SAP® S/4HANA Utilities extensions for meter to cash processes by PROLOGA

Release 2.0 1909

Gas Extensions - Configuration Guide

PROLOGA Energy

15 March 2021

Document History

Before you start the implementation, make sure you have the latest version of this document. You can find the latest version at the following location: http://service.sap.com/instguides -> *SAP Solution Extensions -> S/4HANA Utilities extensions for meter to cash processes -> 2.0 1909*

The following table provides an overview of the most important document changes.

Version	Important Changes
1	Initial version for release 2.0 1909

Table 1: Most important document changes

Table of contents

Table of Figures

Table of Tables

Glossary

Attention

Note

1 Introduction

The functionalities for the *SAP® S/4HANA Utilities extensions for meter to cash processes by PROLOGA* extend the range of functions of *SAP® S/4HANA for Utilities*. In order to guarantee a working configuration, different prerequisites are necessary such as a NetWeaver basic customizing or an *EDM* basic customizing. This configuration guide only treatises the configuration of *SAP® S/4HANA Utilities extensions for meter to cash processes by PROLOGA.*

The following functionalities and their configuration will be described in the next chapters:

- Simple Process Flow a lightweight processing framework to design process workflows
- H-values calculation for the processing of standard load profiles gas
    - In comparison to release 1 of SAP® S/4HANA Utilities extensions for meter to cash processes by PROLOGA the possibility for calculating HTWK-Leipzig-formulas was removed
- Technical quantity determination for performance measured customers
- Cluster Framework – Extended master data checks for IS-U and EDM
- Generic CSV/XML-Importer – Tool for importing CSV and XML data
- Balancing Group Monitor – Overview of quantities structured by balancing information
- Extensions for IS-U & EDM functions
    - Dynamic EDM Settlement – Customizing Settlement procedures without workbench objects
    - Calorific value extensions – billing extensions to determine calorific data from EDM profiles
    - Extension for EDM calculation workbench – automatic execution of the EDM calculation workbench, based on formula ids or formula groups, also additional formulas

2 Configuration Guide

The configuration of the different functionalities of *SAP® S/4HANA Utilities extensions for meter to cash processes by PROLOGA* is explained in detail.

2.1 Simple Process Flow

The *Simple Process Flow* is a light-weight process configuration and monitoring framework. It is designed to trigger process steps (in the following described as *Nodes*) and manage the outcome within the process. Each node is dedicated to a *Lane*, basically an entity to categorize the single nodes which can be used to manage runtime objects for multiple nodes.

Each process can be started creating a process instance with a process date. Currently the framework supports a sequential processing of nodes, which is calculated when a new version of the configuration is saved within the UI5 application.

Instances can be paused and re-triggered when correctly configured within the process configuration.

2.1.1 Authorization checks

Within the *Simple Process Flow* the authorization object, described in /PLGA/EDMP - Authorization object for Simple Process Flow, will be checked on execution of the following reports:

- /PLGA/RP_EDMPR_CREATEPI Create a Processinstance of SPF (transaction /PLGA/SPF_CREATE)
- /PLGA/RP_EDMPR_TRIGGER Trigger Processinstances of SPF

Furthermore, after identifying the correct process identification, the process management class checks for process specific permission.

Until this check, no master data or transactional is altered or written. Triggering a process instance with insufficient permission will not invalidate the transactional data.

2.1.2 Number range object /PLGA/PRNO

The *Simple Process Flow* needs instance numbers to run processes and to save run information of the process and the nodes within the process workflow. This number is generated via the number range object */PLGA/*PRNO. This number range must be set via transaction *SNUM* in the object */PLGA/PRNO*. The number range "01" has to be specified according to the following example:

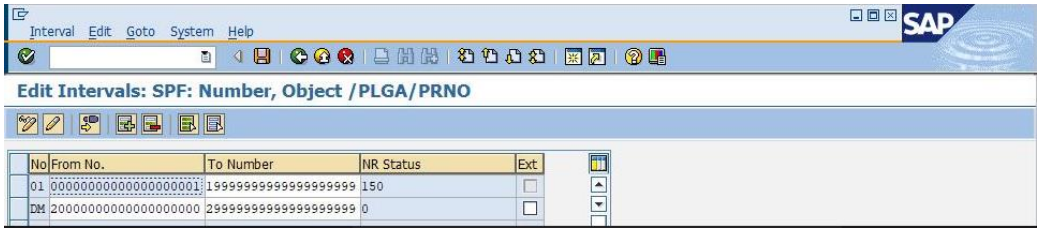


Figure 1: Number range SPF instances

2.1.3 SAP UI5 Configuration

This chapter describes step-by-step an example configuration to successfully creating a Fiori launchpad on the *Fiori Frontend-Server*.

Two applications are shipped within this extension to monitor the process instances of *Simple Process Flow* or configure process versions.

2.1.3.1 Semantic Object

Semantic Objects need to be created, so that Business Tiles and Business Objects can be linked within a "Business Catalog". To create a "Semantic Object" use the transaction code: */UI2/SEMOBJ*.

Within the Transaction select "New Entries" and enter a fitting name, description, and object name for the Semantic Object.

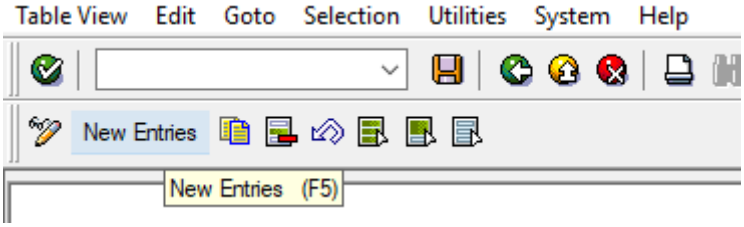


Figure 2: New semantic object

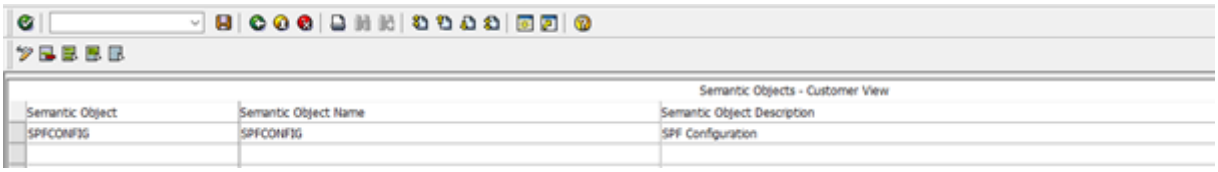


Figure 3: Semantic object configuration

This configuration shall be repeated for each application.

2.1.3.2 Business Catalog

After the configuration is saved a "Business Catalog" has to be defined. To open the *Fiori Launchpad Designer* the transaction */UI2/FLPD_CUST*. In case the transaction is not available the *Launchpad Designer* can also be opened directly in a web browser of your choice. The URL to access the *Launchpad Designer* is composed as followed:

http://<host>:<port>/sap/bc/ui5_ui5/sap/arsrvc_upb_admn/main.html.

To create a new "Business Catalog" select the plus (+) symbol in the launchpad.

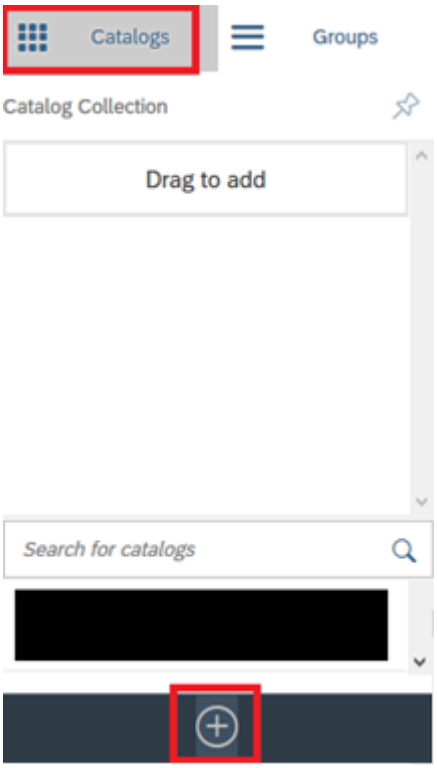


Figure 4: New catalogue

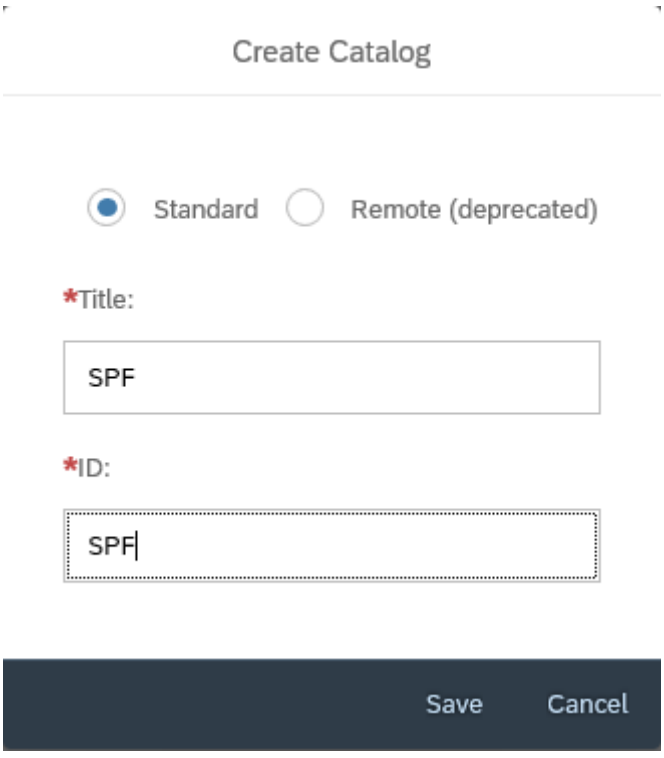


Figure 5: Catalog definition

In the following popup dialog enter a fitting name and ID for the business catalogue.

The following steps add the application to the catalogue. This should be done for each application used in this catalogue.

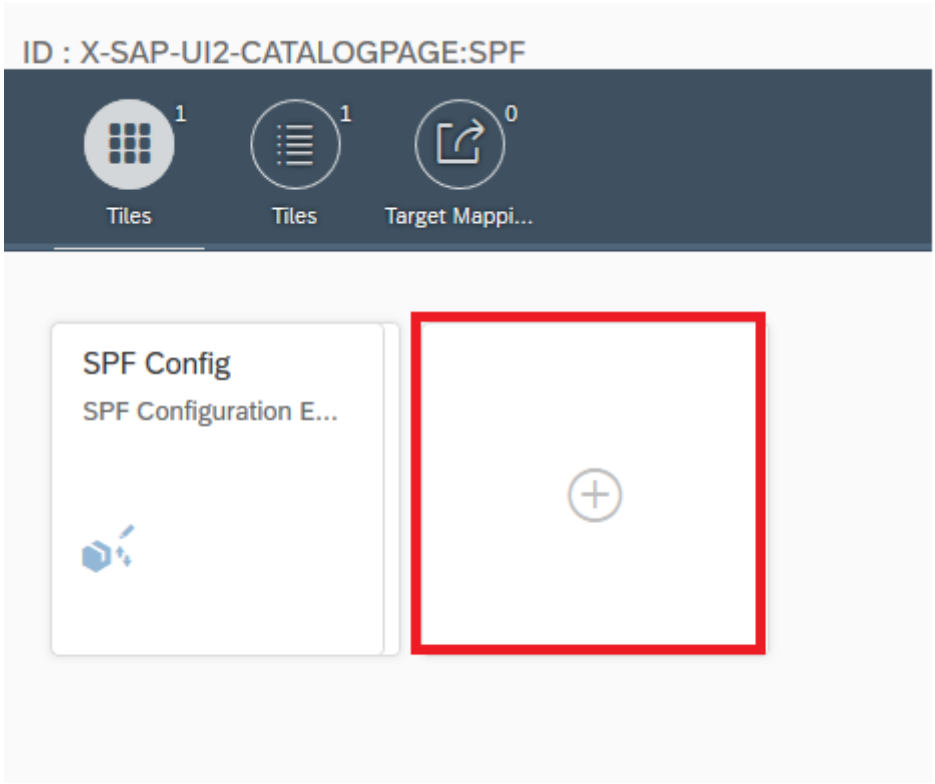


Figure 6: New tile

In the following window click the plus (+) symbol to add a new tile to the launchpad.
Select "App Launcher – Static".

Figure 7: App launcher

Fill in the information as followed:
Title: Title of the application displayed in the "App Tile".
Subtitle: Subtitle of the application displayed in the "App Tile".
Keywords: Keywords or tags that can be used in searches.
Icon: Icon that is displayed in the "App Tile".
Information: Additional information about the application.

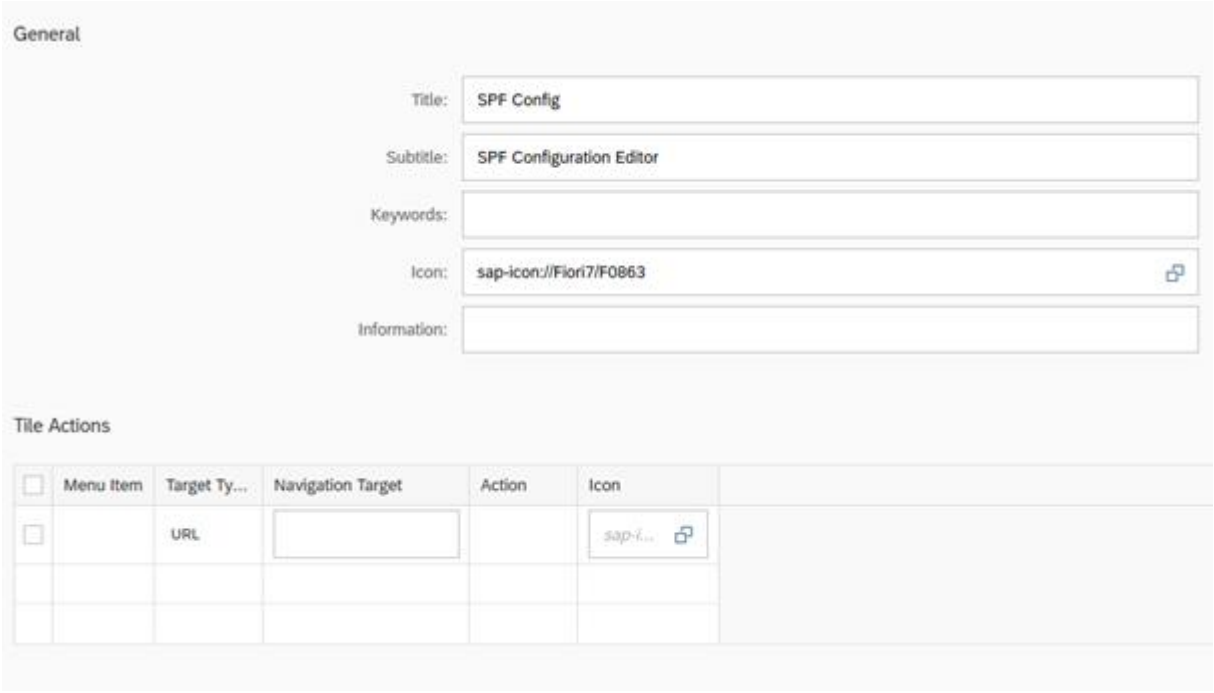


Figure 8: Application configuration (general)

Use semantic object navigation: Selected to use semantic object navigation
Semantic Object: "Semantic Object" that was defined in chapter Semantic Object
Action: Action the Application Tile needs to perform
Target URL: Automatically filled through the semantic object navigation and action

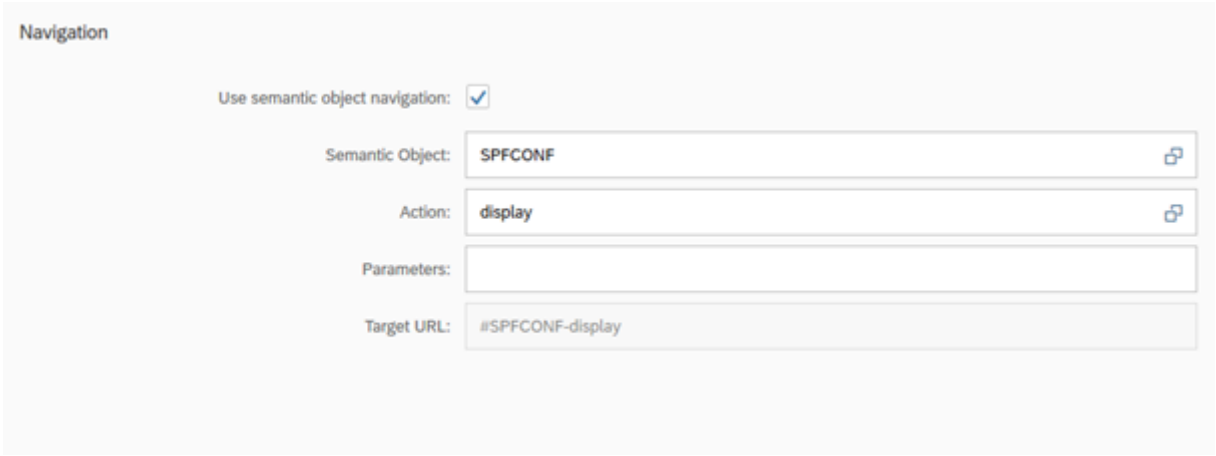


Figure 9: Application configuration (navigation)

2.1.3.3 Target Mapping

After the tile creation has been saved a "Target Mapping" needs to be created. To create a new target mapping select the "Target Mapping" tab in the *Fiori Launchpad Designer*.

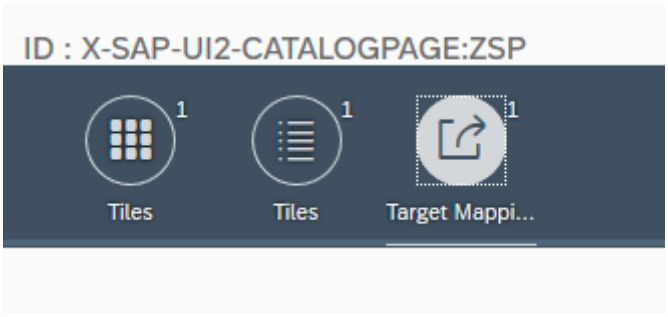


Figure 10: Target mapping

After selecting the "Target Mapping" tab select "Create Target Mapping".

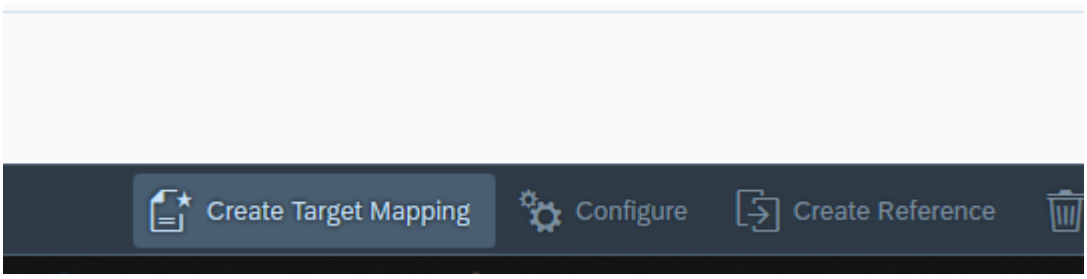


Figure 11: Create target mapping

Enter the following Information in the Configuration:
Semantic Object: Name of the created "Semantic Object"
Action: Display
Application Type: SAPUI5 Fiori App
URL: URL of the Fiori application
ID: Component.js name of the Fiori application

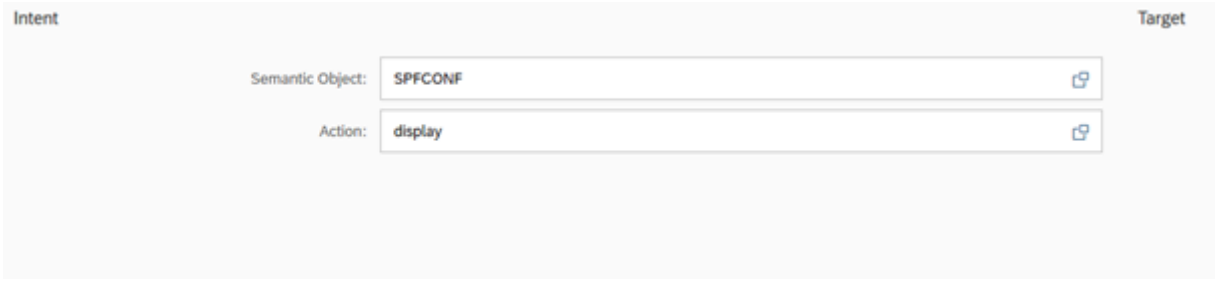


Figure 12: Target mapping (intent)

Within the Target information enter the following URLs and IDs for the corresponding application.

Application	URL
SPF Configuration	/sap/bc/ui5_ui5/plga/spfconfig
SPF Monitoring	/sap/bc/ui5_ui5/plga/spfmonitoring

Table 2: URL configuration

Application	ID
SPF Configuration	com.sap.pe.spfconfig.SPFConfig
SPF Monitoring	com.sap.pe.spfconfig.SPFMonitoring

Table 3: Application component

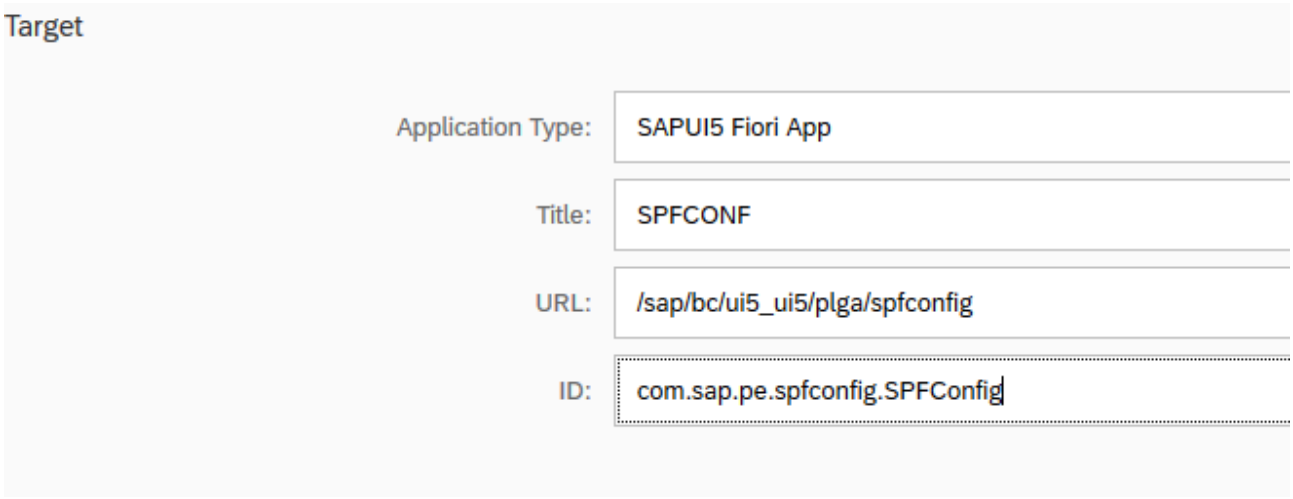


Figure 13: Target mapping (target)

These steps should be repeated for each application.

2.1.3.4 Business Group creation

To create a Business Group, switch to the tab "Groups" in the *Fiori Launchpad Designer*.

