- **HiddenWarningInputCheck** - Checks if any of the inputs is populated and if not, sets the warning message on the dummy WarningElement result.
- **IsInitialRenderSetElement** - Sets the value of IsInitialRenderElement to true after all the initial configurator render passes finished.

## SC\_Dashboard\_Transaction

This is a logic for the Transaction Dashboard. It handles display of all inputs, validation of the inputs, data fetching as well as processing of the obtained data. The dashboard displays three portlets: one summary and two tables.

See also the [architecture documentation](#).

### Elements Description

- **Configuration** - Fetches the Advanced Configuration options for the package. The most important part for the dashboard is the Datamart configuration.
- **ConfiguratorUtils** - Contains methods that are used to fetch data from the dashboard input configurator.
- **CompensationAgreementDataUtils** - Contains methods that are used for processing of data related to Compensation Plan, such as finding plans with appropriate filters.
- **TransactionUtils** - Contains methods related to transaction data, queries and processing. This is the element where the query generation happens.
- **DataTableUtils** - Contains methods related to ResultMatrix and the cell management, such as creating link cells.
- **Configurator** - Displays the inline configurator defined by SC\_Dashboard\_Transactions\_Configurator logic.
- **SyntaxCheckAbort**
- **CalculationAbort** - Safety element that prevents calculation of any other elements when certain conditions are fulfilled. The purpose is to prevent loading a dashboard with a lot of data.
- **SourceCurrency** - Returns the Datamart currency for the Datamart defined in Advanced Configuration.
- **CompensationPlanStatistics** - Displays an HTML portlet that provides the summary of the data displayed on the dashboard, such as Number of Transactions or Sellers.
- **TransactionsTable** - Displays a ResultMatrix table portlet that contains the list of transactions that are matching the provided filters.
- **CompensationRecordsTable** - Displays a ResultMatrix table portlet that contains the list of Compensation Records that are matching the provided filters.
- **MixpanelTracking** - Gathers information about the usage of the dashboard for Mixpanel.

## SC\_Dashboard\_Date\_Configurator

This is the configurator logic for the Admin Dashboard and Payouts and Plans Dashboard. Its main purpose is to allow filtering in the Dashboard, currently the supported inputs are: SellerGroup, Reporting Currency, Plan Currency, Payout Year, Payout Date.

See also the [architecture documentation](#).

### Elements Description

- **ConfiguratorUtils** - Contains utility methods that are used during various ConfiguratorEntry operations such as finding valid values for entries
- **SellerGroupInput** - Generates the Seller Group picker input. The input is read only for Sales Agents.
- **ShouldAbortCalculation** - Creates a Hidden entry that is used to pass information to dashboard logic whether the calculation of the dashboard should be aborted.
- **AbortInputGeneration** - Aborts the generation of inputs if the condition to display them is not fulfilled.
- **SellerIdsInput** - Hidden entry that translates the selected SellerGroup values into SellerIds.
- **ReportingCurrencyInput** - Creates a radio input that allows the selection in which currency the dashboard should report: Source currency or a Payout Currency.

- **PlanCurrencyInput** - If Payout Currency is selected in ReportingCurrencyInput this additional input allows to specify which one should be used.
- **PayoutYear** - Generates and input that allows selection of payout year.
- **PayoutDate** - Generates and input that allows selection of payout date.

## SC\_Dashboard\_Admin

This is a logic for the Admin Dashboard. It handles display of all inputs, validation of the inputs, data fetching as well as processing of the obtained data. The dashboard displays two portlets: one summary and one table.

See also the [architecture documentation](#).

### Elements Description

- **Configuration** - Fetches the Advanced Configuration options for the package. The most important part for the dashboard is the Datamart configuration.
- **PayoutDashboardUtils** - Contains methods that are used to fetch data from the dashboard input configurator.
- **Configurator** - Displays the inline configurator defined by SC\_Dashboard\_Transactions\_Configurator logic.
- **SyntaxCheckAbort**
- **InputValidation** - Performs the validation of any entered inputs, currently the checks are if the user has any of the SC groups assigned and if the Payout Date is selected.
- **PayoutCurrency** - Returns the currency in which the currently processed data is being payed out.
- **SelectedDate** - Returns the value of the selected payout date from the Configurator.
- **SelectedSellers** - Returns the value of the selected seller ids from the Configurator.
- **PayoutData** - Returns the filtered data used by the dashboard, considers Seller Ids, Payout Date and Payout Currency as main data filters. Uses Datamart configuration.
- **PayoutTable** - Prepares the Result Matrix table that displays the Payouts.
- **PayoutSummary** - Prepares the summary of the Payouts. Contains Compensations YTD and all Current Payments.
- **PayeePlansDashboard** - Currently unused.
- **MixpanelTracking** - Gathers information about the usage of the dashboard for Mixpanel.