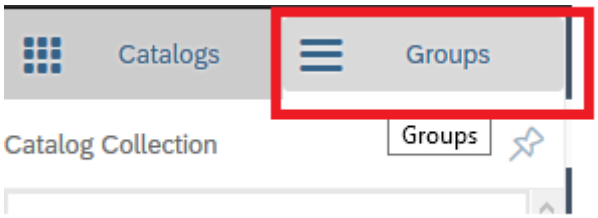


Figure 14: Business group

Then enter the details in the following window

Title: Title of the group

ID: ID of the group

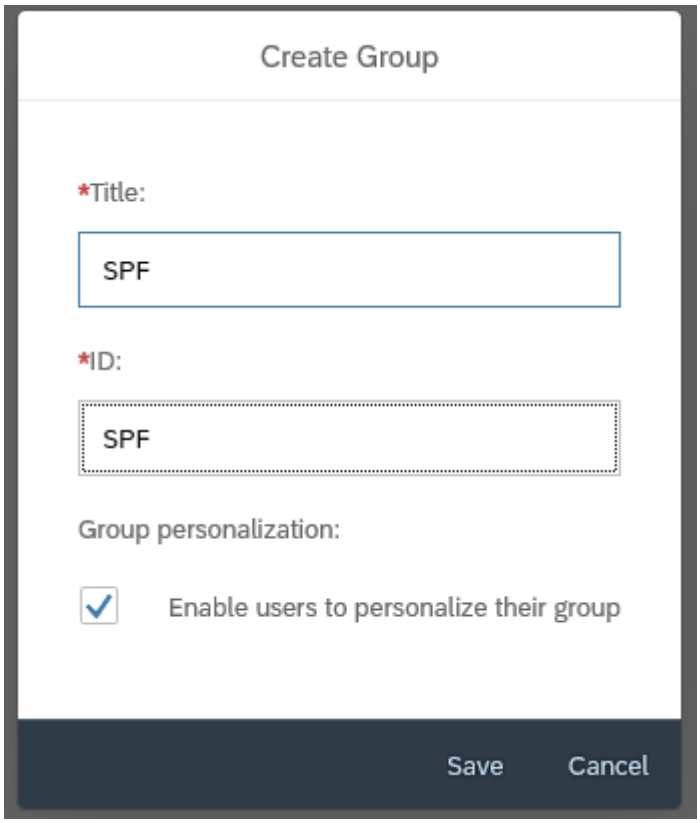


Figure 15: Business group (creation)

Once the group has been created, press the "Add Tile" button and select the previously created catalogue.

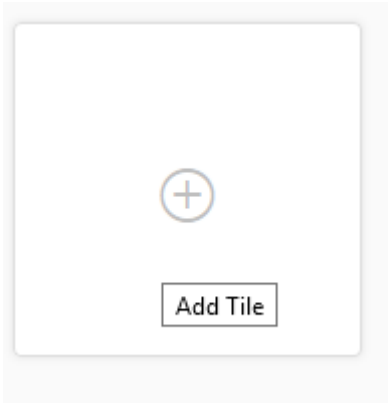


Figure 16: Business group (tile)

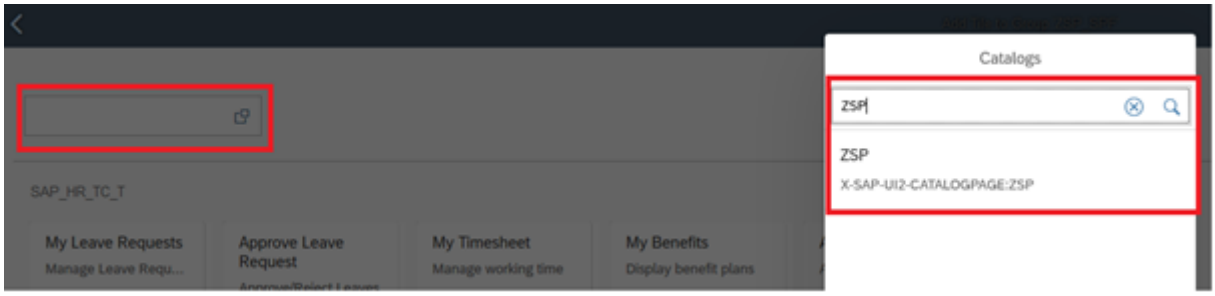


Figure 17: Business group (application)

Once selected every application that is assigned to the selected catalogue will be displayed and can be added to the group by pressing the plus (+) sign underneath the tile.

Figure 18: Business group (add application)

2.1.3.5 PFCG Roles

To configure the "PFCG Roles" navigate to the Transaction *PFCG* within the SAP System. The "PFCG Roles" are necessary to customize the accessibility of the created catalogue and group.

First of a "PFCG Role" needs to be created. Enter the desired name for the role and press "Single Role".

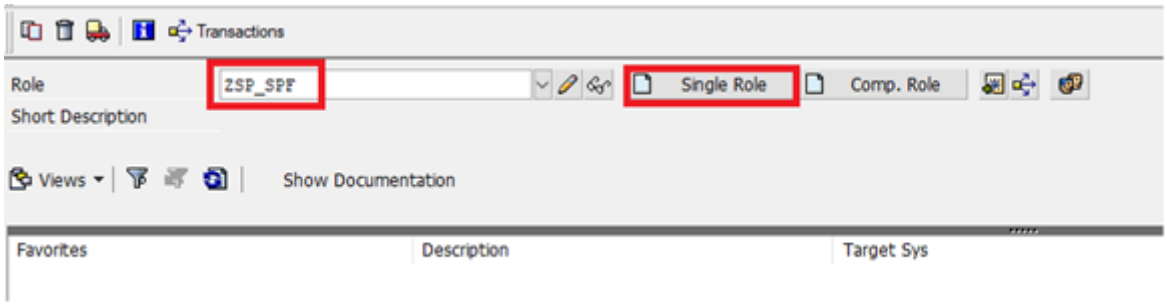


Figure 19: PFCG roles

Within the Role Customization Navigate to the tab "Menu" and select "Authorization Default" from the transaction submenu.

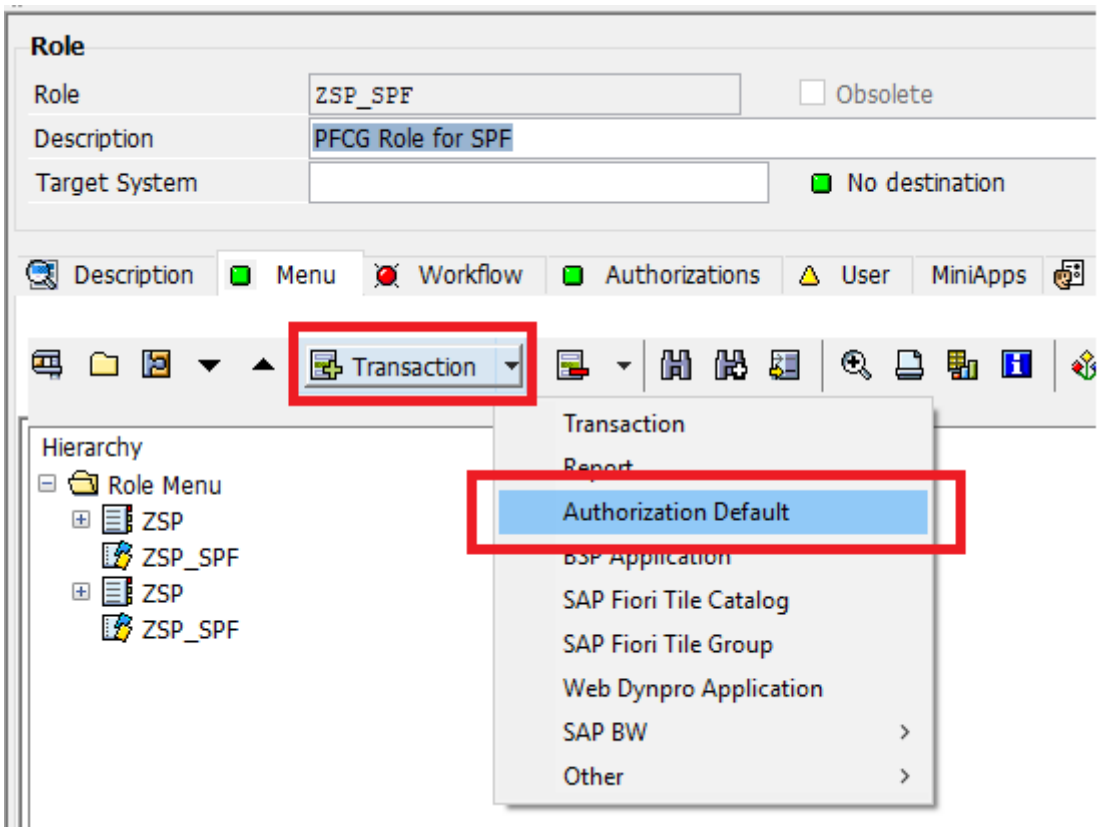


Figure 20: PFCG authorization

Within the dialog enter the specified roles that are needed for users within your SAP System to use *Frontend Server Applications*.

2.1.3.6 Authorizations

After the required roles have been specified or if none of these are needed or already assigned to the desired users navigate to “Authorizations” tab.

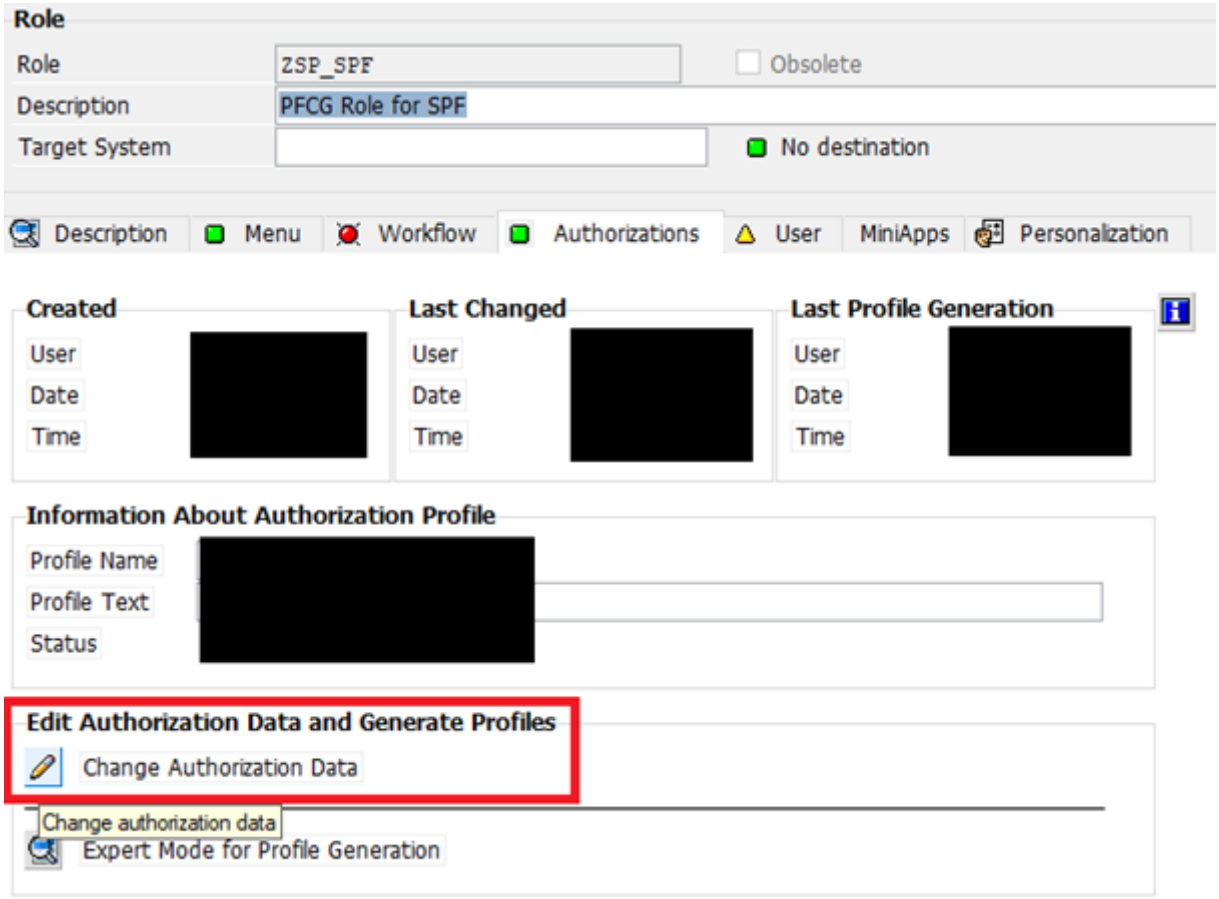


Figure 21: PFCG authorization data

Press the "Change Authorization Data" button and enter the specified "Authorization Profile" that within your organization grants access to *Fiori*/oData services.

2.1.3.7 Catalogues and Groups

Navigate to the "Menu" tab to define the Launchpad and Group within the Role next. Within the "Menu" tab select "Transaction" and then "SAP Fiori Tile Catalog".

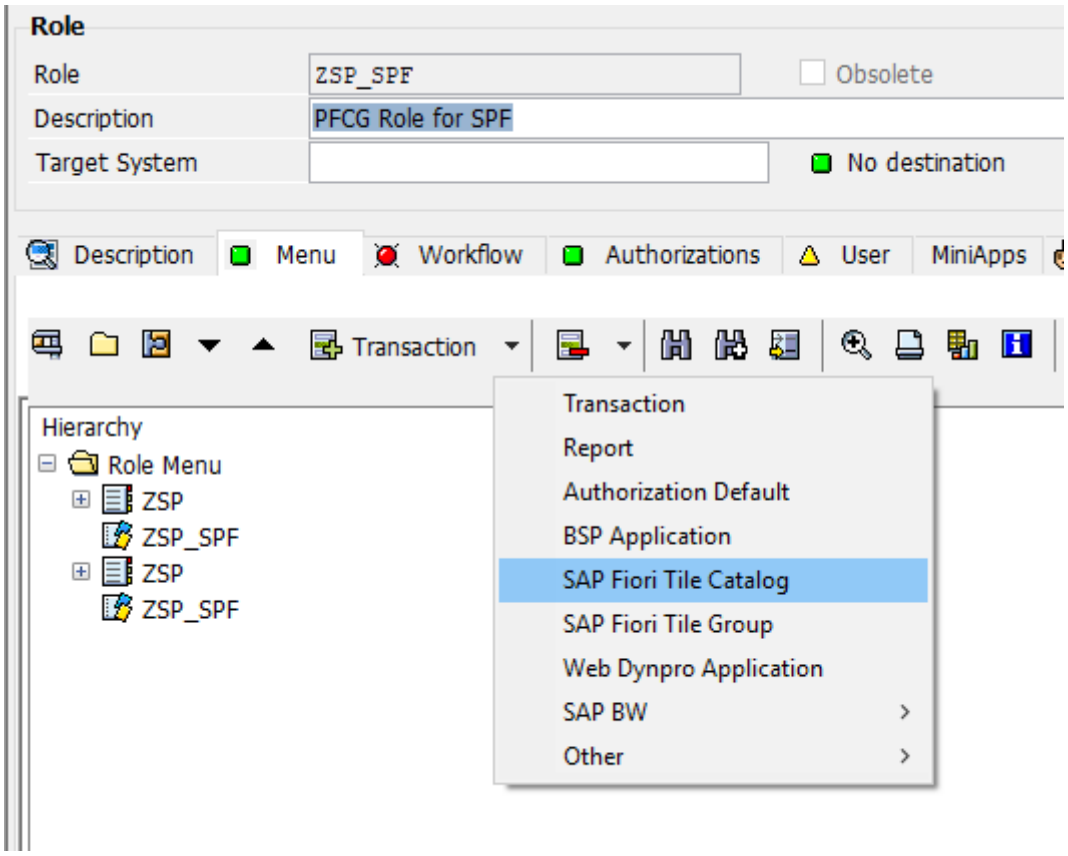


Figure 22: PFCG tile catalog

Within the selection dialog select the previously created catalogue.

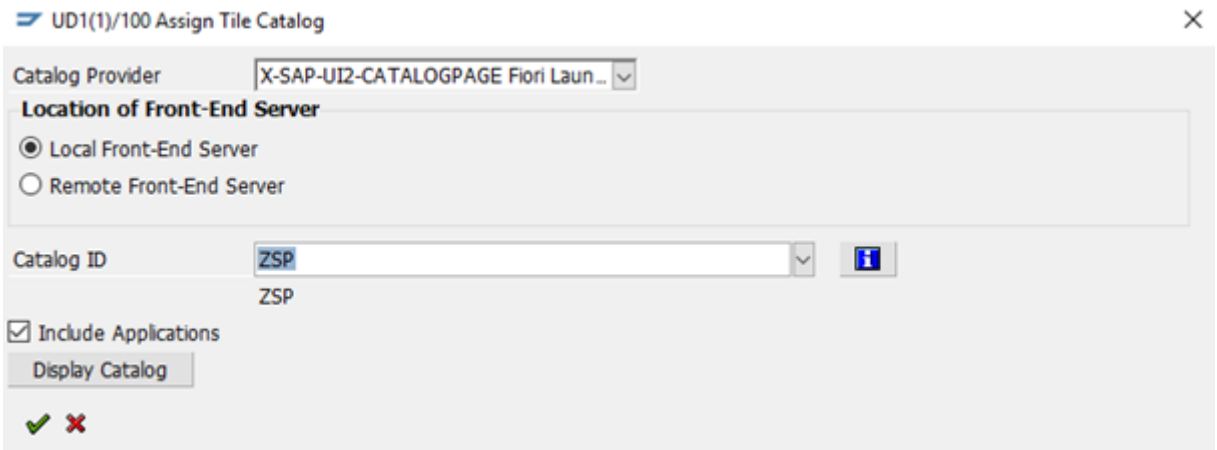


Figure 23: PFCG tile catalog configuration

After the catalogue has been added select the "Transaction" button again, which now should be labelled "SAP Fiori Tile Catalog" and select "SAP Fiori Tile Group" from the Dropdown.

Select the group that was defined previously and add it to the role.

Figure 24: PFCG groups

2.1.3.8 User

Navigate to the "User" tab and add all the users that you want to have access to this catalogue and group.

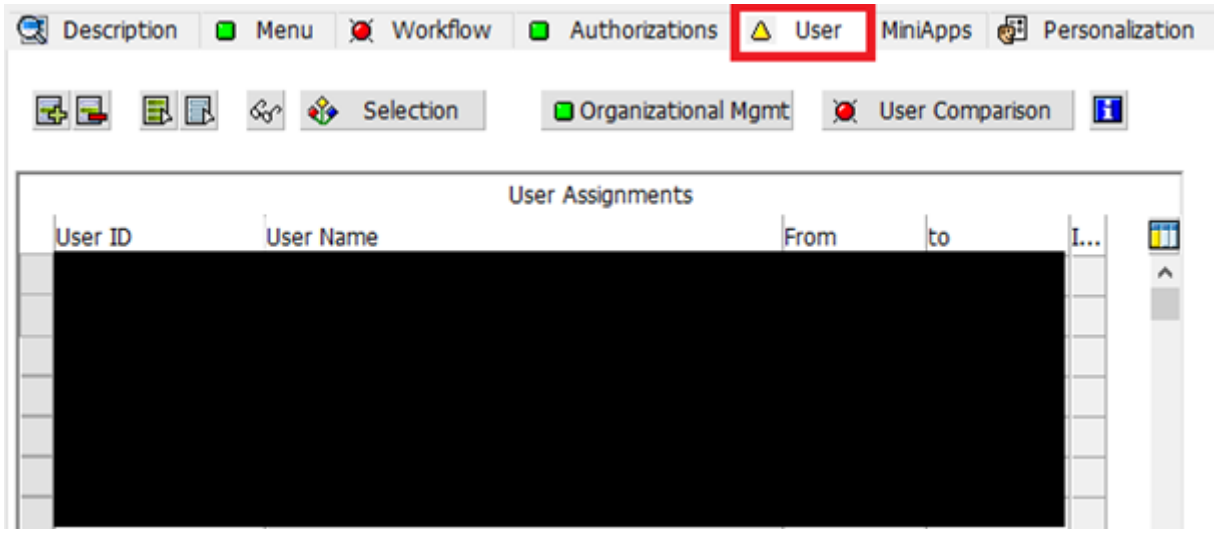


Figure 25: PFCG User

2.1.3.9 Activate services (business server pages and UI5 applications)

In some cases, it can occur that the applications are inactive. Please check the transaction *SICF* if the relevant application-services are started.

The following paths (relative from host) should be activated to use the full functionality:

sap/bc/bsp/plga/spfconfig
sap/bc/bsp/plga/spfmonitoring
sap/bc/ui5_ui5/plga/spfconfig
sap/bc/ui5_ui5/plga/spfmonitoring

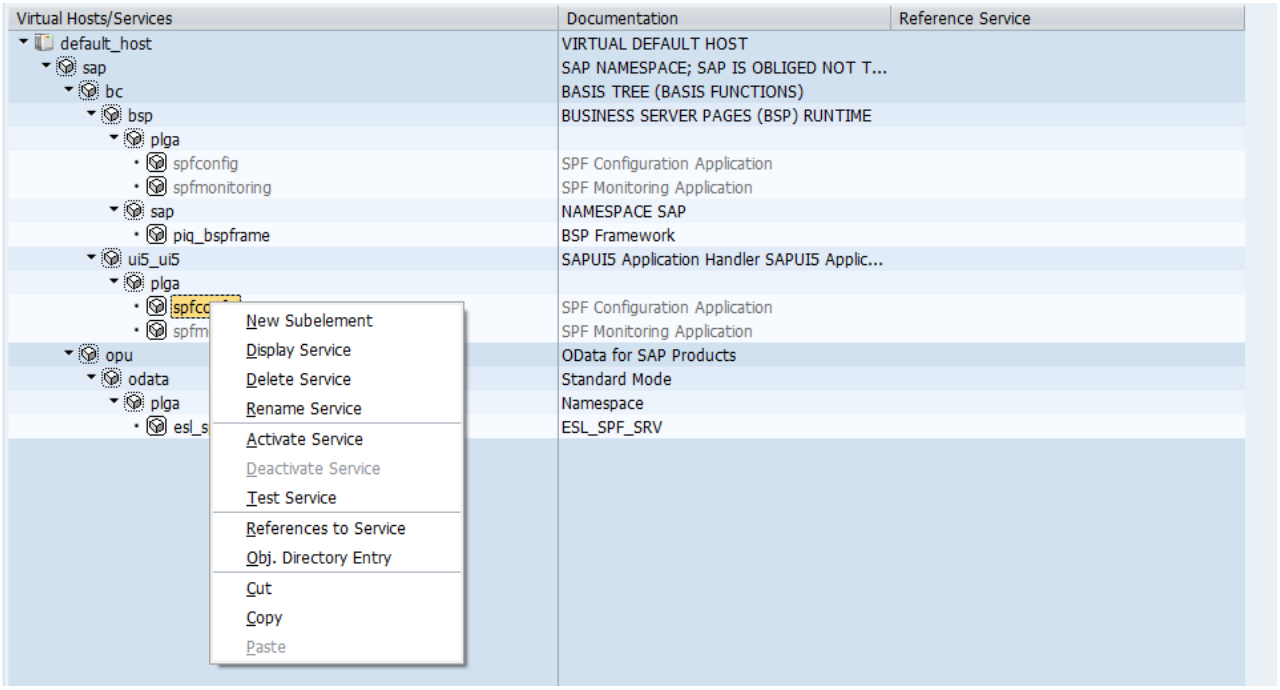


Figure 26: Example selection SICF UI5

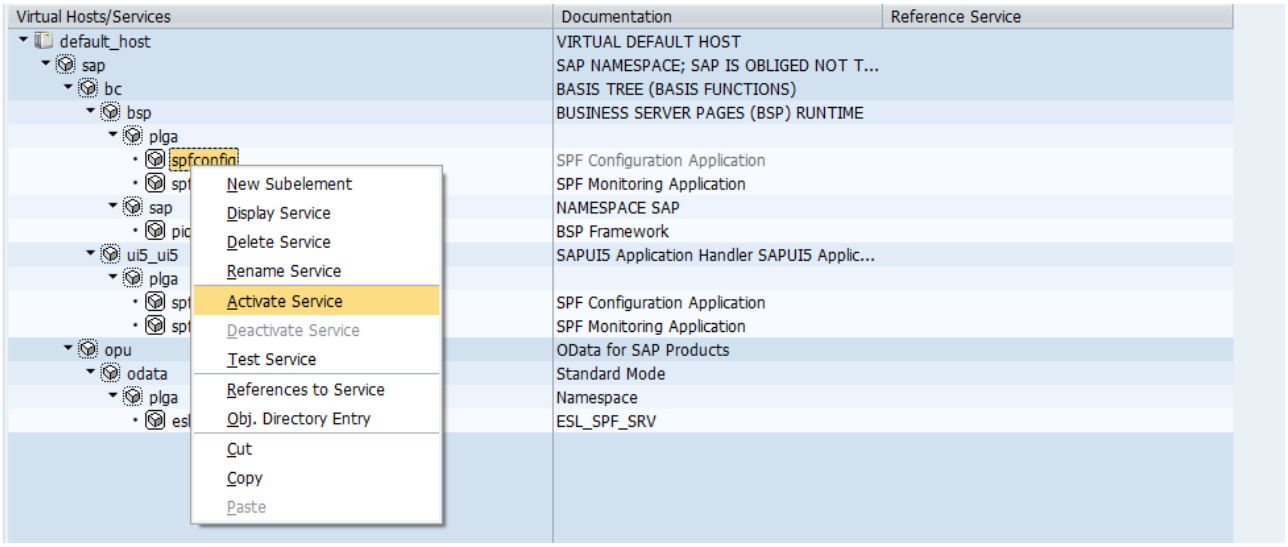


Figure 27: Example selection SICF BSP

2.1.4 Maintaining Lanes

Figure 28: Lane example, displayed in UI5 application

Lanes symbolizes process families or specific process segments within a complex process.

Within the *Simple Process Flow* lanes (consisting of controller classes) will be instantiated on the first occurrence in the process. If a lane is called twice in the process, the same class instance can be used, if the method implementation is done correctly.

The lane controller classes can be implemented using the following interface:

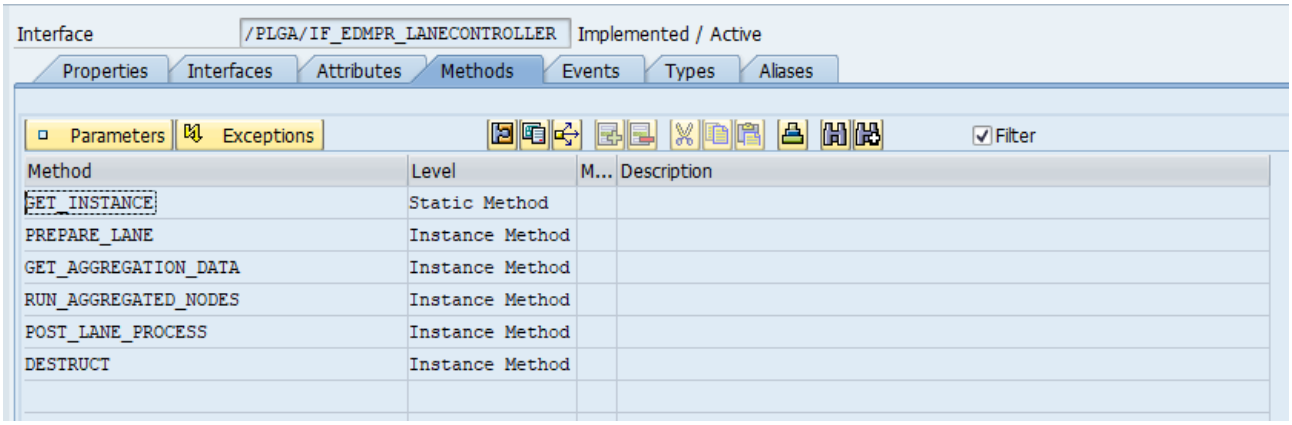


Figure 29: Lane controller interface

Method GET_INSTANCE

This Method returns the instantiated object of the controller (must implement the interface).

Method PREPARE_LANE

Right after the initialization of the control class, this method submits the current lane configuration and the current process instance (instance of class /PLGA/CL_EDMPR_NODECONTROLLER) to the lane controller.

Method GET_AGGREGATION_DATA

If a node is defined as "aggregated", this method will be called with the list of all sequential following nodes with the equal flag and the same lane.

Method RUN_AGGREGATED_NODES

When a node was marked as "aggregated" and the method GET_AGGREGATION_DATA from the lane controller returns a node list, this method has to process this list of nodes.

Method POST_LANE_PROCESS

When the last node of a lane was processed, this method is called. This method can be used to pause or cancel processes gracefully.

Method DESTRUCT

This method can be used to free internal data of the instance and to inform the process to save the instance e.g. for an additional use within the process.

The following controller class implementations are shipped within the addon:

- */PLGA/CL_EDMPR_DLINE_LANECTR Lanecontroller for deadline management*
  - The Lane Controller manages the behaviour of the process instance accordingly to the deadline setting. Therefore, the only implemented method is "POST_LANE_PROCESS", whose outcome can be controlled by the public method "SET_DEADLINE_EXCEEDED".
  - Dependent node controller for calculating timestamp starting from process date or current processing time is */PLGA/CL_EDMPR_DLINE_NODECTR*
  - The control class is not capable of processing aggregated nodes
- */PLGA/CL_EDMPR_NOOP_LANECTR Lanecontroller without business logic*
  - This control class can be used as "dummy" for nodes, which can run stand-alone
  - The control class is not capable of processing aggregated nodes
- */PLGA/CL_EDMSET_LANE_CTRL Lane Controller for Settldoc management*

- This control class is described in Dynamic EDM Settlement extension

The maintenance of lanes can be done within the SAP GUI in the transaction *SPRO (SAP Reference IMG)*. The following path leads to the view cluster for *Simple Process Flow* configuration:

SAP Customizing Implementation Guide ->Extension for Utilities by PROLOGA ->Energy Data Management -> Simple Process Flow -> SPF: Process configuration

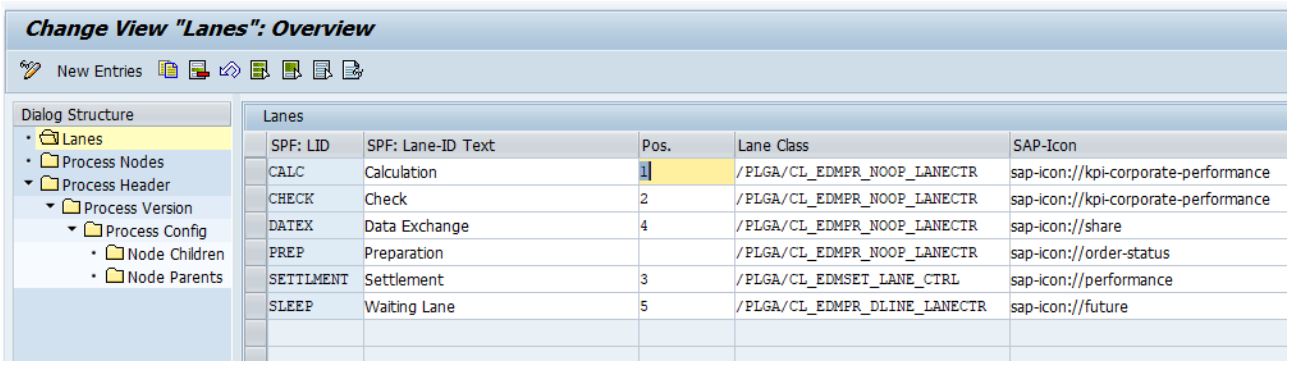


Figure 30: SPF: Lane configuration

Column	Data Element	Description
SPF: Lane Identification	/PLGA/DEDMPR_LID	Identification of the Lane
SPF: Lane-ID Text	/PLGA/DEDMPR_LIDT	Long text for lane description
Lane position	/PLGA/DEDMPR_LANE_POS	Position within the UI
SPF: Lane Classes	/PLGA/DEDMPR_LANECLS	Control class
SAP-Icon	/PLGA/DEDMPR_SAPICON	Icon for UI

Table 4: SPF Lane maintenance

The Lane position must start at position 0 and must be sequential without gaps regarding all lanes

The SAP-Icon can be retrieved from e.g. SAPUI5-Icon-Explorer

2.1.5 Maintaining Nodes

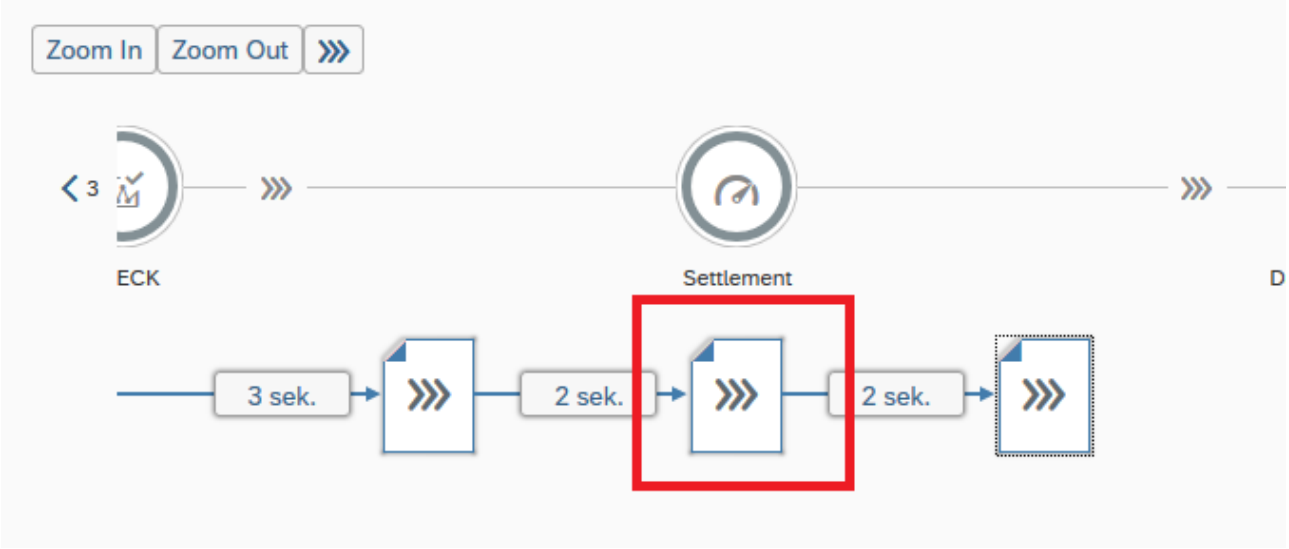


Figure 31: Node appearance example

The Node symbolizes a process step, or a process call within the *Simple Process Flow*. A node is dedicated to exact one lane. If a node has to be used multiple times within the same process identification, it has to be duplicated.

A Node technically consists of a class instance, implementing the interface */PLGA/IF_EDMPR_NODECONTROLLER*.

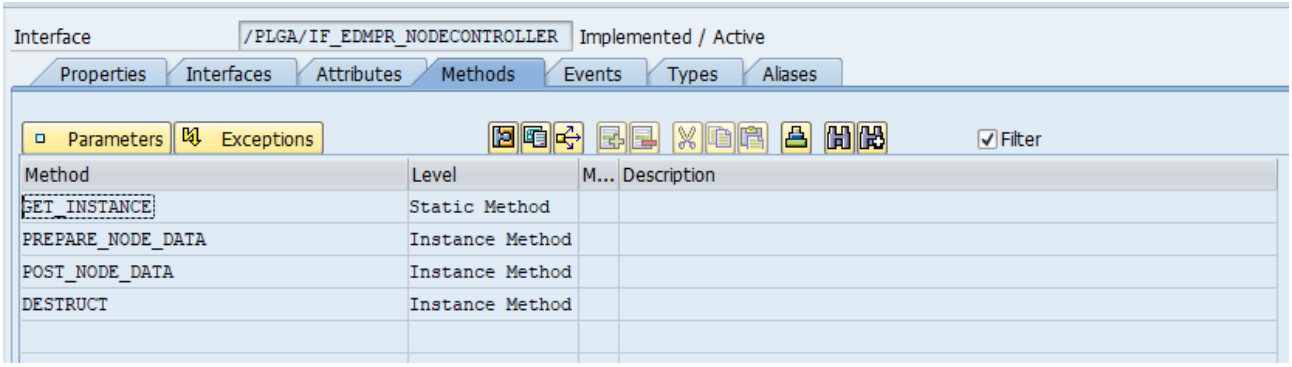


Figure 32: Interface to Node control class

Method GET_INSTANCE

This Method returns the instantiated object of the controller (must implement the interface).

Method PREPARE_NODE_DATA

Right after the initialization of the control class, this method submits the current node configuration, the lane class instance and the current process instance (instance of class /PLGA/CL_EDMPR_NODECONTROLLER) to the node controller.

The Method must return an instantiated object, a method and the corresponding parameter list for a dynamic object execution. Additionally, this method can skip also the execution of itself by setting exporting parameter *ev_skip_node*.

Method POST_LANE_PROCESS

After execution of the previous mentioned object, this method is called with the exception data and the parameter table. In case of errors (determined within the exception or parameter data), the method can be used to cancel processes gracefully.

Method DESTRUCT

This method can be used to free internal data of the instance and to inform the process to save the instance e.g. for an additional use within the process. The feature of reusing node instances currently relies on heavy individual implementation.

The node controller is responsible to set the correct status of the node data (changing parameter in every method). The process execution only handles status changed on error occurrences.

The following controller class implementations are shipped within the addon:

- */PLGA/CL_EDMPR_DLINE_NODECTR nodecontroller for deadline management*
  - The Node Controller calculates and check the deadline of itself, according to the deadline setting.If the deadline is reached or exceeded, the method "POST_NODE_DATA" calls the method *set_deadline_exceeded* of the connected lane instance.

    Deadline calculation support:

    Relative to process date:
    - days (interval type "D"),
    - months (interval type "M")
    - and years (interval type "Y")

    Pattern of External Identifcation (described in the following):

    P_<interval type>_<interval length>

    The calculation is done relative to the to the process date at 0 o'clock in time zone UTC.

    Relative to current system date and time:
    - seconds (interval type "SEC")
    - minutes (interval type "MIN")
    - hours (interval type "H")
    - days (interval type "D"),
    - months (interval type "M")
    - and years (interval type "Y")

    Pattern of External Identifcation (described in the following):

    C_<interval type>_<interval length>

- */PLGA/CL_EDMPR_ACW_NODECRT Nodecontroller for AutocalcWB*
  - This control class is described in Simple Process Flow connector
- */PLGA/CL_EDMPR_ECF_NODECRT Cluster Framework Node*
  - This control class is described in Simple Process Flow connector
- */PLGA/CL_EDMPR_ECVBILLTABLE_NC Nodecontroller for billingtable update in ECV*
  - This control class is described in Simple Process Flow connector
- */PLGA/CL_EDMPR_ECVREF_PV_NC Nodecontroller for billingtable update in ECV*
  - This control class is described in Simple Process Flow connector
- */PLGA/CL_EDMPR_EU3_NODECRT Nodecontroller for EU3*
  - This control class is described in Simple Process Flow connector
- */PLGA/CL_EDMSET_NODE_CTRL Node controller for settlement steps*
  - This control class is described in Dynamic EDM Settlement extension

The maintenance of lanes can be done within the SAP GUI in the transaction *SPRO (SAP Reference IMG)*. The following path leads to the view cluster for *Simple Process Flow* configuration:

SAP Customizing Implementation Guide ->Extension for Utilities by PROLOGA ->Energy Data Management -> Simple Process Flow -> SPF: Process configuration

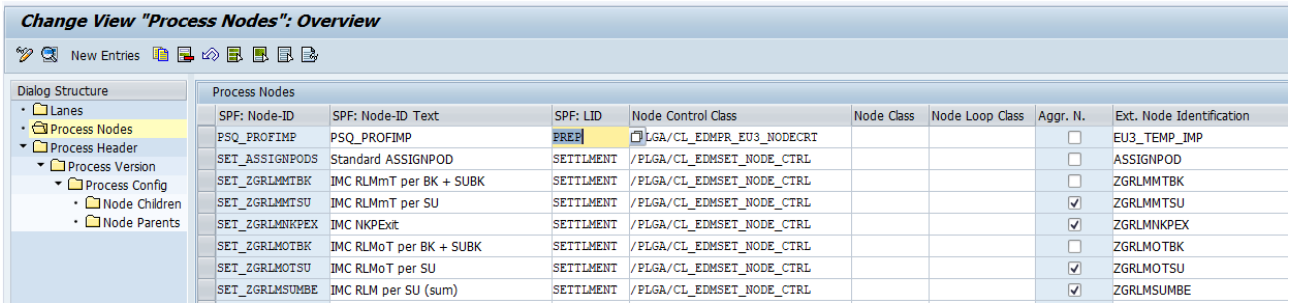


Figure 33: SPF: Node configuration

Column	Data Element	Description
SPF: Node Identification	/PLGA/DEDMPR_NID	Identification of the node
SPF: Node Identification Text	/PLGA/DEDMPR_NIDT	Long text description
SPF: Lane Identification	/PLGA/DEDMPR_LID	ID of the connected lane
SPF: Node Control class	/PLGA/DEDMPR_NODECTRCLS	Control class
SPF: Node class	/PLGA/DEDMPR_NODECLS	Object class, if needed by the node controller
SPF: Node Loop Provider class	/PLGA/DEDMPR_LOOPCLS	Loop provider class, if needed by the node controller. E.g. on aggregated nodes
SPF: Aggregated Node (Loop provider needed)	/PLGA/DEDMPR_LOOP_ACTIVE	Disables the execution within the process, execution must be provided by the lane controller
SPF: External Node Identification	/PLGA/DEDMPR_EXTID	Additional information if needed by the node controller

Table 5: SPF Node maintenance

2.1.6 Maintaining Process identifications

The maintenance of Processes can be done within the SAP GUI in the transaction *SPRO (SAP Reference IMG)*. The following path leads to the view cluster for *Simple Process Flow* configuration:

SAP Customizing Implementation Guide ->Extension for Utilities by PROLOGA ->Energy Data Management -> Simple Process Flow -> SPF: Process configuration

For maintaining processes, it is necessary, that the section "Process Header" is active.

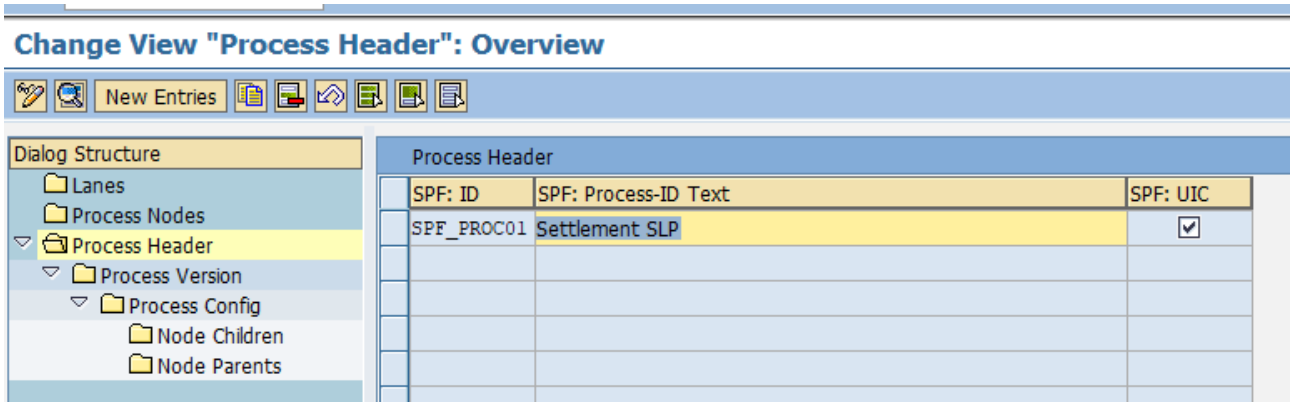


Figure 34: SPF: Process Header - Configuration

Here the following information are available:

Column	Data Element	Description
SPF: ID	/PLGA/EDMPR_PRID	Identification of the ID for the process

SPF: Process-ID Text	TEXT50	Long text description for process
SPF: UIC	/PLGA/DEDMPR_PR_UICONFIG	Flag for activation configuration via UIC. This flag must be active ☑ otherwise the configuration can only be done within the SAP GUI.

Table 6: SPF Process Header maintenance

2.1.7 Maintaining Version

The maintenance of Process Versions can be done within the SAP GUI in the transaction *SPRO (SAP Reference IMG)*. The following path leads to the view cluster for *Simple Process Flow* configuration:

SAP Customizing Implementation Guide ->Extension for Utilities by PROLOGA ->Energy Data Management -> Simple Process Flow -> SPF: Process configuration

For initial maintenance it is necessary that an initial version is maintained.

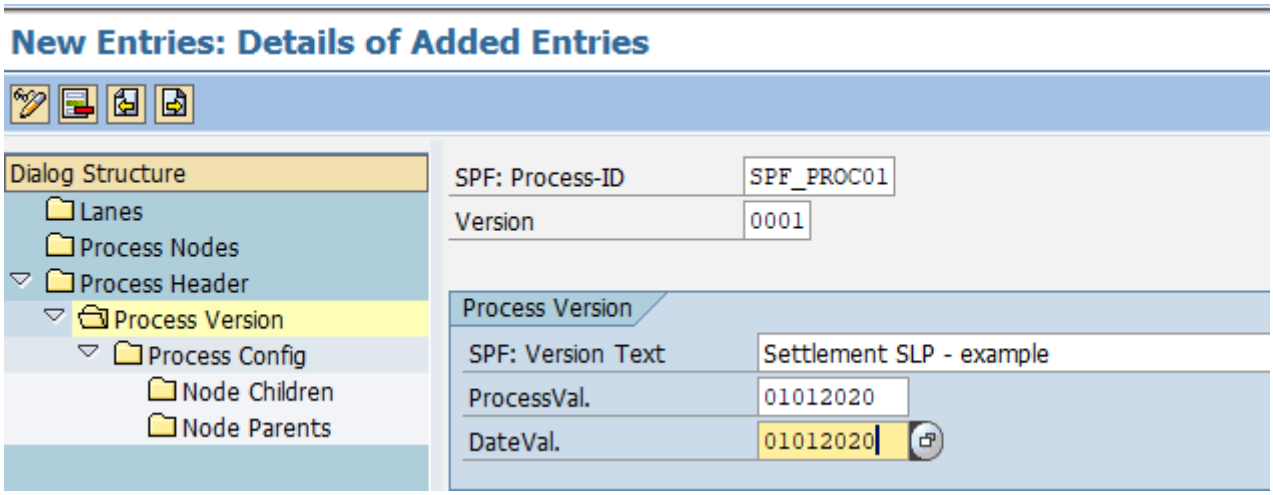


Figure 35: SPF: Process Version - Configuration

Here the following information are available:

Column	Data Element	Description
SPF: Process ID	/PLGA/DEDMPR_PRID	Enter the SPF Process-ID
Version	/PLGA/DEDMPR_VERSION	Enter the Version number
SPF: Version Text	/PLGA/DEDMPR_VERSIONTEXT	Enter a Description for the version.
ProcessVal.	/PLGA/DEDMPR_VERSION_PROCDATE	Enter date for which the process is valid
DateVal.	/PLGA/DEDMPR_VERSION_SYSDATE	Enter date for which the process is valid

Table 7: SPF Process Version maintenance

2.1.8 Maintaining Process Config

The maintenance of Process Config can be done within the SAP GUI in the transaction *SPRO (SAP Reference IMG)*. The following path leads to the view cluster for *Simple Process Flow* configuration:

SAP Customizing Implementation Guide ->Extension for Utilities by PROLOGA ->Energy Data Management -> Simple Process Flow -> SPF: Process configuration

Within the process configuration it is possible to maintain the process by assigning nodes, lanes, and its parent- and child-definitions.