## SC\_Dashboard\_PayeePlans

This is a logic for the Payee Plans Dashboard. It handles display of all inputs, validation of the inputs, data fetching as well as processing of the obtained data. The dashboard displays two portlets: one summary and two tables.

See also the [architecture documentation](#).

### Elements Description

- **DashboardUtils** - Contains methods that are used to fetch data for the dashboard as well as process and prepare it for the use in portlets.
- **Configuration** - Fetches the Advanced Configuration options for the package. The most important part for the dashboard is the Datamart configuration.
- **Configurator** - Displays the inline configurator defined by SC\_Dashboard\_Transactions\_Configurator logic.
- **SyntaxCheckAbort**
- **InputValidation** - Performs the validation of any entered inputs, currently the checks are if the user has any of the SC groups assigned and if the Payout Date is selected.
- **SelectedSellers** - Returns the value of the selected seller ids from the Configurator.

- **SelectedPayoutYear** - Returns the value of the selected payout year from the Configurator.
- **SelectedPayoutDate** - Returns the value of the selected payout date from the Configurator.
- **PayoutCurrency** - Returns the currency in which the currently processed data is being payed out.
- **ActiveCompensationData** - Returns the found Active Compensation Plans data based on the provided filters. To be used in ActiveCompensationsTable portlet.
- **ActiveCompensationsTable** - Displays the Active Compensations Table Result Matrix portlet.
- **CompensationDetailsDashboard** - Opens the SC\_EmbeddedCompensationDetailDashboard which provides additional details related to the currently selected plan. This dashboard opens on click on a row in ActiveCompensationsTable portlet.
- **PayoutData** - Returns the filtered data used by the dashboard, considers Seller Ids, Payout Date and Payout Currency as main data filters. Uses Datamart configuration.
- **PayoutTable** - Prepares the Result Matrix table that displays the Payouts.
- **TransactionDashboard** - Currently not used.
- **PayoutSummary** - Prepares the summary of the Payouts. Contains information about how many sellers are being processed, the number of compensation plans and number of payouts.
- **MixpanelTracking** - Gathers information about the usage of the dashboard for Mixpanel.

### SC\_EmbeddedCompensationDetailDashboard

This is a logic for the embedded dashboard that is used in Payee Plans Dashboard. The dashboard displays additional information about any selected plan.

See also the [architecture documentation](#).

#### Elements Description

- **ConfigManager** - Returns the instance of ConfigManager, which is a structure containing various configuration related information.
- **DashboardUtils** - Contains methods that are used to fetch and process all the data required by the dashboard.
- **AbortSyntaxCheck**
- **CompensationDetailTable** - Displays the details table based on the plan that has been selected in the Payee Plans Dashboard.

### SC\_Compensation\_RecordCalculationFeeder

This is a feeder logic used for fetching and providing items for scheduled compensation calculation executions.

See also the [architecture documentation](#).

#### Elements Description

- **SyntaxCheckAbort**
- **EmittingCompensationRecord** - Main element that emits the Compensation Records based on the provided filters.

### SC\_AdjustmentHeader

This is a header logic for the Adjustment Custom Form object. It is used to read the properties of the object and pass them to the header configurator.

See also the [architecture documentation](#).

## Elements Description

- **Configurator** - Initializes the header configurator in the customFormProcessor and passes appropriate parameters to it.

## SC\_AdjustmentHeaderConfigurator

This is the main Adjustment object configurator that is used to generate adjustments based on user provided inputs.

See also the [architecture documentation](#).