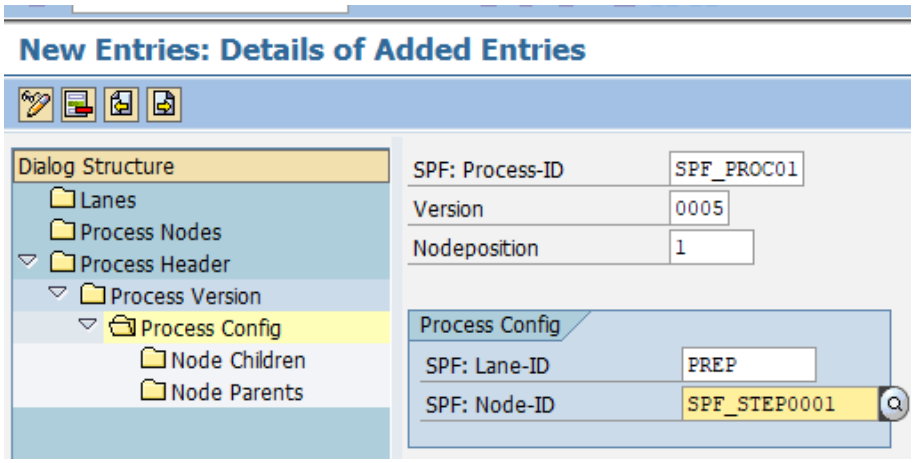


Figure 36: SPF: Configuration of Process via SAPUI

Here the following information are available:

Column	Data Element	Description
SPF: Process ID	/PLGA/DEDMPR_PRID	Enter the SPF Process-ID
Version	/PLGA/DEDMPR_VERSION	Enter the Version number
Nodeposition	/PLGA/DEDMPR_CNF_LFNR	Enter the Nodeposition. The node position must start with the number "1"
SPF: Lane-ID	/PLGA/DEDMPR_LID	Enter the Lane-Id for the node
SPF: Node-ID	/PLGA/DEDMPR_NID	Enter the Node-Id for the chosen lane

Table 8: SPF Process Version maintenance

To support the maintenance via UI5 application, every node must have the correct parents and children, when defining multiple nodes in the process version. The maintenance with the UI5 application is recommended.

2.1.9 Maintaining Node Children

The maintenance of Node Children can be done within the SAP GUI in the transaction *SPRO (SAP Reference IMG)*. The following path leads to the view cluster for *Simple Process Flow* configuration:

SAP Customizing Implementation Guide ->Extension for Utilities by PROLOGA ->Energy Data Management -> Simple Process Flow -> SPF: Process configuration

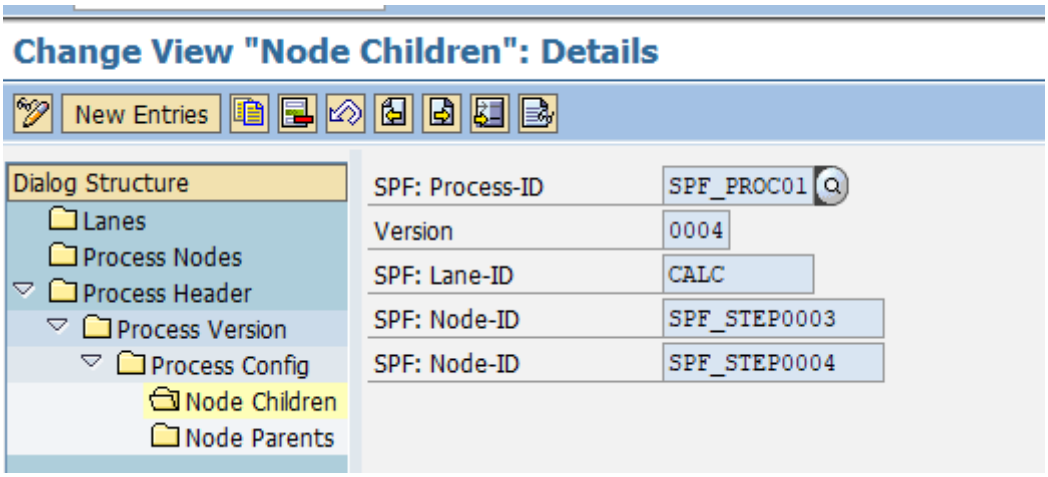


Figure 37: SPF: Configuration of Node Children

Here the following information are available:

Column	Data Element	Description
SPF: Process ID	/PLGA/DEDMPR_PRID	Enter the SPF Process-ID
Version	/PLGA/DEDMPR_VERSION	Enter the Version number
SPF: Lane-ID	/PLGA/DEDMPR_LID	Enter the Lane-Id for the node
SPF: Node-ID	/PLGA/DEDMPR_NID	Enter the Node-Id for the parent (current) node
SPF: Node-ID	/PLGA/DEDMPR_NID	Enter the Node-Id for the child node

Table 9: SPF Process Node Children

The current node can have multiple children.

2.1.10 Maintaining Node Parent

The maintenance of Node Parent can be done within the SAP GUI in the transaction *SPRO (SAP Reference IMG)*. The following path leads to the view cluster for *Simple Process Flow* configuration:

SAP Customizing Implementation Guide ->Extension for Utilities by PROLOGA ->Energy Data Management -> Simple Process Flow -> SPF: Process configuration

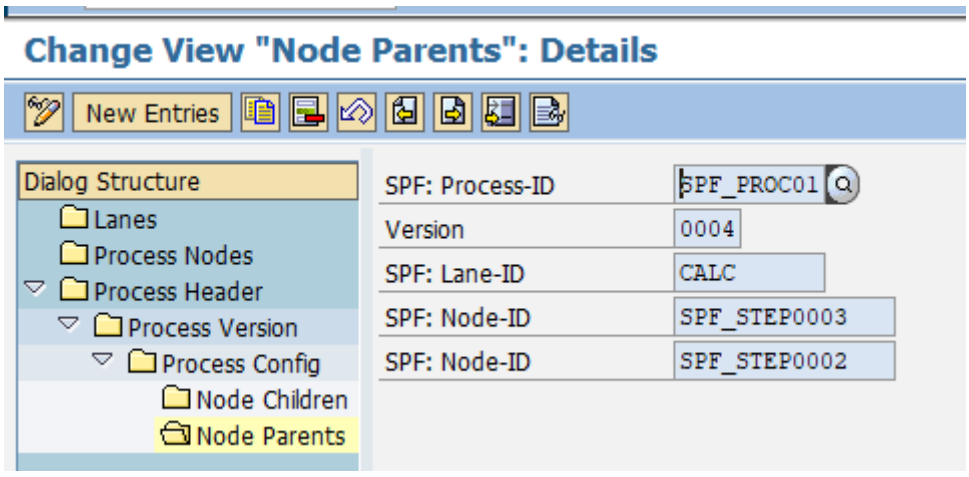


Figure 38: SPF: Configuration of Node Children

Here the following information are available:

Column	Data Element	Description
SPF: Process ID	/PLGA/DEDMPR_PRID	Enter the SPF Process-ID
Version	/PLGA/DEDMPR_VERSION	Enter the Version number
SPF: Lane-ID	/PLGA/DEDMPR_LID	Enter the Lane-Id for the node
SPF: Node-ID	/PLGA/DEDMPR_NID	Enter the Node-Id for the current node
SPF: Node-ID	/PLGA/DEDMPR_NID	Enter the Node-Id for the parent node

Table 10: SPF Process Node Children

The current node can have multiple parents.

2.1.11 Configuration of processes via SAP GUI

The maintenance of Process can be done within the SAP GUI in the transaction *SPRO (SAP Reference IMG)*. The following path leads to the view cluster for *Simple Process Flow* configuration:

SAP Customizing Implementation Guide ->Extension for Utilities by PROLOGA ->Energy Data Management -> Simple Process Flow -> SPF: Process configuration

The process configuration can be done completely in the SAP Gui. If the usage of the UI5 application is not desired, the specification children and parents can be omitted. The processing engine only needs the node position run the process workflow.

Processes configured in the described way should be marked without the UI configuration flag. Without correct parent and child relations, the maintenance via UI5 application is not possible.

2.1.12 Configuration of processes via WebUI

The following steps are relevant for process maintenance using the WebUI application. It is necessary, that some pre-processing is done within the SAP GUI before the configuration via WebUI can be started.

2.1.12.1 Pre-processing configuration within SAP GUI

Before a configuration within the WebUI could take place, it is necessary to maintain the main process within the SAP GUI. The following minimum of information must be maintained:

- Lanes
- Nodes
- Process
    - One process version
    - Any first step within this version

For detailed information for maintenance please see the further sections and information.

The first step is to define the Process Header. Here it is necessary that the config for "SPF: UIC" is active.

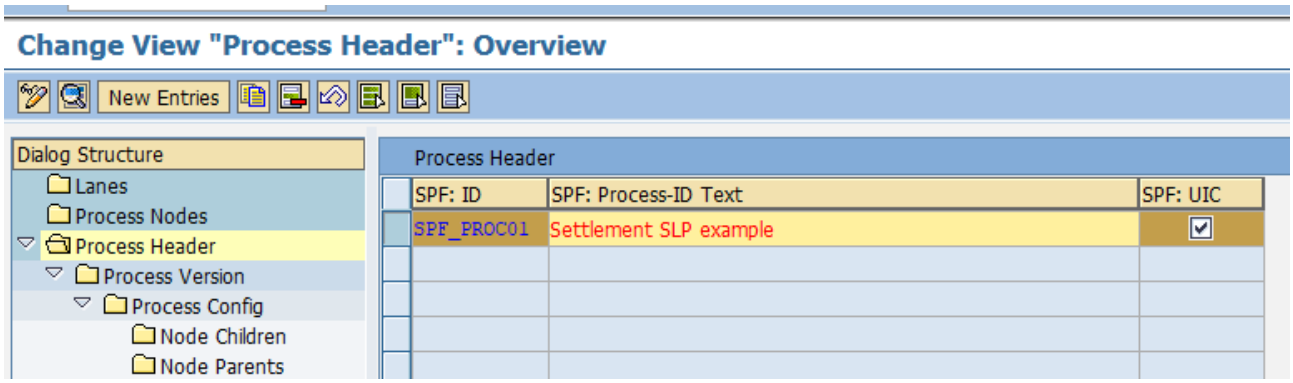


Figure 39: SPF: Configuration of Process via WebUI – 1 -

The second step is to define the Process Header. Here it is necessary that the config for "SPF: UIC" is active.

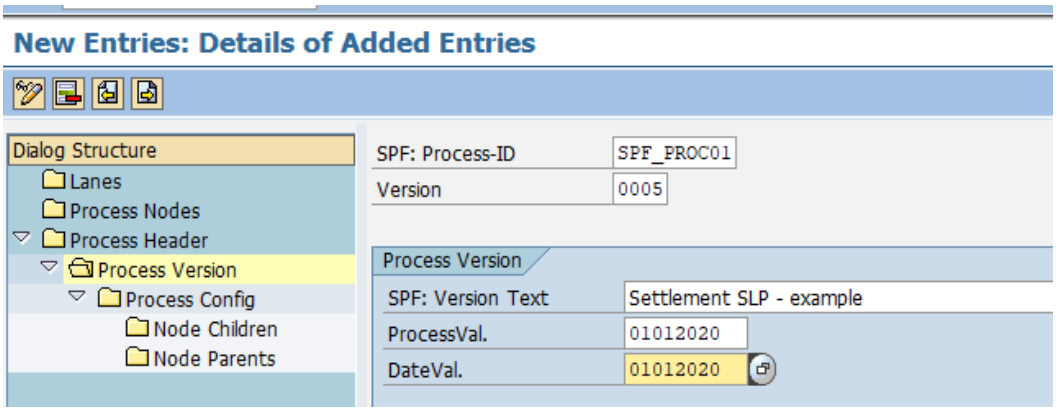


Figure 40: SPF: Configuration of Process via WebUI – 2 -

The third step is to define the first node within the process.

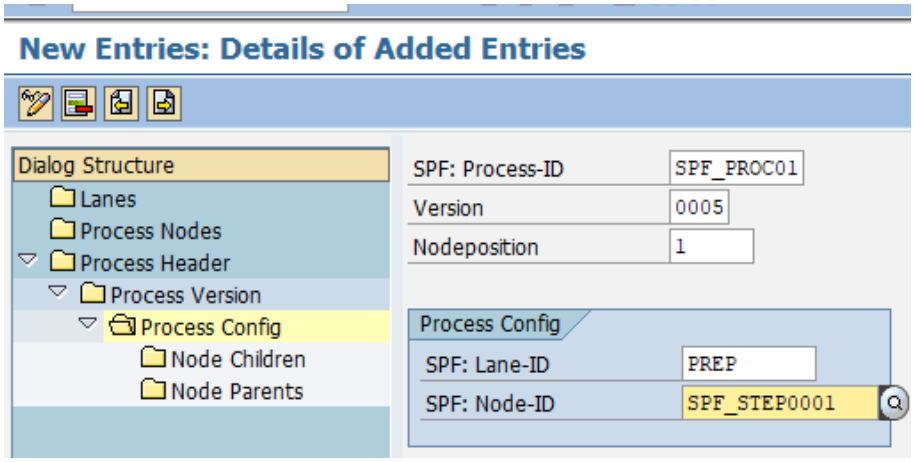


Figure 41: SPF: Configuration of Process via WebUI – 3 -

2.1.12.2 Configuration within Web UI

To start configuring a process via the UI5 application, the dedicated launchpad has to be started. Afterwards the SPF configuration UI can be started via the corresponding tile like the following example.

Figure 42: SPF: Configuration of Process via WebUI – 1 -

Afterwards all available versions are displayed. Here must the relevant process version be chosen by clicking on the correct line.

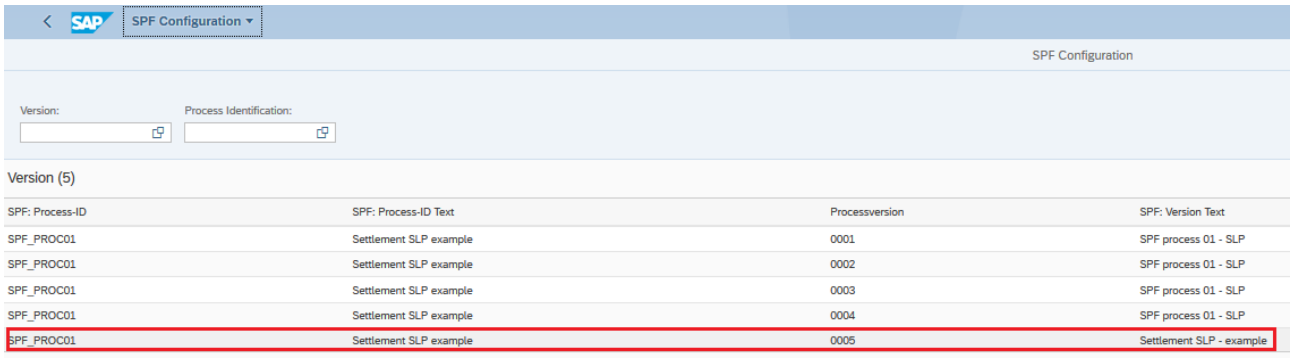


Figure 43: SPF: Configuration of Process via WebUI – 2 -

Afterwards all maintained nodes are available on the left side of the window.

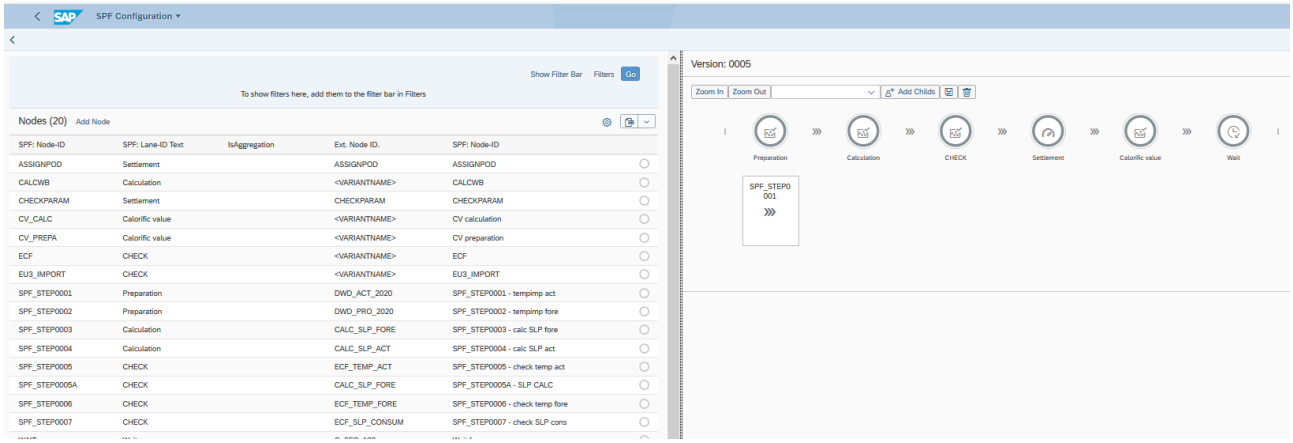


Figure 44: SPF: Configuration of Process via WebUI – 3 -

To maintain a process the following steps are possible.

Add a new node a child from an existing node:

To add a new node the following steps are necessary:

- select an existing node

- select the node to add
- press button "Add Node"

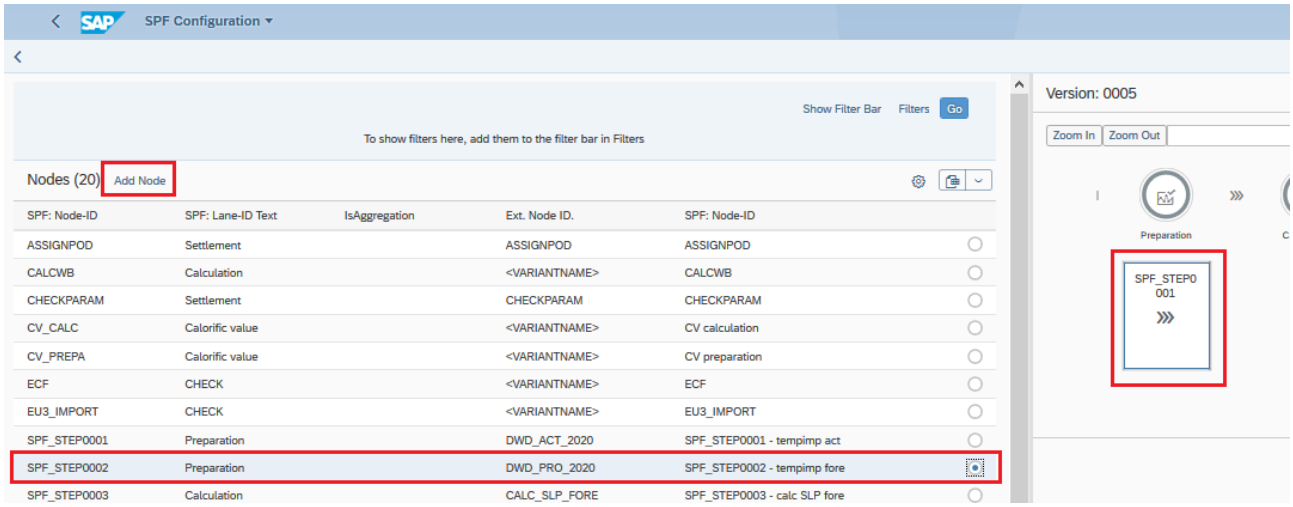


Figure 45: SPF: Configuration of Process via WebUI – 4 -

The selected node is entered as child of the existing node:

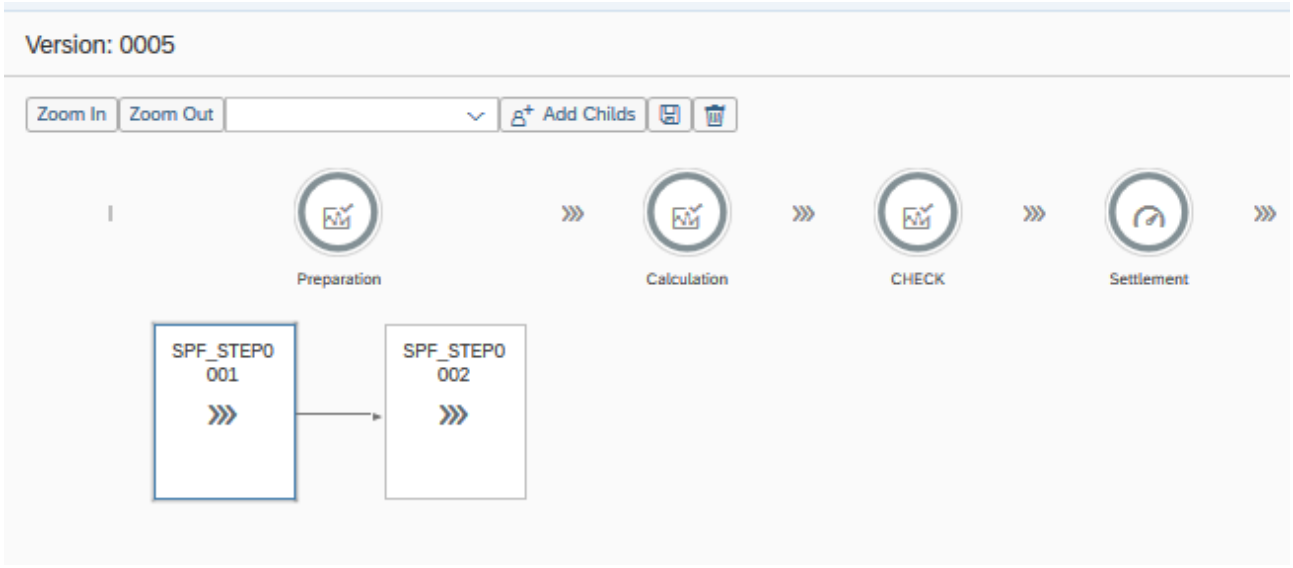


Figure 46: SPF: Configuration of Process via WebUI – 5 -

If an existing node is selected and a second new node is added then an additional child is added. The following figure shows the second node within another lane:

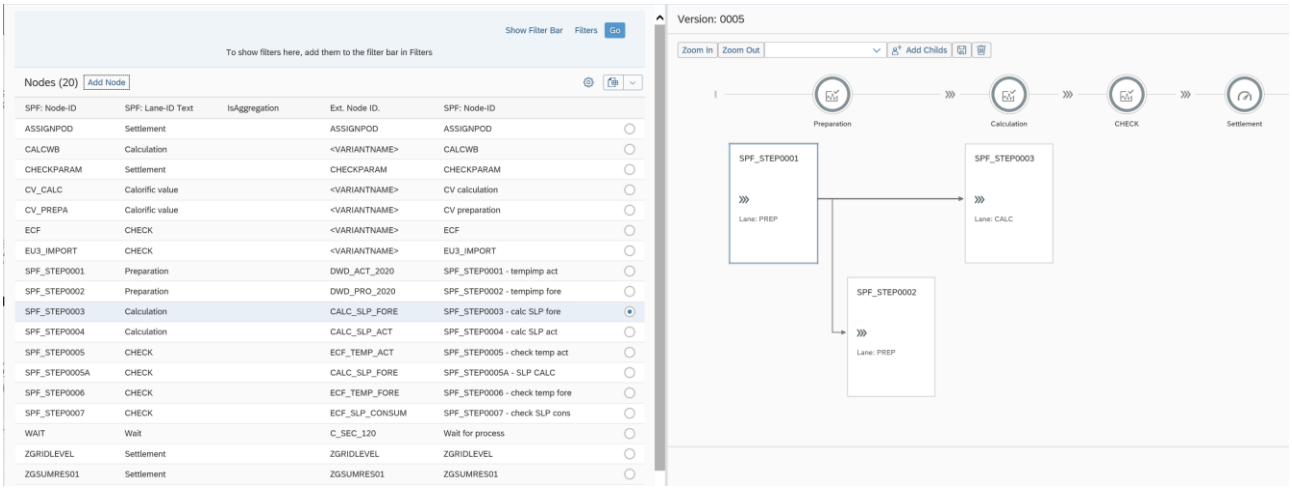


Figure 47: SPF: Configuration of Process via WebUI – 6 -

This configuration of the process can be done until the process is final maintained. For example:

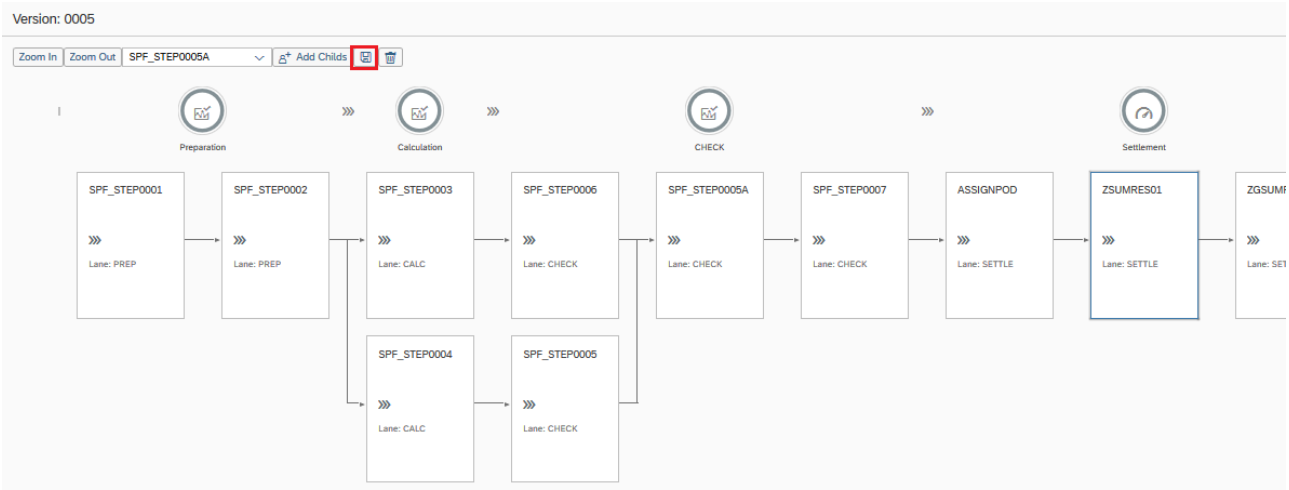


Figure 48: SPF: Configuration of Process via WebUI – 7 -

To save the process the "SAVE" - button must be pressed.

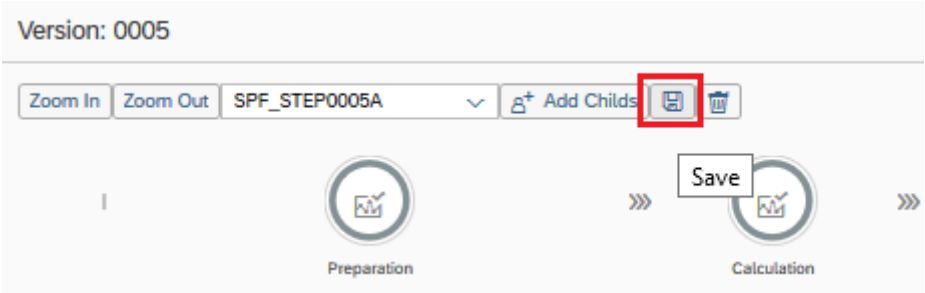


Figure 49: SPF: Configuration of Process via WebUI – 8 -

Afterward the following option must be chosen:

- "Create new Version" – to save as new version
- "Cancel" – to abort the saving process

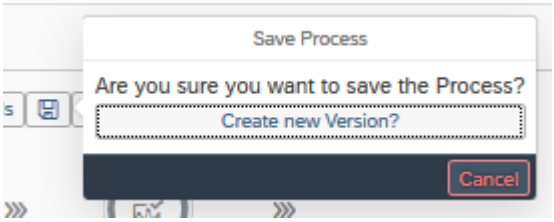


Figure 50: SPF: Configuration of Process via WebUI – 9 -

If "Create new Version" is selected, then additional information must be entered:

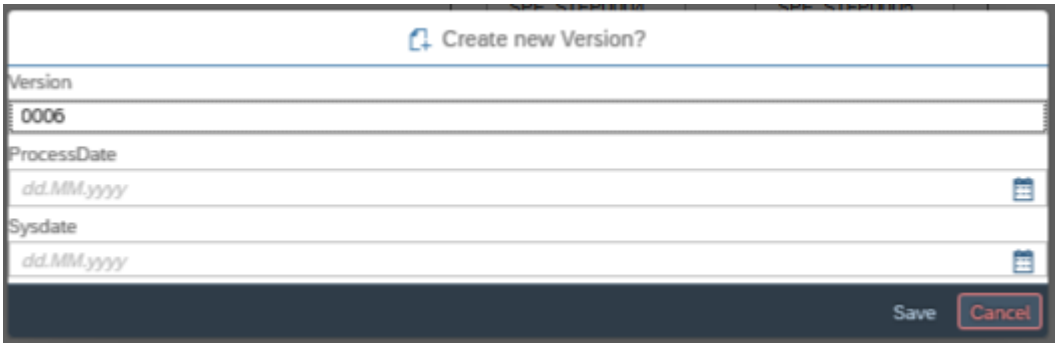


Figure 51: SPF: Configuration of Process via WebUI – 10 -

Here the following information must be entered:

- Version – automatically filled with next higher version number
- ProcessDate – enter the valid date for the process date
- Sysdate – enter the system date

Note: the new version is saved. Please go back to the previews page to update the view.

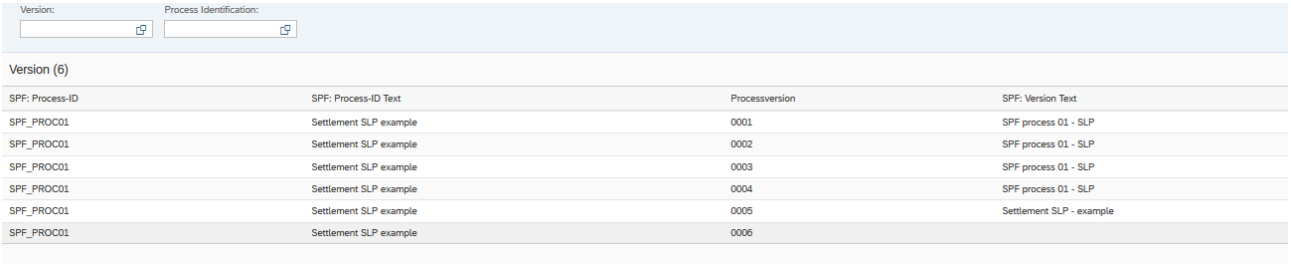


Figure 52: SPF: Configuration of Process via WebUI – 11 -

The maintenance of the SPF: Version Text is only possible within the SAP GUI. This is necessary due to the fact, that the configuration is a transportable object, which has to be included into a request.

This can be done via the standard transport table view of the maintenance view described in Maintaining Version.

Adding an independent Node:

It is possible to add a node without a direct parent relationship. For this only select the new node which should be added and press "Add Node".

Note: only the first node in the process shall not have parents. All other node must have minimal one parent relationship.

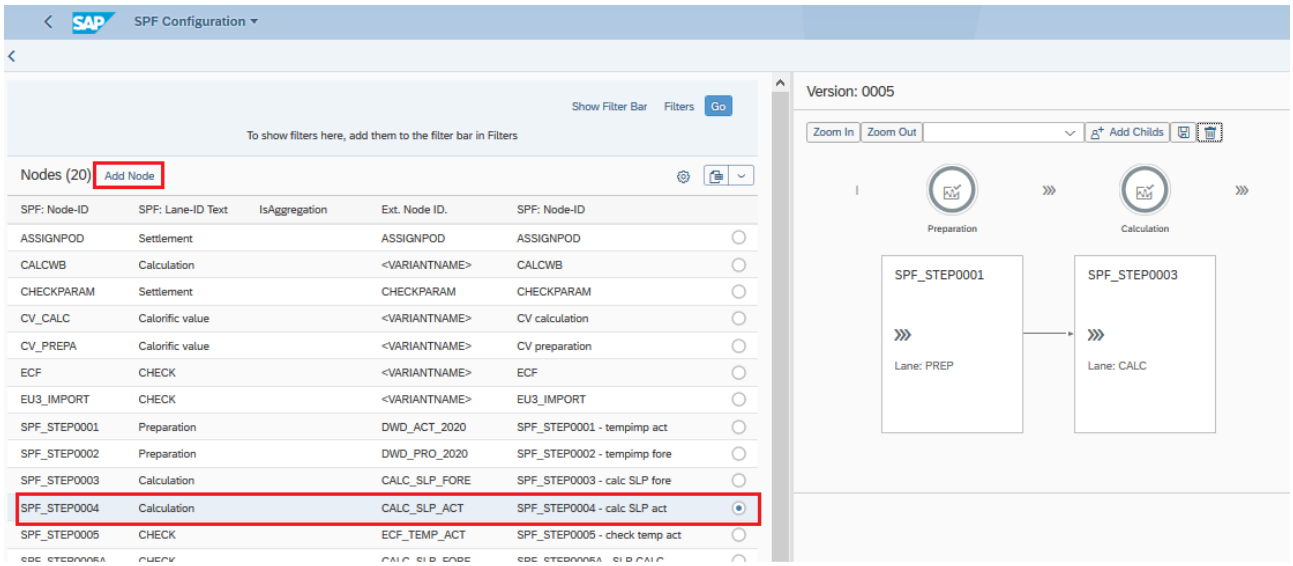


Figure 53: SPF: Configuration of Process via WebUI – 12 -

Due to the limitations of the currently picked fiori element, it is mandatory necessary that this node is edited afterwards and maintained to a parent. For this please use the "Add Childs" functionality.

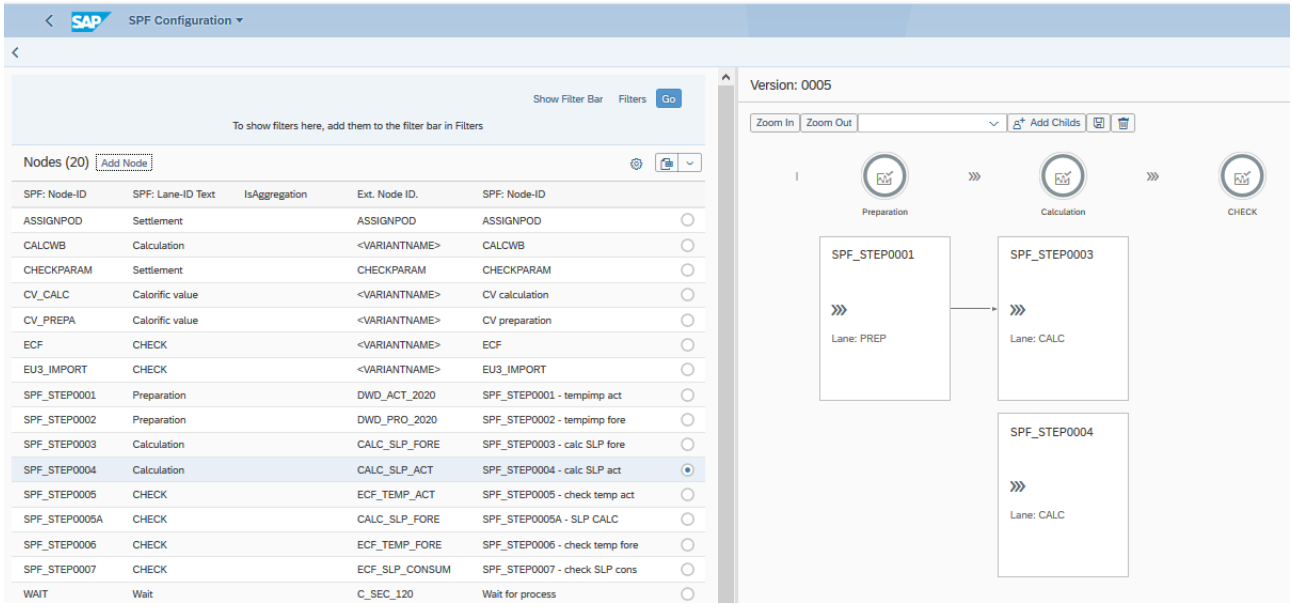


Figure 54: SPF: Configuration of Process via WebUI – 13 -

Add Childs:

Figure 55: SPF: Configuration of Process via WebUI – 14 -

With these functions it is possible to add additional parent to a node. To maintain a node as parent, the following steps are necessary:

- Select Parent Node to whom the node should be added

- Select Child Node in Windows
- Push the button “Add Childs”

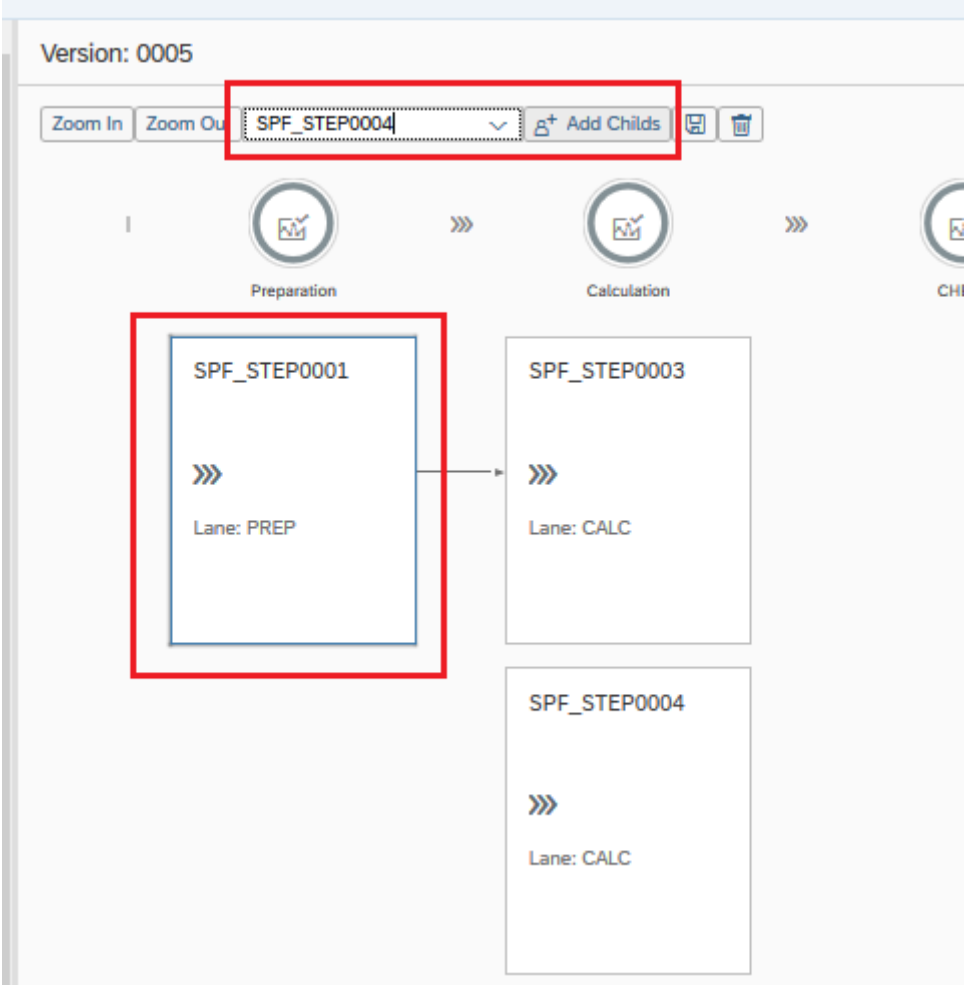


Figure 56: SPF: Configuration of Process via WebUI – 15 -

The steps described above produce the following result.

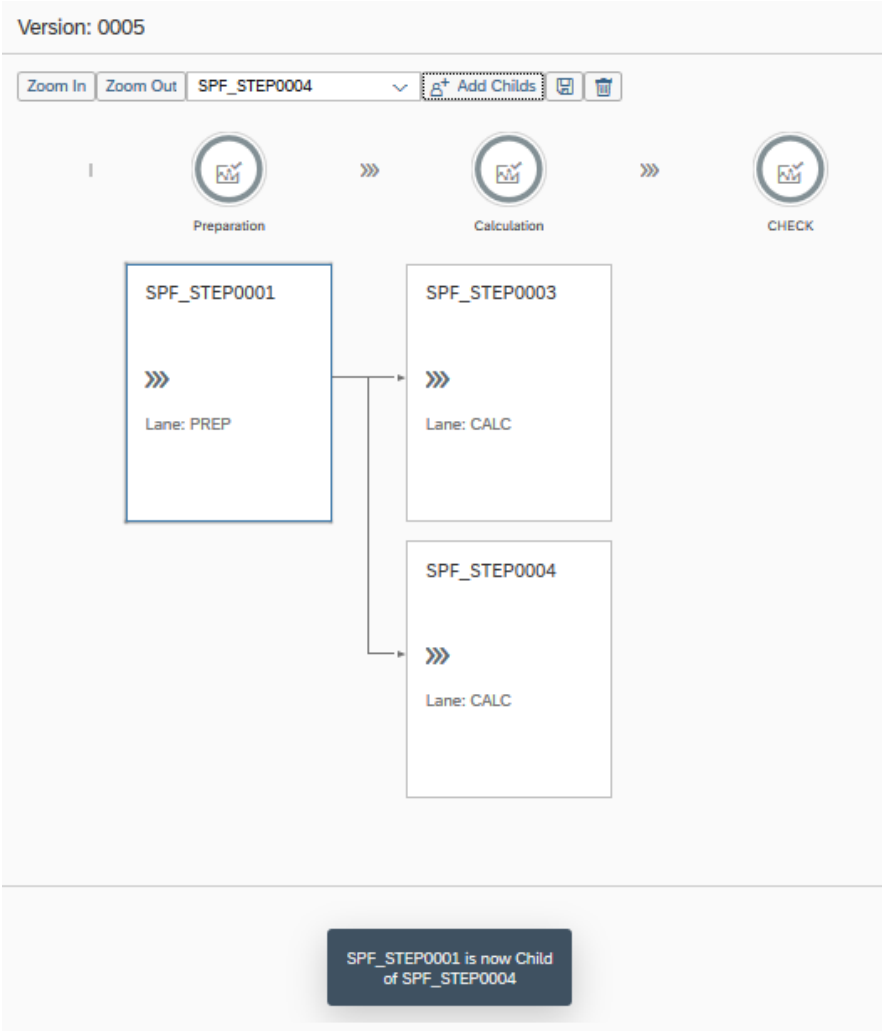


Figure 57: SPF: Configuration of Process via WebUI – 16 -

Delete an existing node:

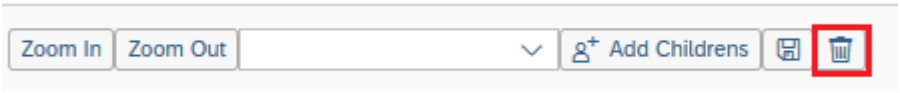


Figure 58: SPF: Configuration of Process via WebUI – 17 -

To delete an existing node, the following steps are necessary:

- Select the node to delete
- Press the Delete button

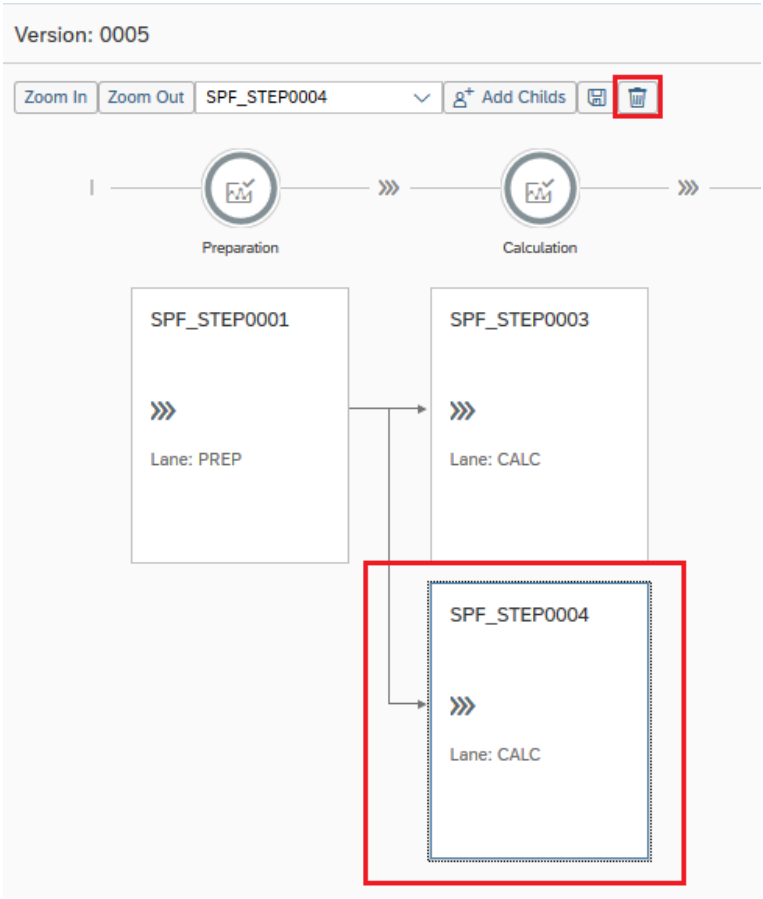


Figure 59: SPF: Configuration of Process via WebUI – 18 -

Note: if a node is deleted, which has parent and child relationships then it is afterwards necessary that the relationship between the existing nodes are checked and if necessary is maintained using "Add Childs" functionality.

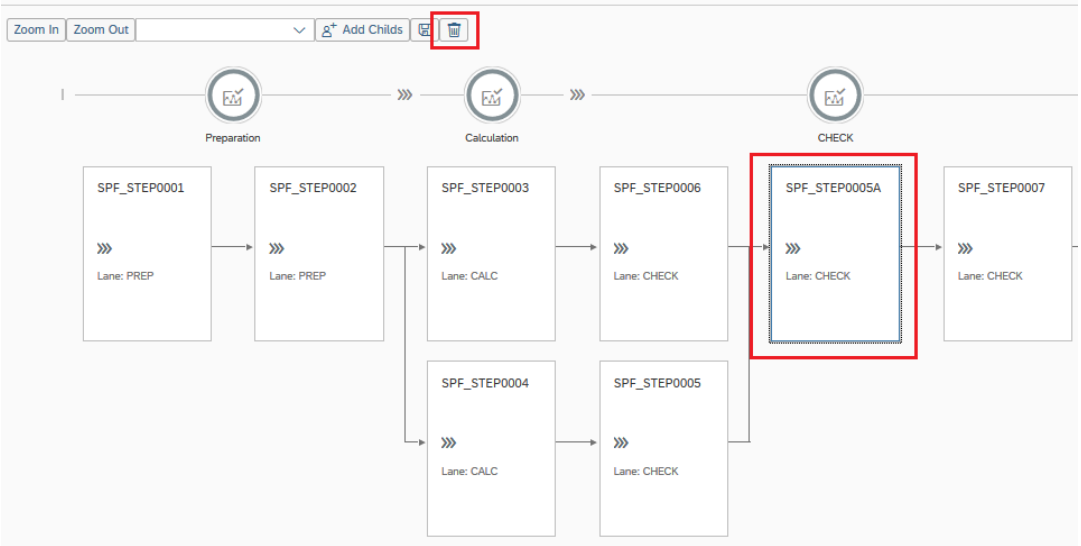


Figure 60: SPF: Configuration of Process via WebUI – 19 -

The following example is a result after deleting a node with child relationships. Here it is necessary that a new relationship is maintained.