## Elements Description

- **DatamartName** - Returns the Datamart name stored in Advanced Configuration.
- **SellerData** - Returns all Sellers (with labels) stored in the system.
- **AdjustmentFormId** - Retrieves the adjustment ID value passed from the header logic.
- **CreatedByInput** - Returns a read-only text input that displays the Created By value.
- **CreatedOnInput** - Returns a read-only text input that displays the Created On value.
- **TitleInput** - Returns a read-only text input that displays the Adjustment title.
- **StatusInput** - Returns a read-only text input that displays the current Adjustment status.
- **DescriptionInput** - Returns a text input that allows the user to insert a description for the Adjustment.
- **ProductGroupInput** - Returns a Product Group selection input used to filter out transactions affected by the Adjustment.
- **CustomerGroupInput** - Returns a Customer Group selection input used to filter out transactions affected by the Adjustment.
- **SellerInput** - Returns a Seller selection input used to filter out transactions affected by the Adjustment.
- **PaymentDateInput** - Returns a Payment Date selection input used to filter out transactions affected by the Adjustment.
- **GeneralFilterInput** - Returns a General Filter input used to filter out transactions affected by the Adjustment.
- **TransactionsDataMatrix** - Displays the transaction matrix that shows transactions that will be affected by the adjustment. The display limit is set to show top 10 transactions, the rest is hidden due to performance reasons.
- **AdjustmentTypeInput** - Returns an option input that allows the selection of the Adjustment type that will define how the adjustment will be generated and handled.
- **NewSellerInput** - In case of "Change Data" Adjustment type allows to insert a new Seller value for the selected transactions.
- **NewProductInput** - In case of "Change Data" Adjustment type allows to insert a new Product value for the selected transactions.
- **NewCustomerInput** - In case of "Change Data" Adjustment type allows to insert a new Customer value for the selected transactions.
- **NewDateInput** - In case of "Change Data" Adjustment type allows to insert a new Pricing Date value for the selected transactions.
- **BaselineInputsInfo** - In case of "Change Data" Adjustment type displays some additional information for the user about baseline fields.
- **NewInvoicePriceInput** - In case of "Change Data" Adjustment type allows to insert a new Invoice Price value for the selected transactions.
- **NewQuantityInput** - In case of "Change Data" Adjustment type allows to insert a new Quantity value for the selected transactions.
- **NewGrossMarginInput** - In case of "Change Data" Adjustment type allows to insert a new Gross Margin value for the selected transactions.

- **NewSellersCreditSplit** - In case of "Split Credit" Adjustment type displays an Input Matrix that allows the user to define the Seller credit split. If the split percentage does not equal to 100%, an error message is shown.

### SC\_AdjustmentWizardExecutor

This is an executor logic for the Adjustment Configurator Wizard, used to create adjustment objects.

See also the [architecture documentation](#).

#### Elements Description

- **CreateAdjustmentCFO** -- Generates the adjustment object based on the provided inputs and stores the appropriate values in attributes.
- **SummaryScreen** -- Displays a summary message informing the user what happened after submission of the Adjustment.

### SC\_AdjustmentWizardInputs

This is an input logic for the Adjustment Configurator Wizard, used to display inputs for the user.

See also the [architecture documentation](#).

#### Elements Description

- **SellerData** -- Returns all Sellers (with labels) stored in the system.
- **TitleInput** -- Generates the text input that allows the user to insert the title of the adjustment.
- **DescriptionInput** - Generates the text input that allows the user to insert the description of the adjustment.
- **ProductGroupInput** -- Generates the Product Group input that allows the user to provide transactions for the Adjustment.
- **CustomerGroupInput** -- Generates the Customer Group input that allows the user to provide transactions for the Adjustment.
- **SellerInput** -- Generates the Seller options selection input that allows the user to provide transactions for the Adjustment.
- **PaymentDateInput** -- Generates the Date input that allows the user to provide transactions for the Adjustment.
- **GeneralFilterInput** -- Generates the General Filter input that allows the user to provide transactions for the Adjustment.

### SC\_QuotingPluginLib

This is a library that is used for integration with quoting solutions. It allows you to inject the Sales Compensation into the calculation results. The library also supports displaying the results in a pie chart or line score chart.

See also the [architecture documentation](#).

#### Elements Description

- **ConstConfig** -- Contains definitions of global constant variables that are used throughout the logic. Any configuration-like constants should be stored here to avoid hard-coding and to allow for better maintainability of the solution
- **ConfigManager** - Strictly defined structure that allows management of solution configuration. The benefit is that the whole configuration is contained within one "class", it cannot be modified and is cached.

- **SalesCompensationEstimation** - Stores methods used to calculate the estimated compensation based on items stored in a quote.
- **ChartUtils** - Contains methods used to generate the charts supported by the solution.

## SC\_OverviewReport

This is a logic for the Overview Report. The logic is strictly tied to a publishing template present also in the repository. The element names that are used for data display need to match 1-1 in order to display properly.

See also the [architecture documentation](#).