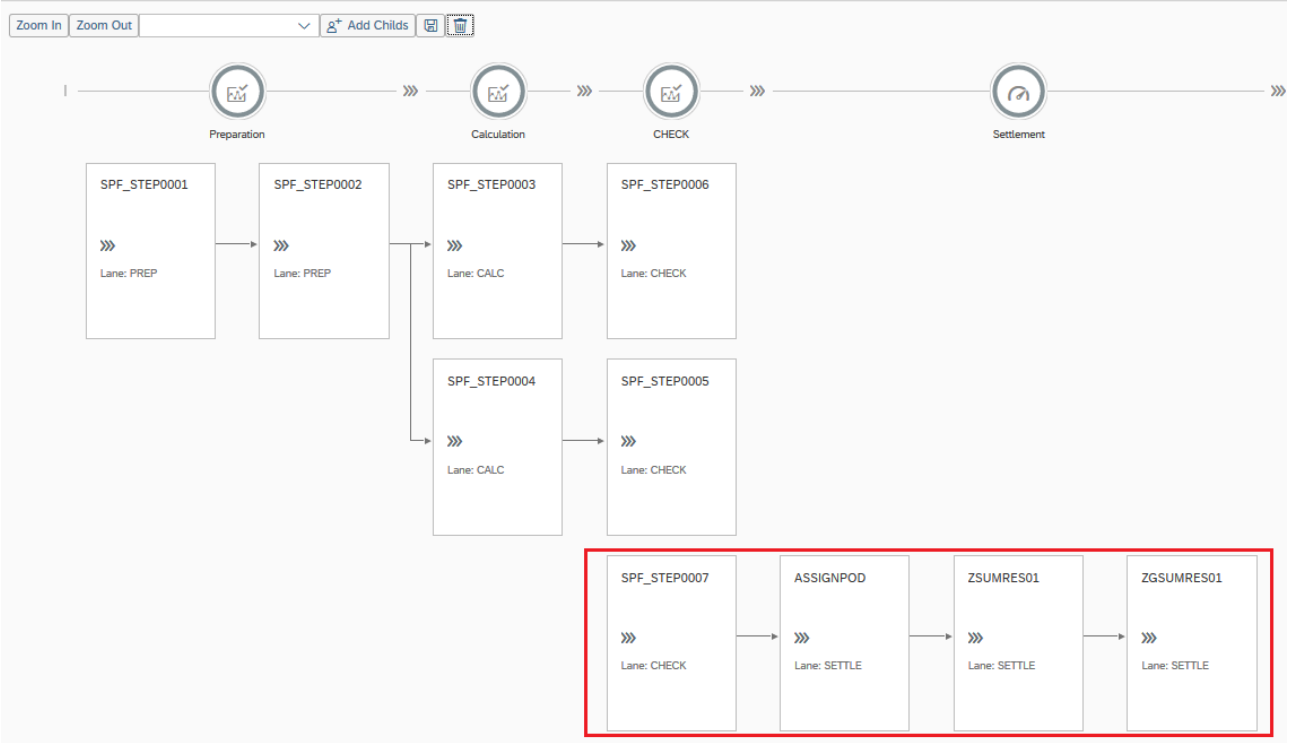


Figure 61: SPF: Configuration of Process via WebUI – 20 -

The current customization model supports extended configuration like:

- "Lane Hopping" – adding nodes to lanes of previous, "finished" lane. According to the example above: Adding a preparation node after processing "Check" or "Settlment" lane.
- Complete separated top-down-processes (multiple start nodes and multiple end nodes)

These configurations will not be shown properly within the UI5 application but can be processed correctly within the backend. In worst cased, the UI5 application is not able to edit these configurations. The monitoring and displaying of such configuration is still possible (but it is not "nice").

The calculation of the correct (currently sequential) order of the nodes is driven by its parent information. The current algorithm ensures, that every parent was processed before the child instance is executed.

2.2 Extension for EDM calculation workbench

The extension to the EDM calculation workbench (e.g. *EEDMCALCWB*) adds the Report /PLGA/RP_EDMCA_AUTOCALC to support background processing of formula IDs for specific time-ranges, disregarding the calculation triggers.

The Report also unlock the calculation automatically.

2.2.1 Authorization checks

The Report checks the following object after selection:

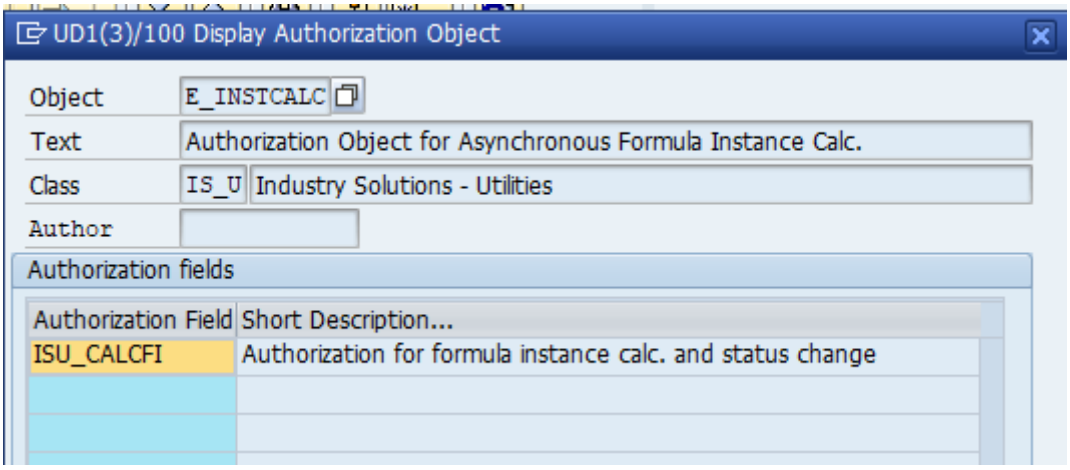


Figure 62: authorization object AUTOCALC

The field must have the following value:

CALC Asynchronous calculation of formula allocation is allowed

When selecting formula groups, the user additionally needs the field:

CALCF Calculation of Formula Groups is Allowed

2.2.2 Enhancement spot /PLGA/EDMCA_ES_PROCESS

The enhancement spot consists of one BAdI definition to adjust processing within the extension. It is described in the following chapter.

2.2.2.1 BAdI /PLGA/EDMCA_BD_FORMULAPROC

The interface provides two methods to adjust the calculation items and to enhance the process after the calculation.

Interface: /PLGA/IF_EDMCA_BD_FORMULAPROC

Default implementation: /PLGA/CL_EDMCA_BD_FORMULAPROC

Method: /PLGA/IF_EDMCA_BD_FORMULAPROC~PRE_PROCESSING_CALCULATION

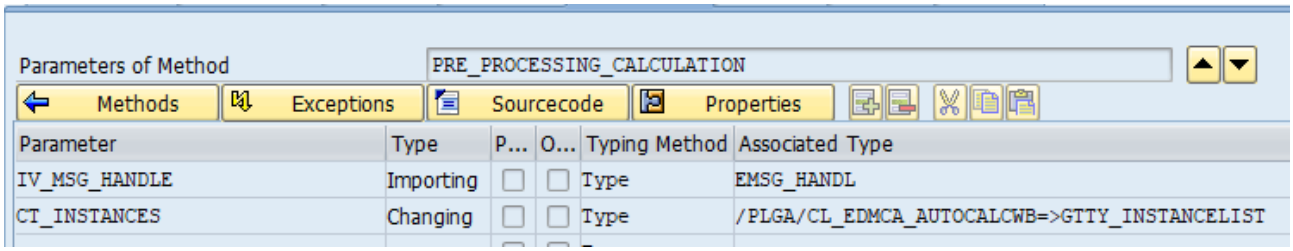


Figure 63: Signature of PRE_PROCESSING_CALCULATION

This method is called before looping on all formula IDs set via selection screen.
The default implementation alters the calculation time-range to the completeness of all input parameters.

Method: /PLGA/IF_EDMCA_BD_FORMULAPROC~ POST_PROCESSING_CALCULATION

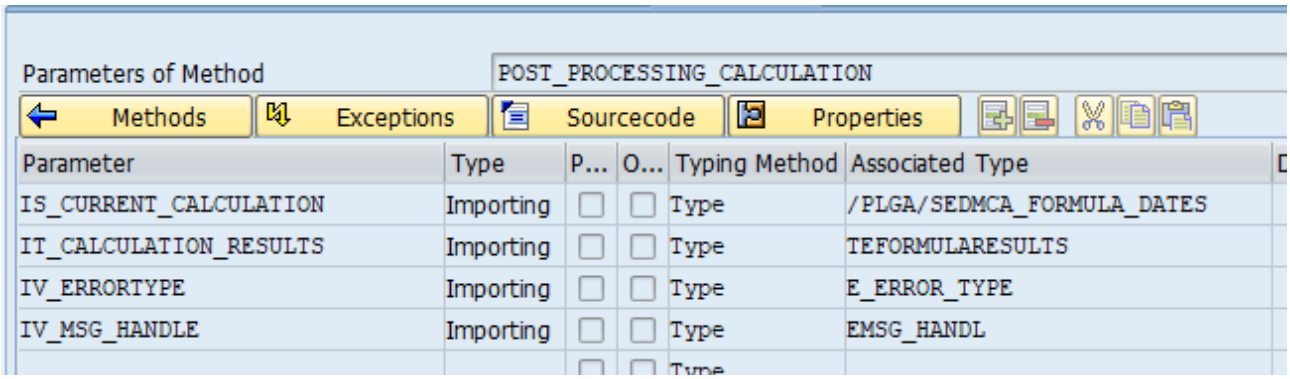


Figure 64: Signature of POST_PROCESSING_CALCULATION

This method is called after each calculation.
The default implementation does nothing.

To create an own implementation of the BAdI you can go to *SPRO -> SAP Customizing Implementation Guide -> Extension for Utilities by PROLOGA -> Energy Data Management -> Extension for EDM calculation workbench -> BAdI for Formula processing*

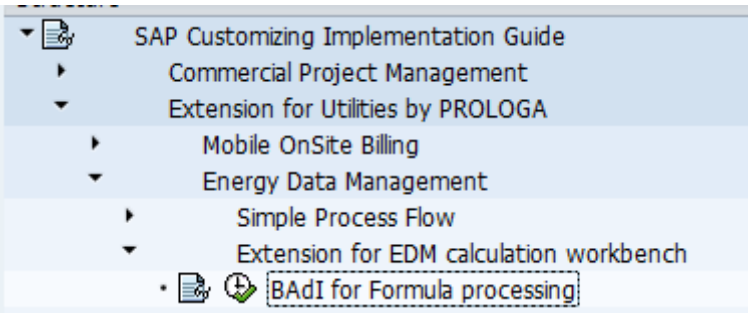


Figure 65: EDM Calculation Extension: SPRO-path for BAdI implementation

2.2.3 Formulas

The following formulas were added to extend the calculation capabilities of the EDM calculation workbench.

2.2.3.1 Amount-based average calculation

Function module: /PLGA/EDMCA_AVG_AMOUNT.

This function calculated the amount-based average value for the calculation period. The second output parameter sums all input amounts.

With the transaction *EEDM_FRM01* an equivalent formula can be created according to the following information:

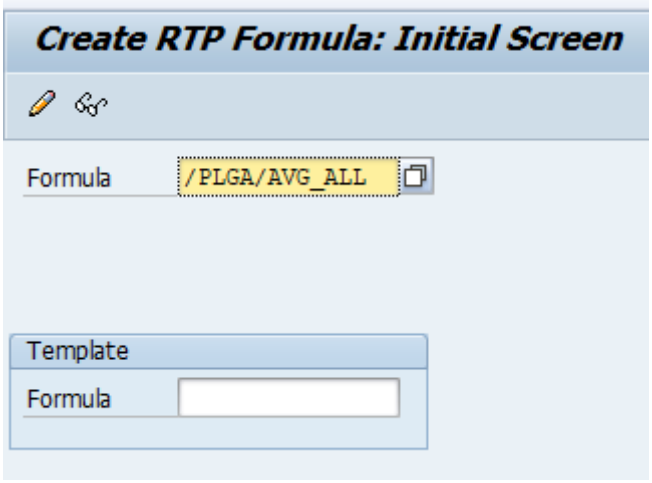


Figure 66: Creating formula /PLGA/AVG_ALL

The following figure shows the parameter setting:

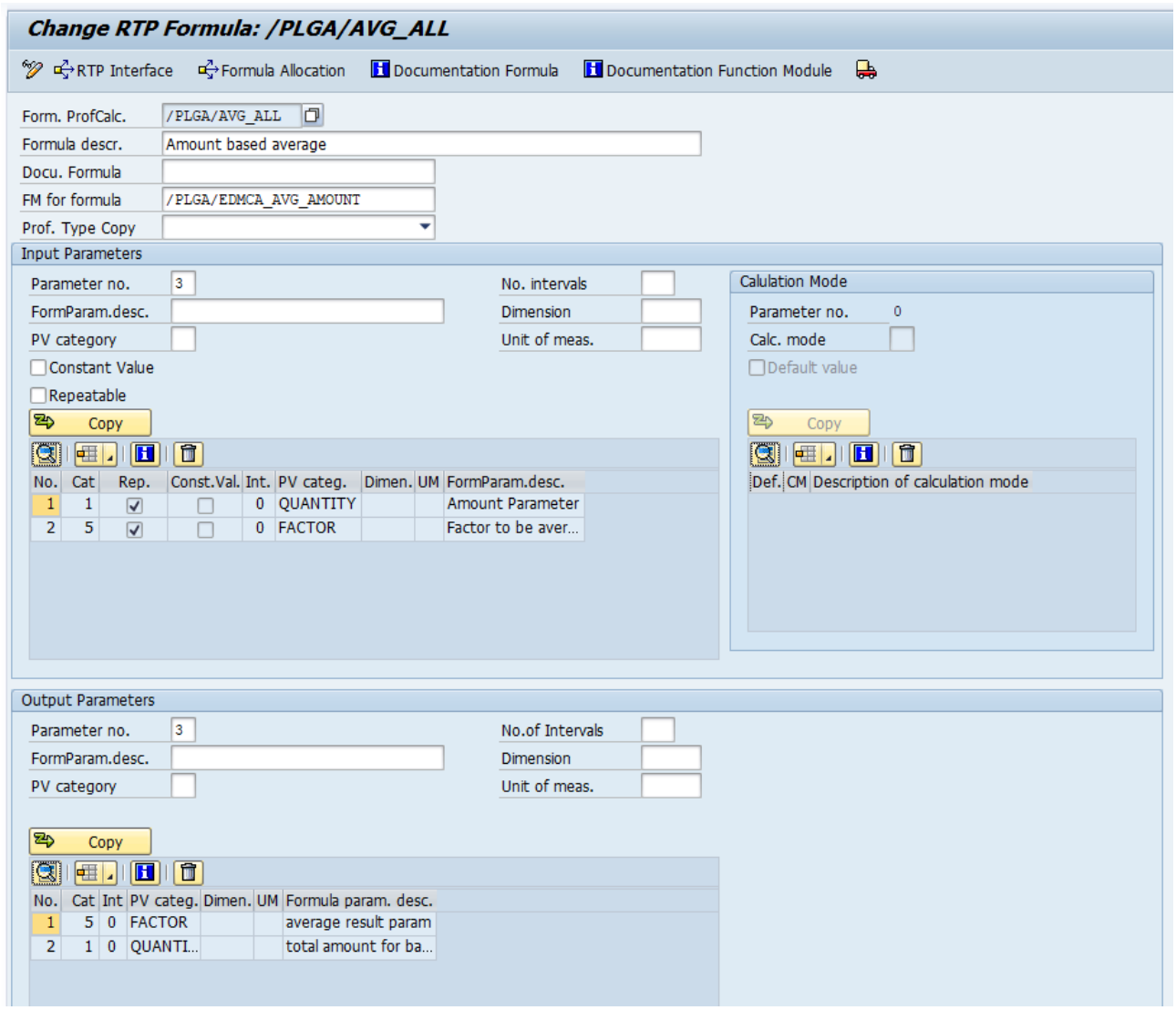


Figure 67: Parameter of /PLGA/AVG_ALL

Both input parameters are repeatable. The usage of specific profile value categories is exemplarily. These only influence preaggregation features prepended by the calculation workbench from EDM.

2.2.4 Simple Process Flow connector

The extension to the EDM calculation workbench can be integrated into the Simple Process Flow as a stand-alone node.

Each node represents a selection variant for the report /PLGA/RP_EDMCA_AUTOCALC:

The following parameters shall be entered within the SPF node maintenance:

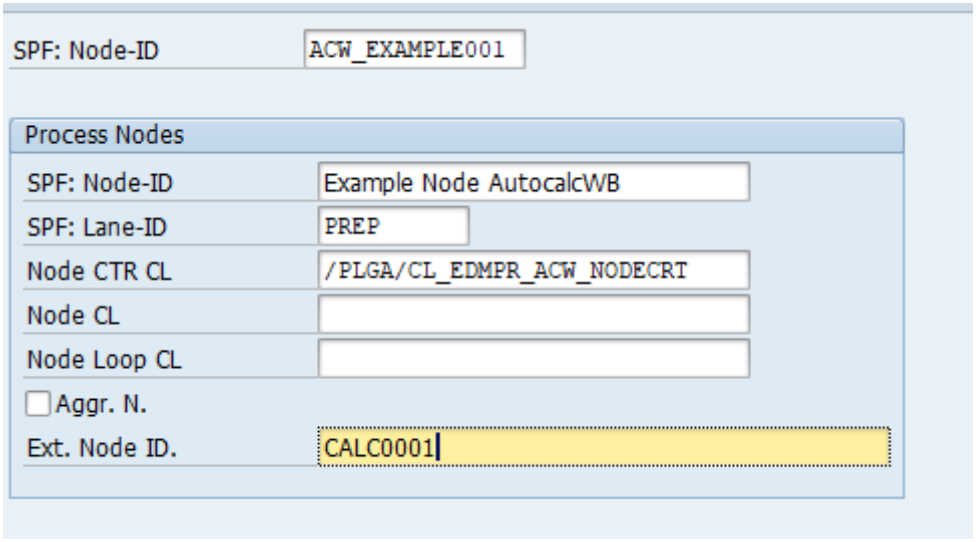


Figure 68: Example Node for ACW

Notes on Parameters:

Lane-ID: The selected lane identification should be compatible to stand-alone nodes (e.g. lanes with Lane-Class */PLGA/CL_EDMPR_NOOP_LANECTR*)

The node control class is */PLGA/CL_EDMPR_ACW_NODECRT*

The External Node identification shall contain the technical name of a variant of the report /PLGA/RP_EDMCA_AUTOCALC.

2.3 H-Values Calculation for the Processing of Standard Load Profiles (SLP) Gas

The customizing for SLP consists of the customizing of the SLP type and the customizing of a formula.

2.3.1 Customizing SLP Types – TU Munich (TUM)

The customizing for the SLP will be started within the SAP® Reference IMG by using transaction *SPRO.* The maintenance for SLP is done under the path: *SAP Customizing Implementation Guide -> Extension for Utilities by Prologa -> Energy Data Management -> H-Values Calculation for the Processing of Standard Load Profiles (SLP)Gas -> TUM: Synthetic Loadprofile configuration*

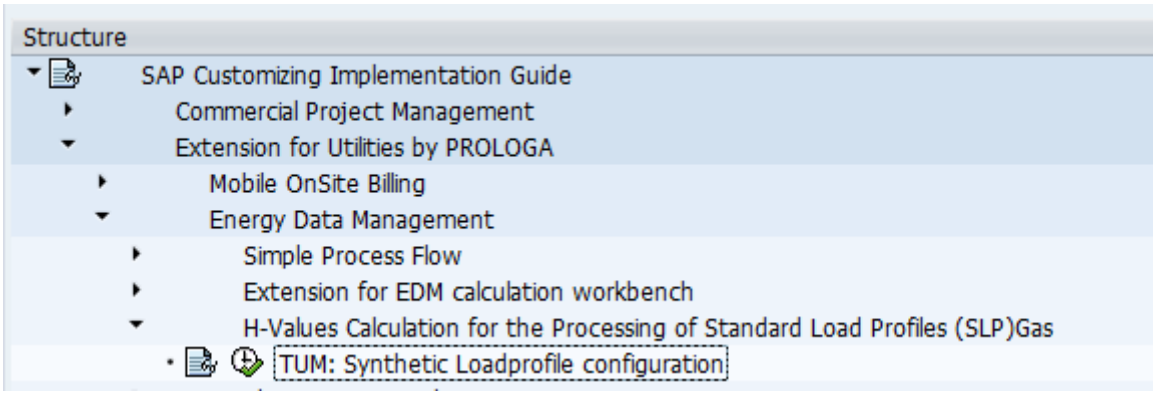


Figure 69: Customizing path for Extension for Utilities by Prologa

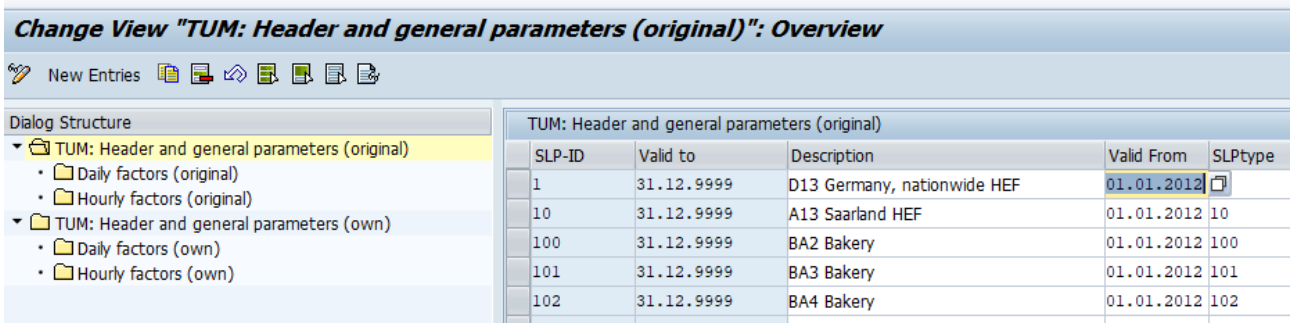


Figure 70: Customizing example – TUM SLP types

Relevant session contents for the SLP customizing are configurable in the following order:

1. SLP GAS Customer Types
2. TUM SLP Daily Factors
3. TUM SLP Hourly Factors

2.3.1.1 TUM: Header and general parameters

Table*: /PLGA/TTEDMTUMH*

Text table: */PLGA/TTEDMTUMHT*

Description: The table */PLGA/TTEDMTUMH* contains the TUM coefficients A, B, C, D per SLP type with an accuracy of nine decimal places as well as the specific SLP coefficients for SIGLINDE. They are used to forecast the consumption with the help of SLP dependent to the outdoor temperature.

Figure 71: Customizing Example: SLP Gas Customer Types

Column	Data Element	Description
SLP ID	/PLGA/DEDM_TUM_ID	Internal ID for SLP type
Valid to	BISZEITSCH	Date up to which a time slice is valid
Description	BU_BEZ50	Name
Valid From	ABZEITSCH	Date from which a time slice is valid
SLPtype	/PLGA/DEDM_CUST_CAT	Customergroup gas SLP
External Description	/PLGA/DEDM_CUST_CAT_EXT	External description of the SLP profile
Calendar ID	WFCID	Factory Calendar
TUM coefficients	/PLGA/DEDM_TUM_COEFF9	Coefficients TU Munich (9 post decimal posts) (A)
TUM coefficients	/PLGA/DEDM_TUM_COEFF9	Coefficients TU Munich (9 post decimal posts) (B)
TUM coefficients	/PLGA/DEDM_TUM_COEFF9	Coefficients TU Munich (9 post decimal posts) (C)
TUM coefficients	/PLGA/DEDM_TUM_COEFF9	Coefficients TU Munich (9 post decimal posts) (D)
FfE coefficients SLP GAS	/PLGA/DEDM_FFE_COEFF9	Coefficients FfE (9 post decimal posts) (MH)
FfE coefficients SLP GAS	/PLGA/DEDM_FFE_COEFF9	Coefficients FfE (9 post decimal posts) (BH)
FfE coefficients SLP GAS	/PLGA/DEDM_FFE_COEFF9	Coefficients FfE (9 post decimal posts) (MW)
FfE coefficients SLP GAS	/PLGA/DEDM_FFE_COEFF9	Coefficients FfE (9 post decimal posts) (BW)
Reference Temperature	/PLGA/DEDM_TUM_DEVIATION	Reference temperature
Daily Factor	/PLGA/DEDM_TUM_FLAG_DAILY	TUM flag if daily factors exist

Table 11: SLP Gas Customer Types (9)

2.3.1.2 TUM SLP Daily Factors

Table*: /PLGA/TTEDMTUMD*

Description: In table */PLGA/TTEDMTUMD* it is determined how much the single weekdays influence the calculated SLP value. To maintain the daily factors first the corresponding SLP-ID has to be selected.

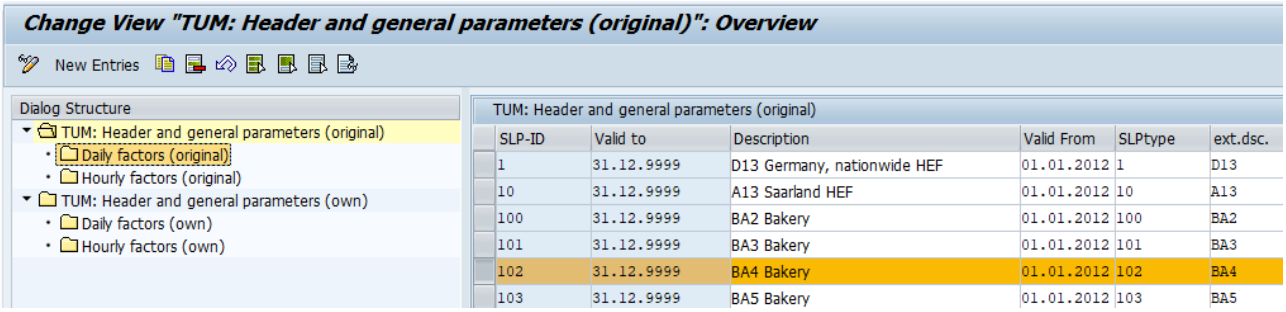


Figure 72: Customizing example: select SLP type

The Daily factors should be maintained afterwards. For this it is necessary to click twice on "Daily factors" in the dialog structure. Furthermore, it is possible to maintain the daily factors for the SLP type.