### Elements Description

- **CompensationAgreementData** - Returns the data of the currently open Compensation Agreement.
- **AgreementName** - Display element. Returns the agreement name.
- **Label** - Display element. Returns the agreement label.
- **StartDate** - Display element. Returns the agreement start date.
- **EndDate** - Display element. Returns the agreement end date.
- **CreatedByPrimary** - Display element. Returns the Created By primary part of the Created By name. The primary part contains either name and surname of the Seller if defined or the system username if the Seller is not found.
- **CreatedBySecondary** - Display element. Returns the secondary part of the Created By name. The secondary part is displayed only if the name and surname of the Seller are known and contain a system username in brackets.
- **CreatedDate** - Display element. Returns the creation date of the agreement.
- **ApprovedByPrimary** - Display element. Returns the Created By primary part of the Approved By name. The primary part contains either name and surname of the Seller if defined or the system username if the Seller is not found.
- **ApprovedBySecondary** - Display element. Returns the secondary part of the Approved By name. The secondary part is displayed only if the name and surname of the Seller are known and contain a system username in brackets.
- **PrimarySellerName** - Display element. Returns the Created By primary part of the Seller name. The primary part contains either name and surname of the Seller if defined or the system username if the Seller is not found.
- **SecondarySellerName** - Display element. Returns the secondary part of the Seller name. The secondary part is displayed only if the name and surname of the Seller are known and contain a system username in brackets.
- **Currency** - Display element. Returns the currency of the agreement.
- **GeneralFilter** - Display element. Returns the value of the general filter input that is present on the agreement.
- **CompensationItems** - Returns a list of all line items present in the agreement. The line items have to be returned as a list of maps where each map reflects one line item value, the structure is clearly defined and has to match 1-1 what is in the publishing template.
- **TotalCompensation** - Display element. Returns the sum of total compensations of all agreement line items.

## SC\_SalesCompensationPlanReport

This is a logic for the Sales Compensation Agreement Report. The logic is strictly tied to a publishing template present also in the repository. The element names that are used for data display need to match 1-1 in order to display properly.

See also the [architecture documentation](#).

## Elements Description

- **CompensationAgreementData** - Returns the data of the currently open Compensation Agreement.
- **StartDate** - Display element. Returns the agreement start date.
- **EndDate** - Display element. Returns the agreement end date.
- **PrimarySellerName** - Display element. Returns the Created By primary part of the Seller name. The primary part contains either name and surname of the Seller if defined or the system username if the Seller is not found.
- **SecondarySellerName** - Display element. Returns the secondary part of the Seller name. The secondary part is displayed only if the name and surname of the Seller are known and contain a system username in brackets.

## SC\_YTDReport

This is a logic for the YTD Report. The logic is strictly tied to a publishing template present also in the repository. The element names that are used for data display need to match 1-1 in order to display properly.

See also the [architecture documentation](#).

## Elements Description

- **CompensationAgreementData** - Returns the data of the currently open Compensation Agreement.
- **AgreementName** - Display element. Returns the agreement name.
- **Label** - Display element. Returns the agreement label.
- **ReportStartDate** - Stores the start date of the current year for further processing.
- **ReportEndDate** - Stores the end date of the current month for further processing.
- **ReportCurrentDate** - Stores the current date for further processing.
- **CreatedByPrimary** - Display element. Returns the Created By primary part of the Created By name. The primary part contains either name and surname of the Seller if defined or the system username if the Seller is not found.
- **CreatedBySecondary** - Display element. Returns the secondary part of the Created By name. The secondary part is displayed only if the name and surname of the Seller are known and contain a system username in brackets.
- **ApprovedByPrimary** - Display element. Returns the Created By primary part of the Approved By name. The primary part contains either name and surname of the Seller if defined or the system username if the Seller is not found.
- **ApprovedBySecondary** - Display element. Returns the secondary part of the Approved By name. The secondary part is displayed only if the name and surname of the Seller are known and contain a system username in brackets.
- **PrimarySellerName** - Display element. Returns the Created By primary part of the Seller name. The primary part contains either name and surname of the Seller if defined or the system username if the Seller is not found.
- **SecondarySellerName** - Display element. Returns the secondary part of the Seller name. The secondary part is displayed only if the name and surname of the Seller are known and contain a system username in brackets.
- **Currency** - Display element. Returns the currency of the agreement.
- **GeneralFilter** - Display element. Returns the value of the general filter input that is present on the agreement.
- **CompensationItems** - Returns a list of all line items present in the agreement. The line items have to be returned as a list of maps where each map reflects one line item value, the structure is clearly defined and has to match 1-1 what is in the publishing template.

- **TotalCompensation** - Display element. Returns the sum of total compensations of all agreement line items.

## SC\_TransactionsReport

This is a logic for the YTD Report. The logic is strictly tied to a publishing template present also in the repository. The element names that are used for data display need to match 1-1 in order to display properly.

See also the [architecture documentation](#).