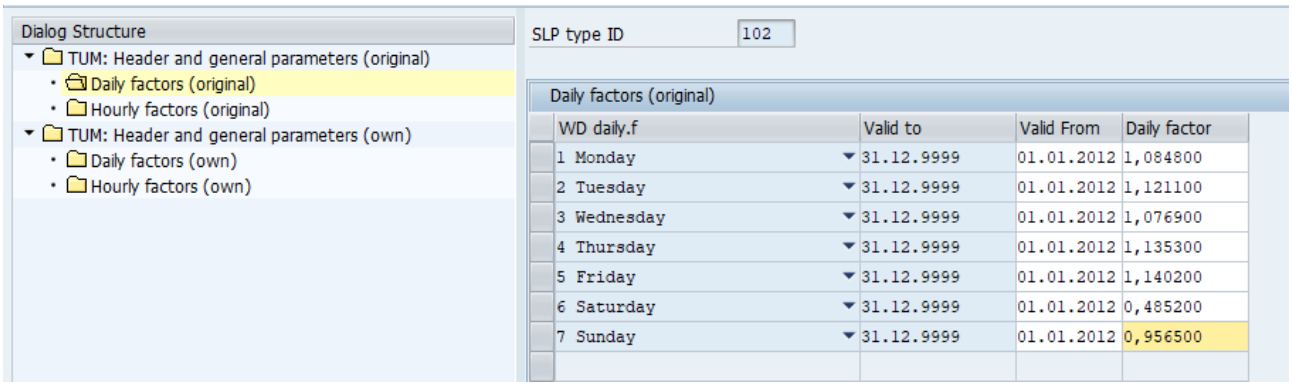


Figure 73: Customizing example: TUM SLP Daily Factors

Column	Data Element	Description
SLP ID	/PLGA/DEDM_TUM_ID	Customer category gas SLP
Wd daily. fac	/PLGA/DEDM_DAILY_DEPENDENCY	Weekdays for daily factors gas slp
Valid to	BISZEITSCH	Date up to which a time slice is valid
Valid from	ABZEITSCH	Date from which a time slice is valid
Daily factors	/PLGA/DEDM_TUM_FACTOR_DAILY	Factor for daily usage factor of tum profiles

Table 12: TUM SLP Daily Factors

For the field "Wd da. Fac" counts: Integer number between 0 and 7:

- 0 stands for an equal weekly allocation. Just one entry for the SLP type will be made then.
- 1 = Monday
- 2 = Tuesday
- 3 = Wednesday
- 4 = Thursday
- 5 = Friday
- 6 = Saturday
- 7 = Sunday

The numbers greater than 0 will be executed as one occurrence so that 7 rows will be created and the sum of the deposited daily factors should be 7. For the daily factors SLP counts: For the weekdays' factor 0 the number 1 will be entered, or otherwise the relative value.

2.3.1.3 TUM SLP Hourly Factors

Table*: /PLGA/TTEDMTUMFH*

Description: In table */PLGA/TTEDMTUMFH* it is determined how much the single hour influence the calculated SLP value. To maintain the hourly factors first the corresponding SLP-ID has to be selected.

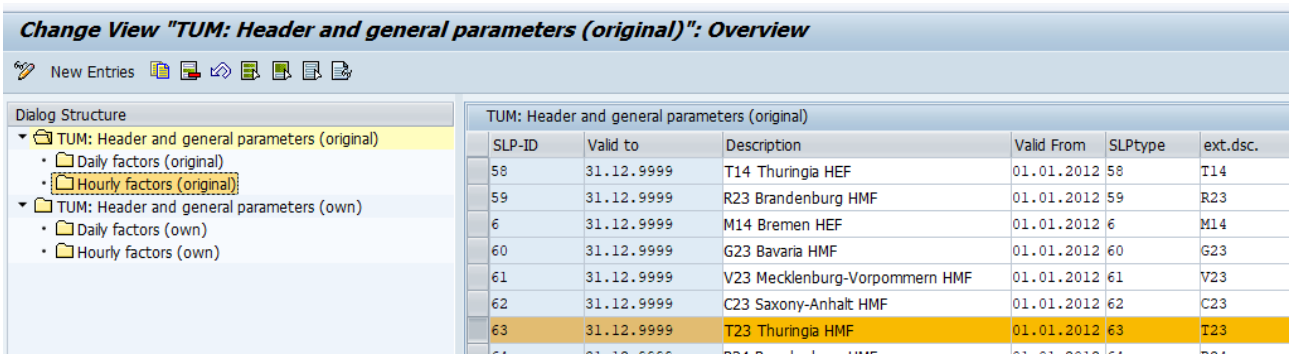


Figure 74: Customizing example: select SLP type

The Daily factors should be maintained afterwards. For this it is necessary to click twice on "Daily factors" in the dialog structure. Furthermore, it is possible to maintain the daily factors for the SLP type.

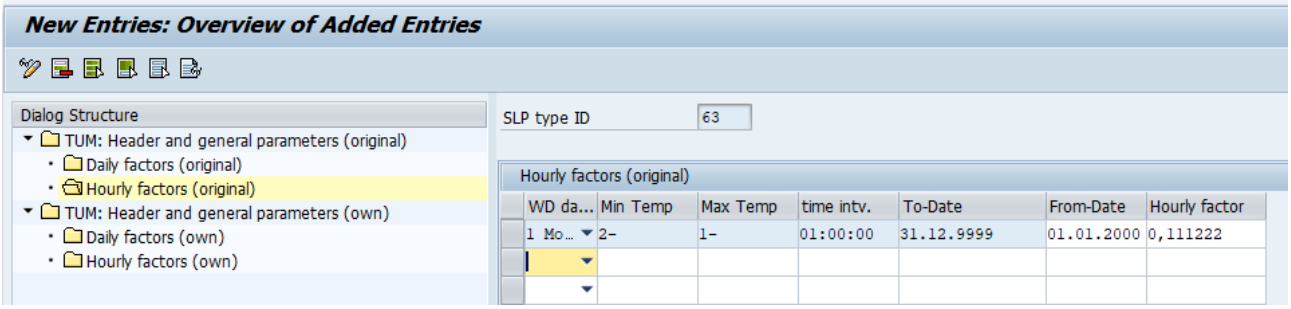


Figure 75: Customizing example: TUM SLP Daily Factors

Column	Data Element	Description
SLP ID	/PLGA/DEDM_TUM_ID	Internal ID of SLP type TUM
WD daily.f	/PLGA/DEDM_DAILY_DEPENDENCY	Weekdays for daily factors gas slp
Min Temp	/PLGA/DEDM_TEMP_INTERVAL_MIN	Temperature interval (minimum)
Max Temp	/PLGA/DEDM_TEMP_INTERVAL_MAX	Temperature interval (maximum)
Time intv.	/PLGA/DEDM_TIME_INTERVAL	beginning time interval for hourly factor gas slp
Valid to	E_EDMDATETO	To-Date
Valid from	E_EDMDATEFROM	From-Date
Hourly factors	/PLGA/DEDM_TUM_FACTOR_HOURLY	Factor for hourly usage factor of tum profiles

Table 13: TUM SLP Hourly Factors

2.3.2 Customizing of SLP formula

The H-Value calculation is based on the function module */PLGA/EDM_CALC_SLP_TUM* or /PLGA/EDM_CALC_SLP_HTWK for HTWK (Leipzig/Halle). For that function module, the following formulas will be created using transaction *EEDM_FRM01* and confirmed with ENTER:

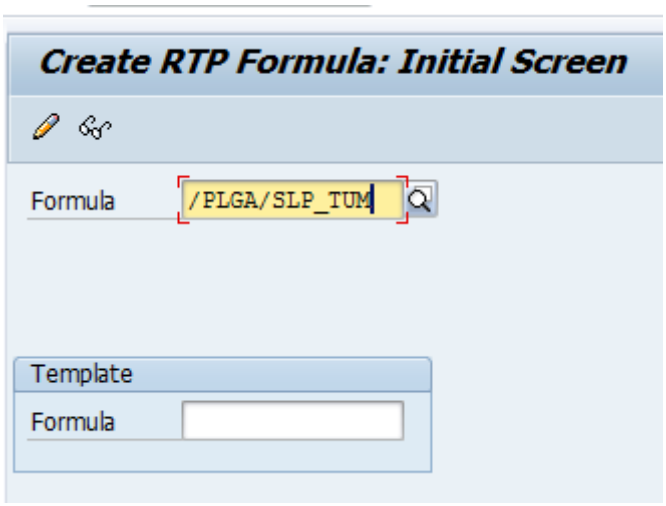


Figure 76: Create Formula for SLP Calculation (TU Munich)

To do that, the two input parameters

- SLP type and
- Temperature profile

must be entered and transferred according to previous figure. Each parameter will get the calculation mode according to the following figure.

Profile value types, dimensions and units have to be defined in the customizing settings.

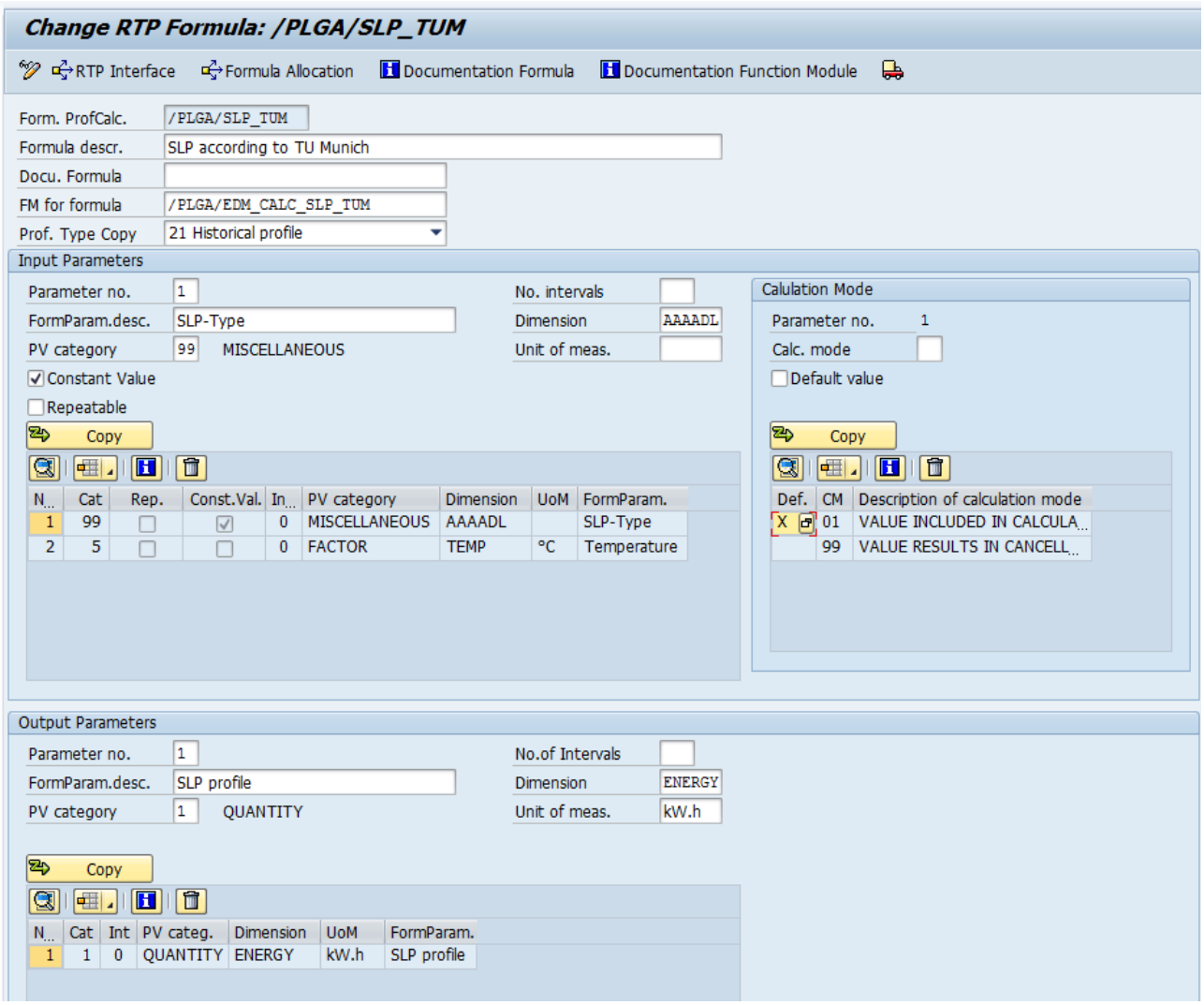


Figure 77: Formula Structure SLP Calculation

Furthermore, the output parameter

- Calculated results

must be entered and transferred according to the previous figure. After entering this parameter, the settings must be saved.

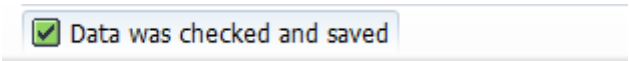


Figure 78: Saved Formula SLP Calculation

This formula can be used on application side afterwards.

Profile value types, dimensions and units have to be defined in the customizing settings.

Furthermore, the possibility to use within SLP calculation with hourly factors is given. For this please create the following formula. The H-Value calculation is based on the function module */PLGA/EDM_CALC_SLP_TUM*. For that function module, the following formulas will be created using transaction *EEDM_FRM01* and confirmed with ENTER:

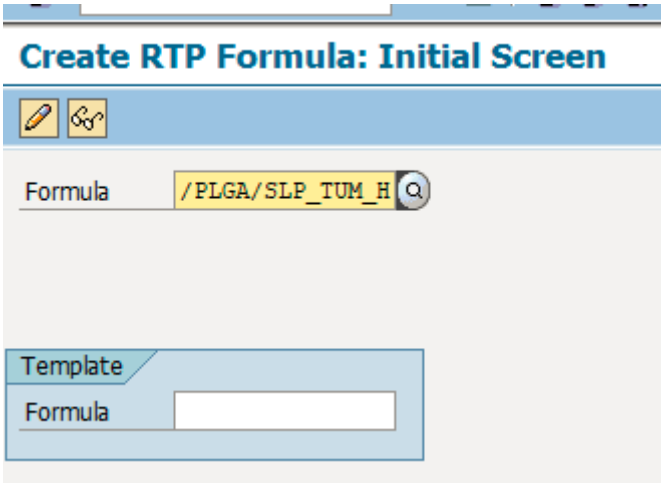


Figure 79: Create Formula for SLP Calculation hourly factor (TU Munich)

To do that, the two input parameters

- SLP type and
- Temperature profile
- Use hourly factors

must be entered and transferred according to following figure. Each parameter will get the calculation mode according to following figure.

Profile value types, dimensions and units have to be defined in the customizing settings.

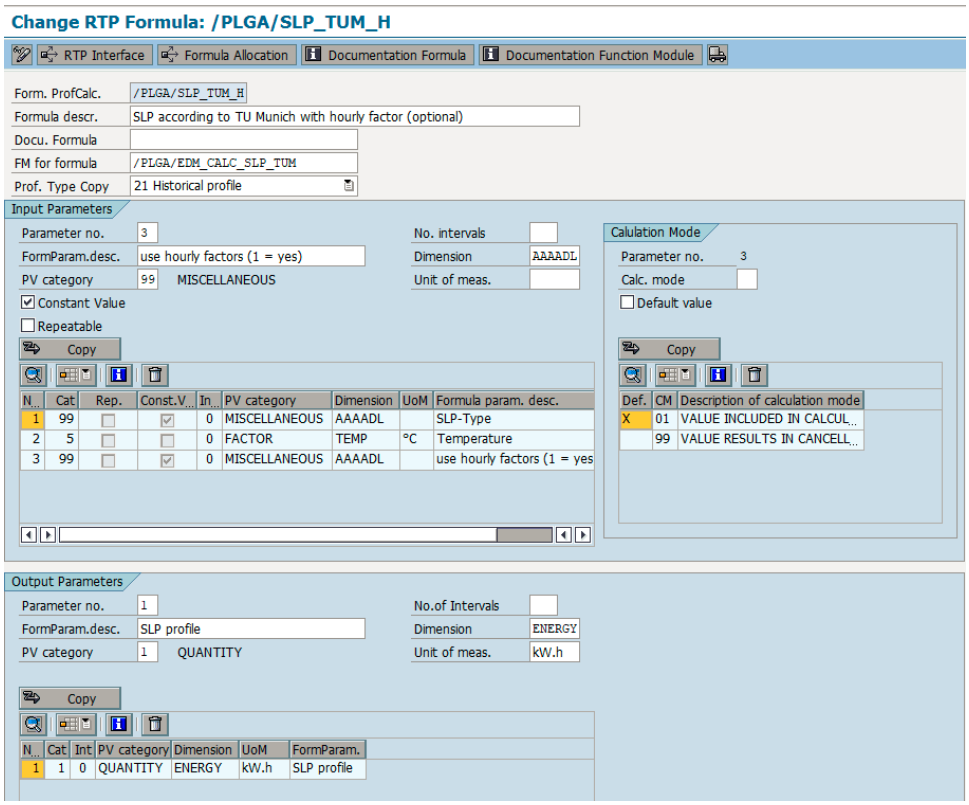


Figure 80: Formula Structure SLP Calculation

Furthermore, the output parameter

- Calculated results

must be entered and transferred and after entering this parameter, the settings must be saved.

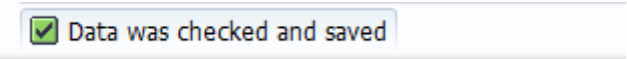


Figure 81: Saved Formula SLP Calculation

This formula can be used on application side afterwards.

2.4 Calorific value extension

The report /PLGA/RP_EDMCV_BILLINGDATA (transaction /PLGA/ECV_MAINTBILL) helps to fill billing master data from EDM profile values.

The Report /PLGA/RP_EDMCV_REFERENCEV (transaction /PLGA/ECV_REF_PV) copies values for settlement purpose from billing relevant profile according to the database table /PLGA/TECVPROF (Profile table for calorific value management, transaction /PLGA/ECV_PROF_ST).For further details, please consult the User documentation.

2.4.1 Authorization checks

The following authorization objects are checked in following reports:

Report: /PLGA/RP_EDMCV_REFERENCEV Calculate reference factor profile value

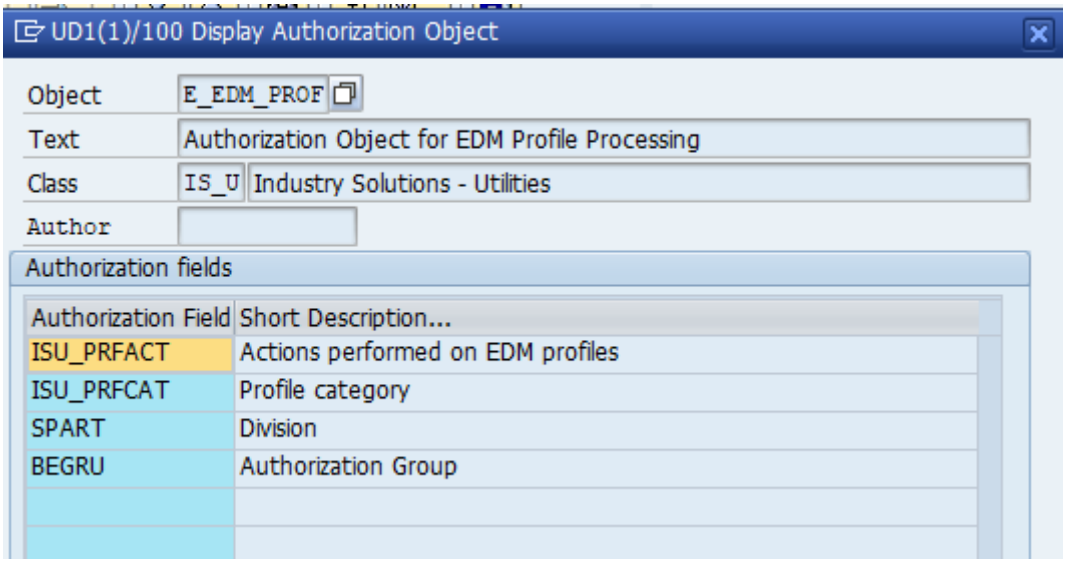


Figure 82: Authorization check /PLGA/RP_EDMCV_REFERENCEV

ID: ISU_PRFACT

Field: 06 (Display profile values) and 05 (change profile values)

Report: /PLGA/RP_EDMCV_BILLINGDATA Maintaining billing relevant tables from EDM

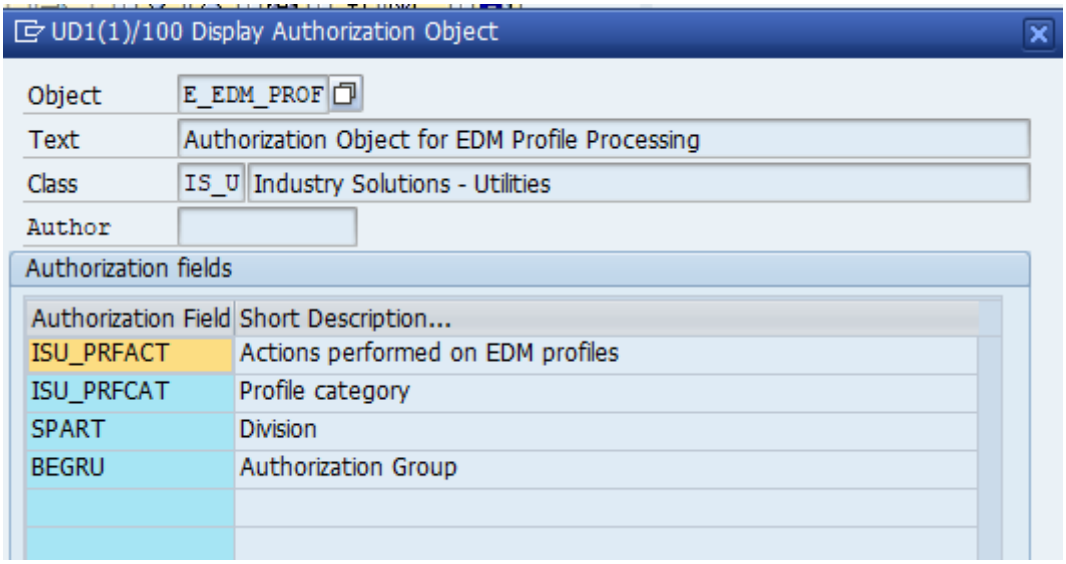


Figure 83: Authorization check /PLGA/RP_EDMCV_BILLINGDATA

ID: ISU_PRFACT
Field: 06 (Display profile values)

Additionally, the S_TCODE-Object will be checked for *S_KK4_74000852* (Maintaining Billing Calorific Values)

2.4.2 Formula /PLGA/EDMCV_AB_AVERAGE

Function module: /PLGA/EDMCV_AB_AVERAGE.

This function calculated the amount-based average calorific value for billing purpose according the regulatory document G685. The second output parameter sums all input amounts.