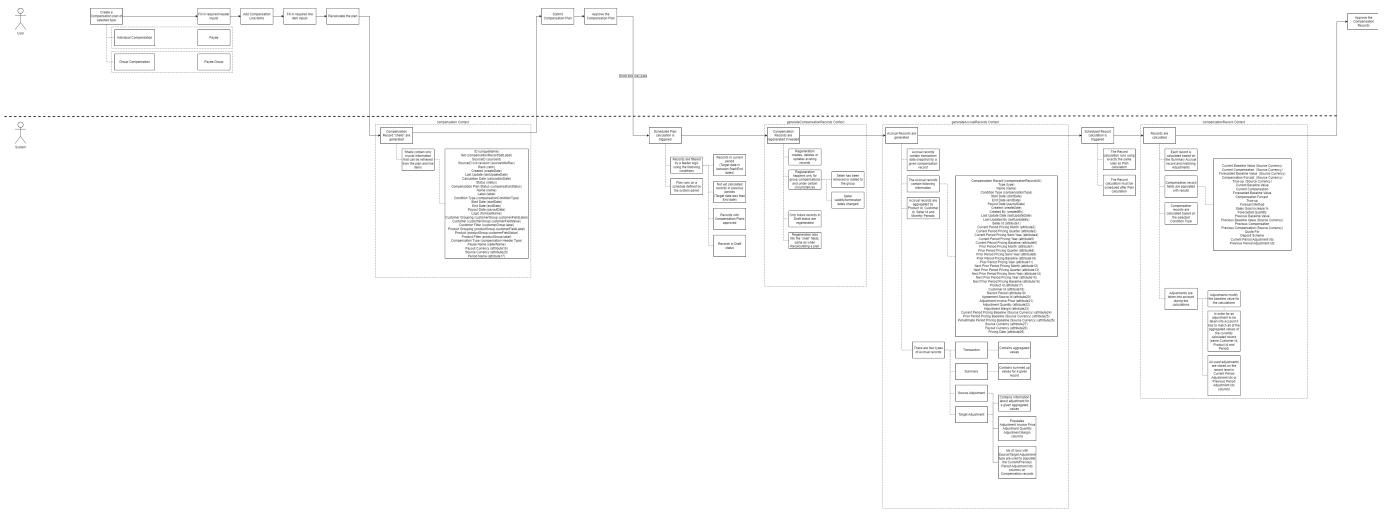
### Elements Description

- **ConfigManager** - Provides an instance of ConfigManager of the package.
- **CompensationAgreementData** - Returns data of the currently open Compensation Agreement.
- **AgreementName** - Display element. Returns the agreement name.
- **Label** - Display element. Returns the agreement label.
- **CreatedByPrimary** - Display element. Returns the Created By primary part of the Created By name. The primary part contains either name and surname of the Seller if defined or the system username if the Seller is not found.
- **CreatedBySecondary** - Display element. Returns the secondary part of the Created By name. The secondary part is displayed only if the name and surname of the Seller are known and contain a system username in brackets.
- **CreatedDate** - Display element. Returns the creation date of the agreement.
- **ApprovedByPrimary** - Display element. Returns the Created By primary part of the Approved By name. The primary part contains either name and surname of the Seller if defined or the system username if the Seller is not found.
- **ApprovedBySecondary** - Display element. Returns the secondary part of the Approved By name. The secondary part is displayed only if the name and surname of the Seller are known and contain a system username in brackets.
- **TransactionItems** - Returns a list of all transactions that were considered when calculating the values in the agreement. The transaction items have to be returned as a list of maps where each map reflects one item, the structure is clearly defined and has to match 1-1 what is in the publishing template.
- **TotalTransactions** - Display element. Returns the count of all transactions.

## Flow Diagrams (SC)

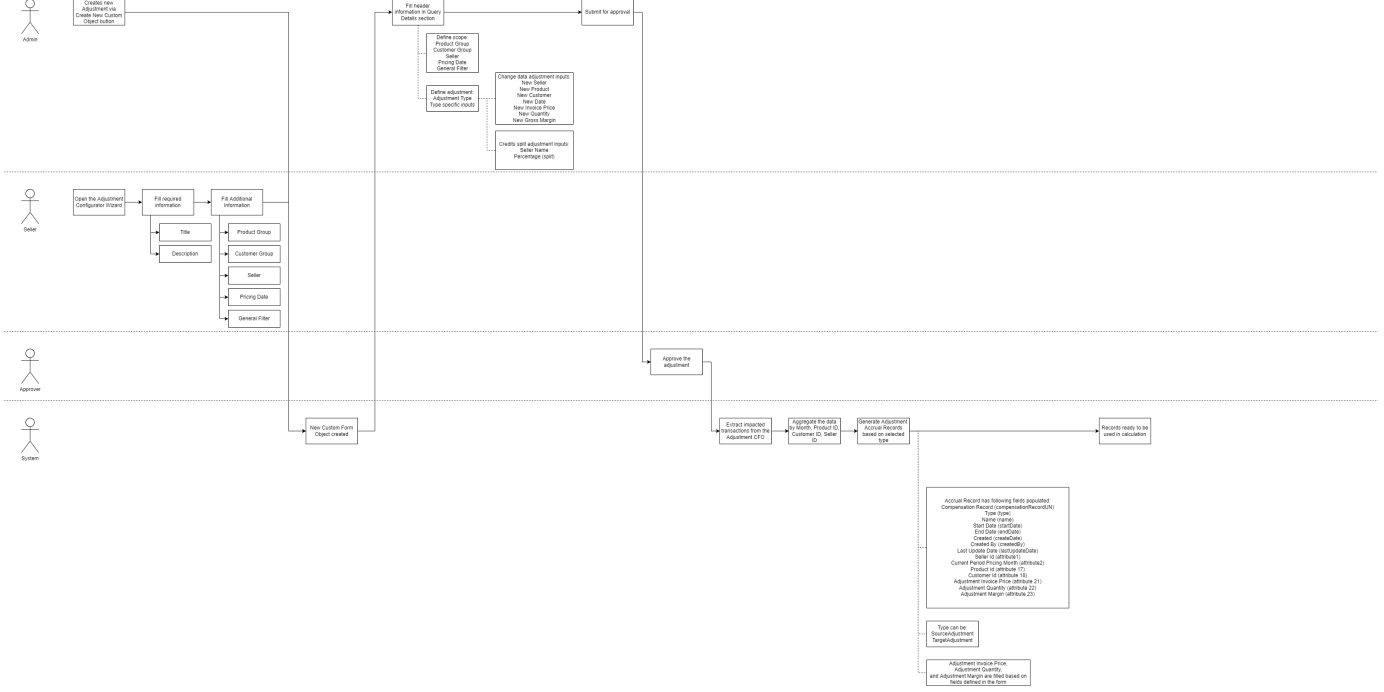
### Calculation Flow

The following diagram represents the main calculation flow of the solution, starting from the creation of a plan, through generation of accrual records, adjustments and ending with the calculation results. This is the core of the solution.



### Adjustments Flow

The following diagram represents the flow of adjustments, how they are created, what impact they have and what the result of an adjustment creation is.




## Manual Upgrade Steps (SC)

In this section, you will find all manual steps which need to be done while upgrading between versions. A description of the whole upgrade procedure is described in the [Upgrade \(SC\)](#) section. In addition, you can also check out the latest release notes.

1.0.0 1.1.0

We do not support upgrading between these versions. 1.1.0 uses the Sales Compensations module instead of the Rebates module, so it needs to be deployed from scratch.

# Sales Compensation 1.5.0

 Version 1.5.0 requires Pricefx Hurricane 9.4.0 to work properly.

## Changes to Business Roles

In this release we have made some changes to the Business Roles included in the package. These are the differences:

Business Role	Removed	Kept	Added
Sales Agent	<ul style="list-style-type: none"> <li>View Compensation Plans</li> <li>View Compensation Records</li> </ul>	<ul style="list-style-type: none"> <li>View Customers</li> <li>View Products</li> <li>Export Dashboards</li> <li>View Dashboards</li> </ul>	<ul style="list-style-type: none"> <li>View Custom Form</li> <li>Run Analytics Queries</li> <li>View Product Details</li> </ul>
Sales Manager	<ul style="list-style-type: none"> <li>View Custom Form</li> </ul>	<ul style="list-style-type: none"> <li>View Customers</li> <li>View Products</li> <li>Manage Compensation Plans</li> <li>Manage Compensation Records</li> <li>Export Dashboards</li> <li>View Dashboards</li> </ul>	<ul style="list-style-type: none"> <li>Manage Custom Form</li> <li>View Seller Extensions</li> <li>View Sellers</li> <li>Run Analytics Queries</li> <li>View Product Details</li> </ul>
Sales Administrator	<ul style="list-style-type: none"> <li>Administer Company Parameters</li> <li>Administer Custom Form</li> <li>Administer Sellers</li> <li>Edit Localization</li> <li>Manage Dashboards</li> <li>Administer Schemas</li> <li>Manage Data Analyzer</li> <li>Manage Data Manager</li> <li>Manage Workflow Logics</li> </ul>	<ul style="list-style-type: none"> <li>View Customers</li> <li>View Products</li> <li>Administer Compensation Plans</li> <li>Administer Compensation Records</li> <li>Administer Sales Compensation module</li> <li>Export Dashboards</li> <li>View Dashboards</li> <li>Manage Workflows</li> </ul>	<ul style="list-style-type: none"> <li>Manage Custom Form</li> <li>Manage Seller Extensions</li> <li>Manage Sellers</li> <li>Edit Global Preferences</li> <li>Run Analytics Queries</li> <li>View Product Details</li> </ul>

This means that the Business Roles have now the following permissions:

### Sales Agent

- View Dashboards
- Export Dashboards
- View Custom Form
- View Customers
- View Products
- View Product Details
- Run Analytics Queries

### Sales Manager

- View Dashboards
- Export Dashboards
- View Sellers
- View Seller Extensions
- Manage Compensation Plans
- Manage Compensation Records
- Manage Custom Form
- View Customers
- View Products
- View Product Details
- Run Analytics Queries

### **Sales Admin**

- View Dashboards
- Export Dashboards
- Manage Sellers
- Manage Seller Extensions
- Manage Custom Form
- Administer Compensation Plans
- Administer Compensation Records
- Administer Sales Compensations Module
- Edit Global Preferences
- Manage Workflows
- View Customers
- View Products
- View Product Details
- Run Analytics Queries

### Currency Support

The currency support has been added. The plans can now be payed out in different currencies. This requires one manual change listed in Upgrade Notes below.

### Upgrade Notes

#### **Add AttributeExtension on Compensation Plan Screen**

Due to the Currency support there is a need to add an attribute extension on the Compensation Plan screen. The attribute extension should have the following structure:

- Name: currency
- Label: Payout Currency
- Type: String

#### **Update Business Roles**

The Business Roles should be updated according to the tables above.

#### **Add Hidden User Group**

A new Hidden User Group needs to be added in User Group Admin:

- Name: SC\_Hidden
- Label: [SC] Hidden

The group is never assigned to anyone.

### Add Default Compensation Type Filter Formula

Add the SC\_DefaultConditionTypeFilter into AdvancedConfiguration /defaultCompensationTypeFilterFormula. This change is due to [PFPCS-6120](#).

### Update CompensationAccrualRecordsAttributes

There are new attributes present in the Accrual Records:

- Attribute24
  - Name: SourceCurrencyCurrentPeriodPricingBaseline
  - Label: Current Period Pricing Baseline (Source Currency)
  - Type: Real
  - Format Type: Money
  - Description: This value typically represents volume, revenue, or margin depending on the relevant condition type.
- Attribute25
  - Name: SourceCurrencyPriorPeriodPricingBaseline
  - Label: Prior Period Pricing Baseline (Source Currency)
  - Type: Real
  - Format Type: Money
  - Description: This number represents a value from one period ago. It typically represents volume, revenue, or margin depending on the relevant condition type.
- Attribute26
  - Name: SourceCurrencyPenultimatePeriodPricingBaseline
  - Label: Penultimate Period Pricing Baseline (Source Currency)
  - Type: Real
  - Format Type: Money
  - Description: This number represents a value from two periods ago. It typically represents volume, revenue, or margin depending on the relevant condition type.
- Attribute27
  - Name: SourceCurrency
  - Label: Source Currency
  - Type: String
- Attribute28
  - Name: PayoutCurrency
  - Label: Payout Currency
  - Type: String
- Attribute29
  - Name: PricingDate
  - Label: Pricing Date
  - Type: String

### Set up Default Compensation Type

Modify the (Default) type in the Compensation Types screen with the following:

- Filter formula: No SC Condition Type Filter
- User Group (View): SC\_Hidden
- User Group (Edit): SC\_Hidden

## Update CompensationRecordAttribute

- Attribute13
  - Name: SourceCurrencyPreviousCompensation
  - Label: Previous Compensation (Source Currency)
  - Type: Real
  - Format Type: Money
- Attribute16
  - Name: PayoutCurrency
  - Label: Payout Currency
  - Type: String
- Attribute20
  - Name: SourceCurrency
  - Label: Source Currency
  - Type: String
- Attribute23
  - Name: SourceCurrencyForecastBaselineValue
  - Label: Forecasted Baseline Value (Source Currency)
  - Type: Real
  - Format Type: Money
  - Description: This value typically represents forecasted volume, revenue, or margin depending on the relevant condition type.
- Attribute24
  - Name: SourceCurrencyForecastCompensation
  - Label: Compensation Forecast (Source Currency)
  - Type: Real
  - Format Type: Money
- Attribute25
  - Name: SourceCurrencyTrueUp
  - Label: True-up (Source Currency)
  - Type: Real
  - Format Type: Money
- Attribute5
  - Name: SourceCurrencyCurrentBaselineValue
  - Label: Current Baseline Value (Source Currency)
  - Type: Real
  - Format Type: Money
  - Description: This value typically represents volume, revenue, or margin depending on the relevant condition type.
- Attribute6
  - Name: SourceCurrencyCurrentCompensation
  - Label: Current Compensation (Source Currency)
  - Type: Real
  - Format Type: Money
- Attribute7
  - Name: SourceCurrencyPreviousBaselineValue

- Label: Previous Baseline Value (Source Currency)
- Type: Real
- Format Type: Money
- Description: This number represents a value from one period ago. It typically represents volume, revenue, or margin depending on the relevant condition type.

### Remove Adjustments Configuration Wizard User Permissions

Previously the Configuration Wizard for Adjustments (SC\_AdjustmentWizard) had restrictions on user groups: SC\_Administrator and SC\_SalesManager, they should be removed now.

### Modify SalesOrg Seller Extension Keys

There should be a modification to the keys in the SalesOrg SX, now the keys are Seller Id, Valid From and Team, as opposed to previous Seller Id, Valid From and Valid To.

### Update SellerSalaries SX

There should be a new attribute added to the SellerSalaries, the attribute is as follows:

- Attribute2
  - Name: Currency
  - Label: Currency
  - Type: String

This attribute should be also added as a new Business Key for this table (besides SellerId).

## Fixed Issues

### Stories

[PFPCS-6003](#) Editable currency

[PFPCS-6331](#) Add Payout Currency support to the Calculation Flow

[PFPCS-6335](#) Admin/Payouts and Plans Dashboards adjustments for Payout Currency

[PFPCS-6336](#) Transaction Dashboard adjustments for Payout Currency

[PFPCS-6337](#) Compensation Dashboard adjustments for Payout Currency

[PFPCS-6342](#) Payouts and Plans, Transaction Dashboard: Show "There's no seller defined for user" message if Pricefx user ID is not associated with any seller Id

[PFPCS-6221](#) Introduce Compensation Plan Status input for Transaction Dashboard

[PFPCS-6320](#) Transaction Dashboard / Compensation Dashboard: Should include Co ID in the dropdown list of Compensation Plan(s) input

[PFPCS-6322](#) Dashboards Improvements: Align dashboard visibility

[PFPCS-6339](#) Hide "Variable Pay" condition types if no all sellers have salary specified

[PFPCS-6359](#) Address changes in core for Sellers User Roles

[PFPCS-6332](#) Add Payout Currency support to the Calculation Flow - Plan Part

- PFPCS-6333 Add Payout Currency support to the Calculation Flow - Compensation Record Part
- PFPCS-6334 Add Payout Currency support to the Calculation Flow - Accrual Record Part
- PFPCS-6355 Update logic to get correct currency value of Accrual record in Compensation record context
- PFPCS-6358 Add Payout Currency support to the Calculation Flow - Adjustment Part

## Bugs

- PFPCS-6120 CO: Error thrown when creating a new Default type
- PFPCS-6216 Adjustment workflow step is not displayed after submission
- PFPCS-6304 COR: Return EndDate incorrectly if seller termination date is changed to another date
- PFPCS-6330 Group: Incorrect previous baseline value on CO because of validFrom/validTo (SX) and startDate/EndDate (CO)
- PFPCS-6346 Variable Pay Multi Target Amount: Showing incorrect Target Incentive (%) input label
- PFPCS-6350 Transaction Dashboard: Showing irrelevant Compensation Plan type although it is not selected on the input
- PFPCS-6352 Transaction Dashboard: Compensation Plan Type(s) input shows No data when having multiple sellers selected
- PFPCS-6362 Transaction Dashboard: Data still displays although the user account is not associated any sellerId
- PFPCS-6395 Payout Currency COR: Error thrown after running record job and creating adjustment data
- PFPCS-6398 Missing Source Currency value in compensation record detail
- PFPCS-6401 Difference between value in DM and after converted from payout currency to source currency in COR
- PFPCS-6402 Adjustment: Caught error in [SC\_AdjustmentsPostApprovalWorkflow - AdjustmentProcessing - DateUtils] : ERROR(@164): Unparseable date: "Thu Sep 08 00:00:00 UTC 2022"
- PFPCS-6406 Missing PA\_Query role for SC business roles
- PFPCS-6407 Payout Currency Dashboard: Currencies are not sorted in alphabetical order
- PFPCS-6408 Missing to add Team in business key of SalesOrg SX
- PFPCS-6409 Error caught in the regeneration compensation records after adding a new seller in a team
- PFPCS-6411 Payouts and Plans Dashboard: Value is not converted based on the enabled Source Currency radio
- PFPCS-6412 Payouts and Plans Dashboard: Still has data displayed in inputs if the user account is not associated any sellerId
- PFPCS-6413 Transaction Dashboard: Still has data displayed in inputs if the user account is not associated any sellerId
- PFPCS-6414 Transaction Dashboard: Compensation Records table shows irrelevant sellers based on the selected Seller input

[PFPCS-6415](#) Transaction Dashboard: No data in tables and portlets if selecting compensation record (payout currency)

[PFPCS-6417](#) Transaction Dashboard: Compensation Records table shows irrelevant COs based on the selected Compensation Plan Status input

[PFPCS-6418](#) Missing View Product Details for SC business roles