With the transaction *EEDM_FRM01* an equivalent formula can be created according to the following information:

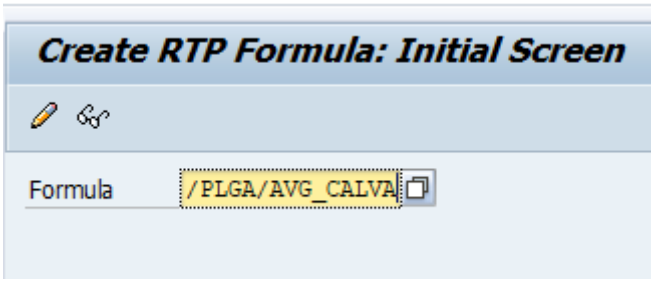


Figure 84: create example formula calorific amount-based average

The following figure shows the parameter exemplarily:

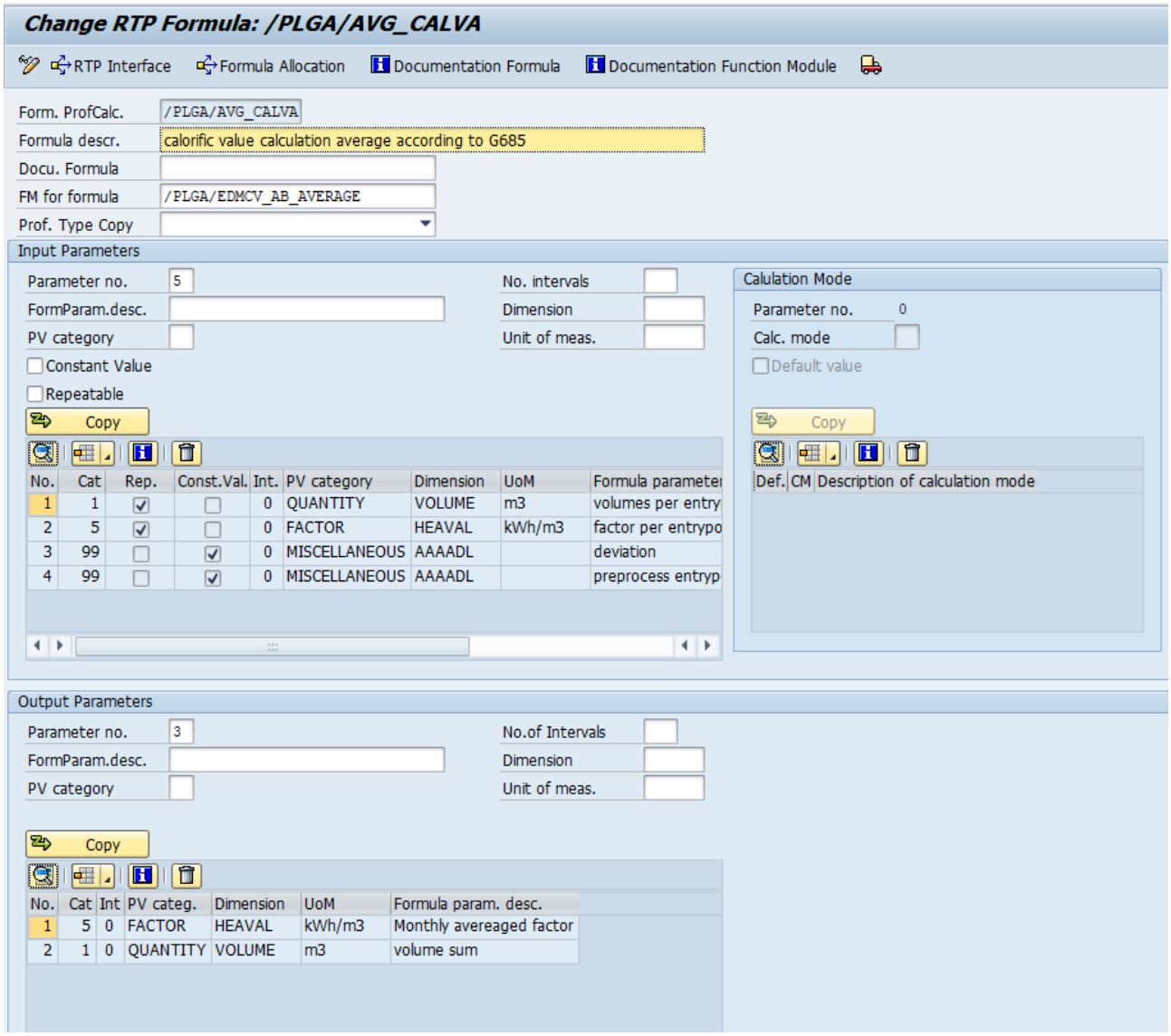


Figure 85: Parameters of /PLGA/EDMCV_AB_AVERAGE

Inputs:

First parameter: Quantity, repeatable

Second parameter: Factor, repeatable

Third parameter: Deviation value. Each input pair's average factor is checked versus the final result to this deviation. It has to be identical in each month of calculation

Fourth parameter: "bool" value to enable pre-calculation of the input pairs. This "flag" supports the usage of this formula for amount-based average calculation, when every input factor already averaged by the pre-processing system.

Outputs:

First parameter: Factor, averaged for each month. Calculation has to be done for full months. Due to regulations the output profile must be of an hourly interval.

Second parameter: Optionally sums up all amounts of all input pairs.

2.4.3 Simple Process Flow connector

Both report-based functions can be integrated into the Simple Process Flow as stand-alone nodes. The configuration according to chapter Maintaining Nodes has to be done with the following parameters:

Function: Maintain billing master data from EDM profiles

Example:

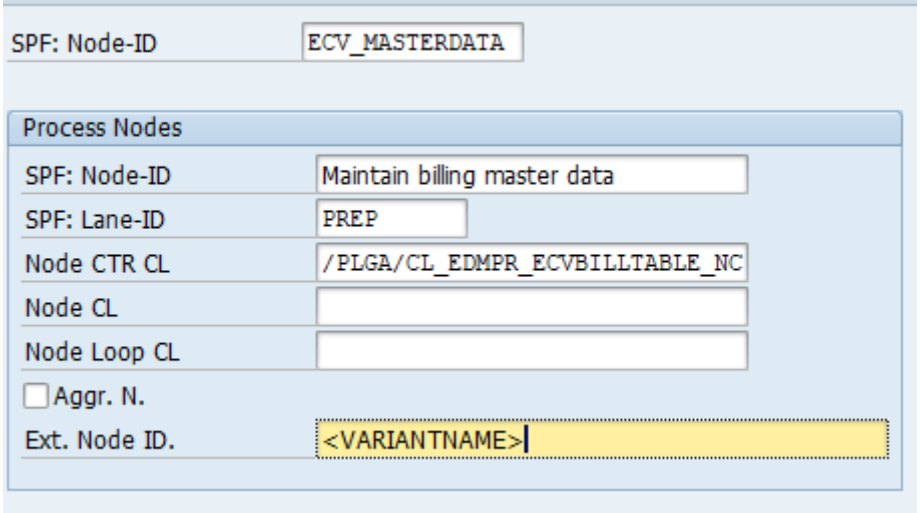


Figure 86: Example billing table maintenance with SPF

Parameter notes:

Lane-ID: The selected lane identification should be compatible to stand-alone nodes (e.g. lanes with Lane-Class */PLGA/CL_EDMPR_NOOP_LANECTR*)

The Node-Control-Class must be set to */PLGA/CL_EDMPR_ECVBILLTABLE_NC*

The External Node identification shall contain the technical name of a variant of the report /PLGA/RP_EDMCV_BILLINGDATA.

Function: Generating Reference calorific profile values

Example:

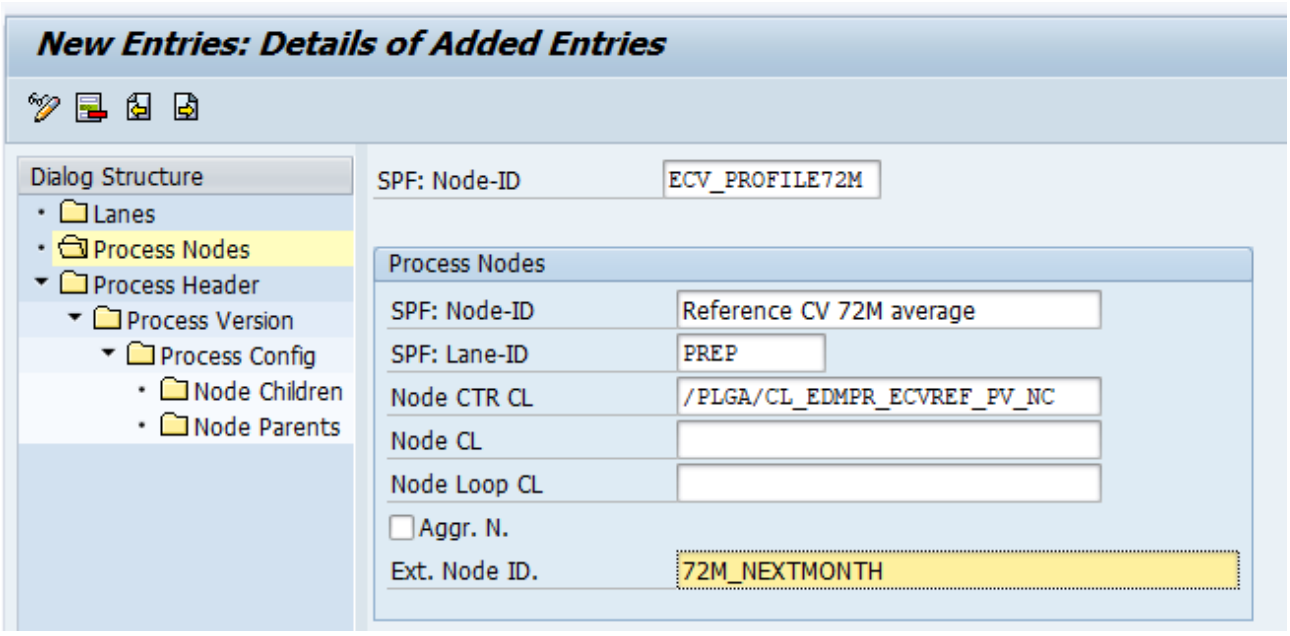


Figure 87: Example reference factor profile values with SPF

Parameter notes:

Lane-ID: The selected lane identification should be compatible to stand-alone nodes (e.g. lanes with Lane-Class */PLGA/CL_EDMPR_NOOP_LANECTR*)

The Node-Control-Class must be set to */PLGA/CL_EDMPR_ECVREF_PV_NC*

The External Node identification shall contain the technical name of a variant of the report /PLGA/RP_EDMCV_REFERENCEV.

Example variant:

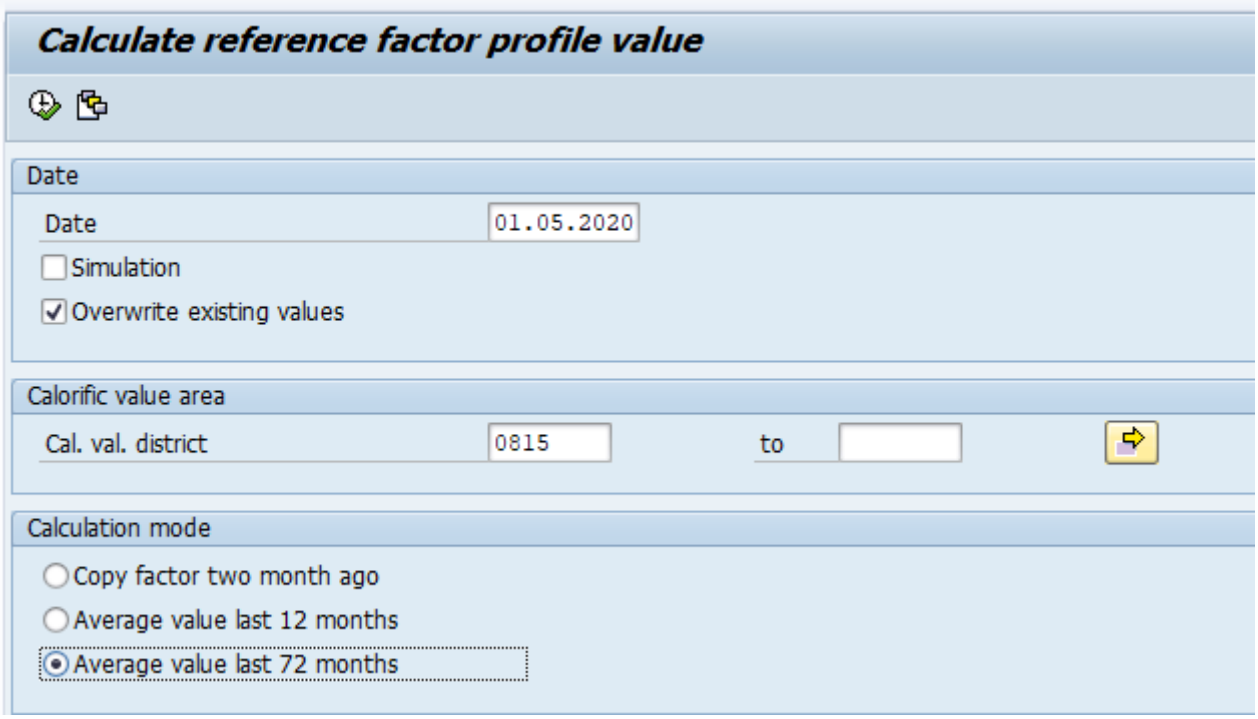


Figure 88: Example variant reference factor profile

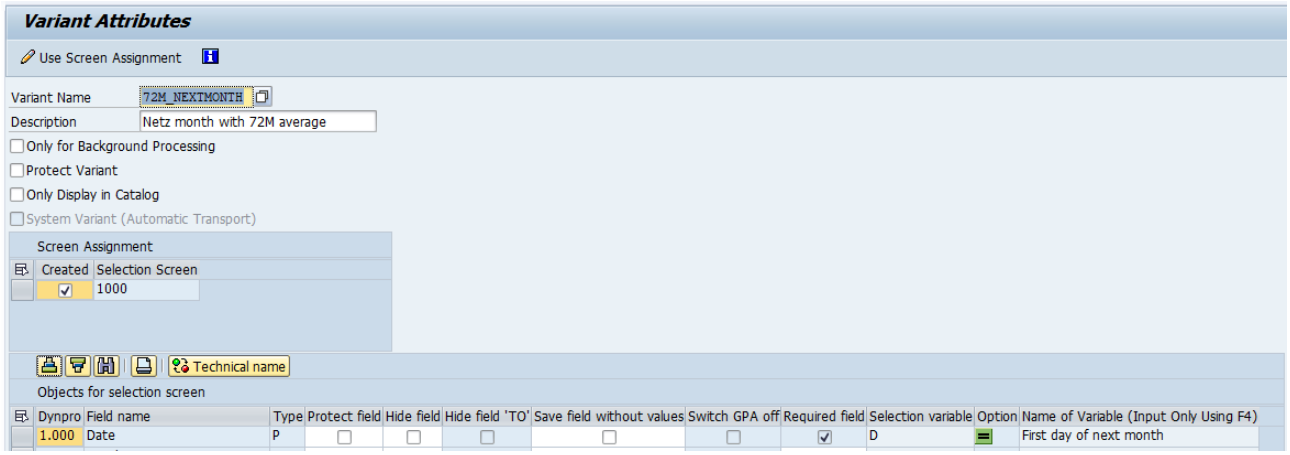


Figure 89: Example variant reference factor profile save screen

The formula for volume-based billing calorific value calculation can be integrated via automatic calculation workbench SPF integration Simple Process Flow connector.

2.5 Technical Quantity Determination for Performance Measured Customers

The technical quantity determination (TQD) is based on the functional modules

- */PLGA/EDM_CALC_TQD_RND* for the calculation of compressibility
- */PLGA/EDM_CALC_Z_SGERG_88_DVGW* for the TQD calculation according to G685
- */PLGA/EDM_CALC_Z_AGA8* for calculation of compression factor AGA8
- */PLGA/EDM_CALC_Z_SGERG_88* for SGERG 88 calculation of compressibility factor (ISO 12213-3)
- */PLGA/EDM_CALC_K_CORR_G486* Compressibility factor correction according to G486

For these functional modules a formula could be created.

2.5.1 Create Formula /PLGA/TQD accordance G685

For the function module */PLGA/EDM_CALC_TQD_RND* the following formula will be created using transaction *EEDM_FRM01* and confirmed pressing ENTER:

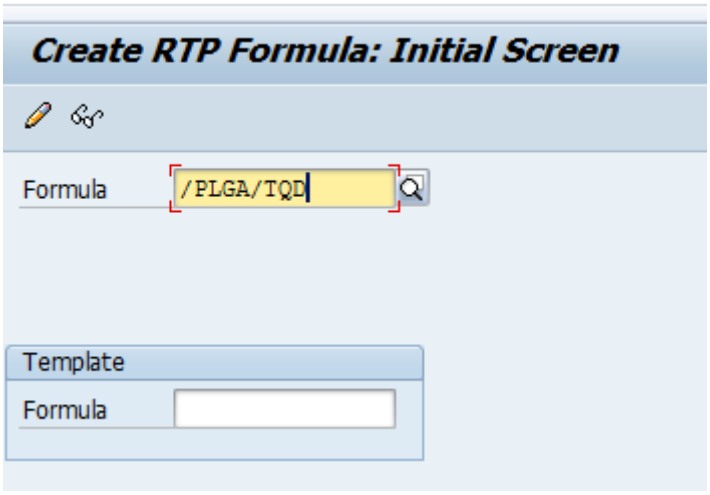


Figure 90: Create Formula for TQD

To do so the parameters

- Measured volume
- Temperature
- Standard temperature
- Zero temperature
- Measured pressure
- Air pressure
- Standard air pressure
- Compressibility
- Calorifc value

must be entered according to Figure . For each parameter the calculation mode will be handed over according to Figure .

The profile value categories, dimensions and units of measurement can be defined in customizing.

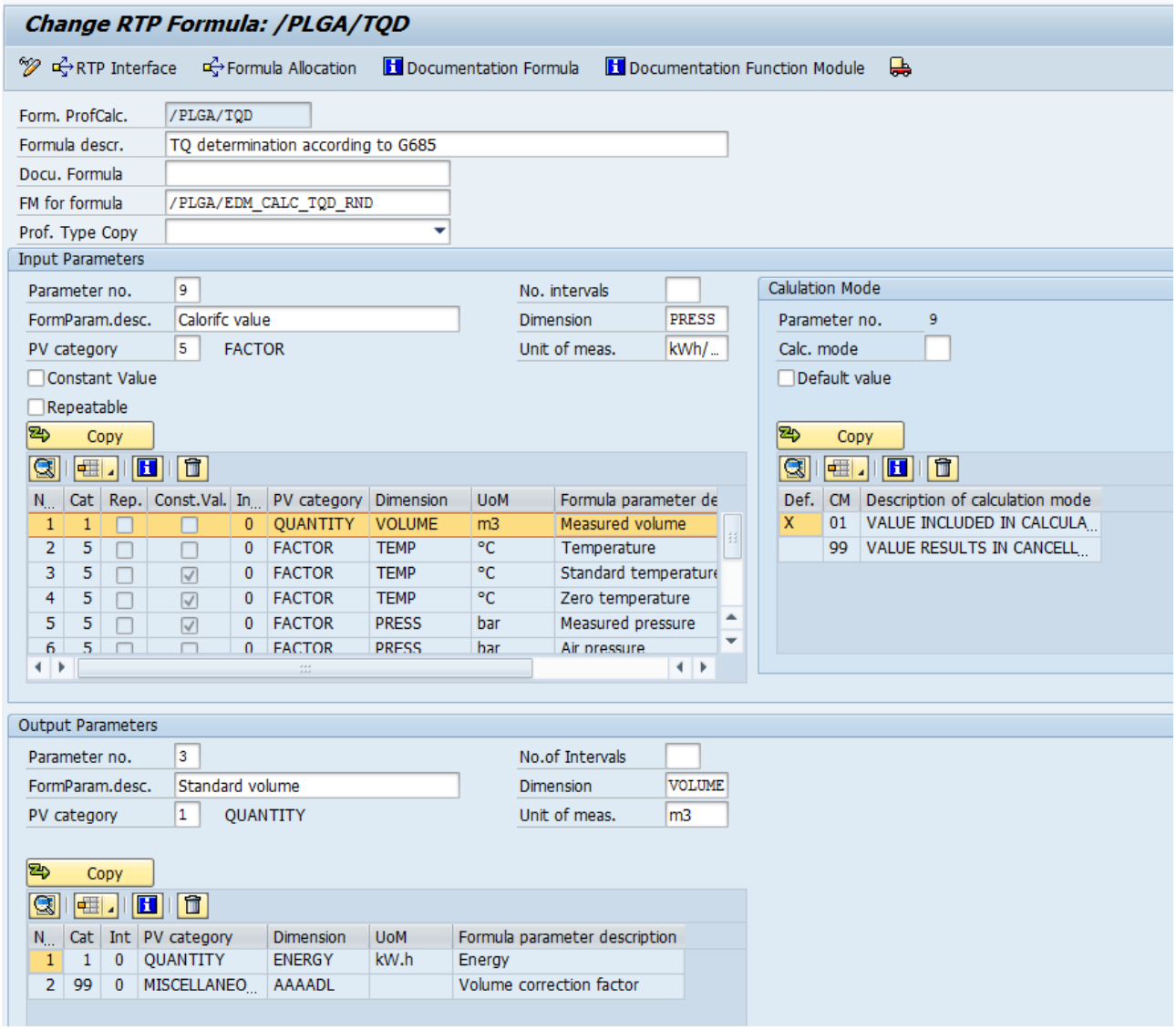


Figure 91: Formula Structure TQ determination according to G685

Furthermore, the output parameter

- Energy
- Volume correction factor

must be entered and transferred according to the previous figureFigure . After entering the data, the settings must be saved. The following message will appear:

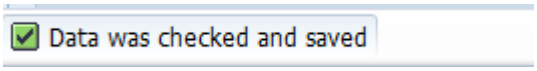


Figure 92: Saved Formula for TQ determination according to G685

The formula can be used on user side afterwards.

2.5.2 Create Formula /PLGA/COM_DVGW after DVGW

This function module implements the calculation of compression factors Implementing "ISO 12213 Part 3: Calculation using physical properties" also known as SGERG-88 equation. It's a fairly accurate conversion of the original FORTRAN code to ABAP. Implementation passes the validation test in ISO 12213-3 Annex C Parameters 1 to 6 are mapped to the input parameters of SGERG-88 with the sole exception of parameter 5 expecting kwh/m³ instead of MJ/m³. Parameter 7 serves as conversion base for processing

percentages based on either 1 or 100. Parameter 8 is needed for controlling the conversion of absolute density measured in kg/m³ to relative density as needed for the calculation. For suiting DVGW G486-B2 in addition to the original calculation a second one is performed for calculating the compression factor in normative conditions (Zn). The result is thus given as K = Z / Zn.

For the functional module */PLGA/EDM_CALC_Z_SGERG_88_DVGW,* the following formula will be created using transaction *EEDM_FRM01* and confirmed pressing ENTER:

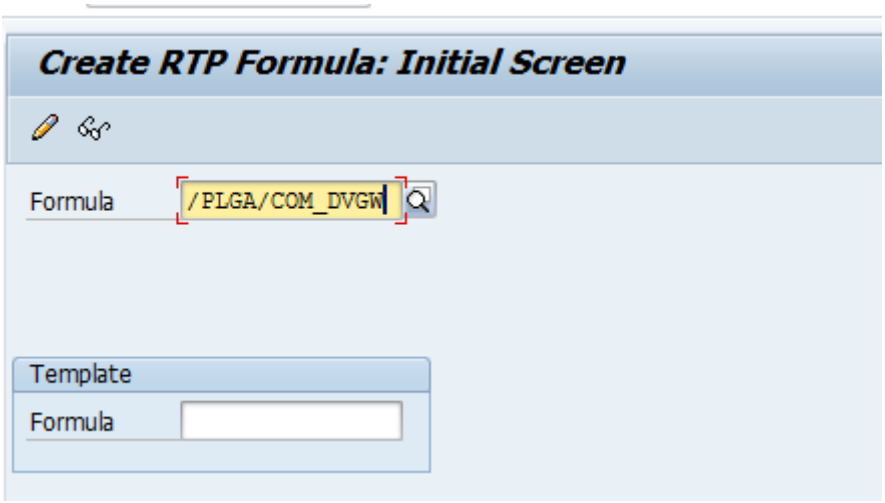


Figure 93: Create Formula

To do that, the nine input parameters

- Pressure,
- Temperature,
- Carbon dioxide,
- Hydrogen,
- Calorific value,
- Standard density,
- % basis,
- Conversion density

must be entered and transferred according to the following figure. For each parameter the calculation mode will be handed over according to the following figure.

Formula variable settings can be found in customizing. Profile value categories, dimensions and units of measurement must be defined in SAP® Customizing.

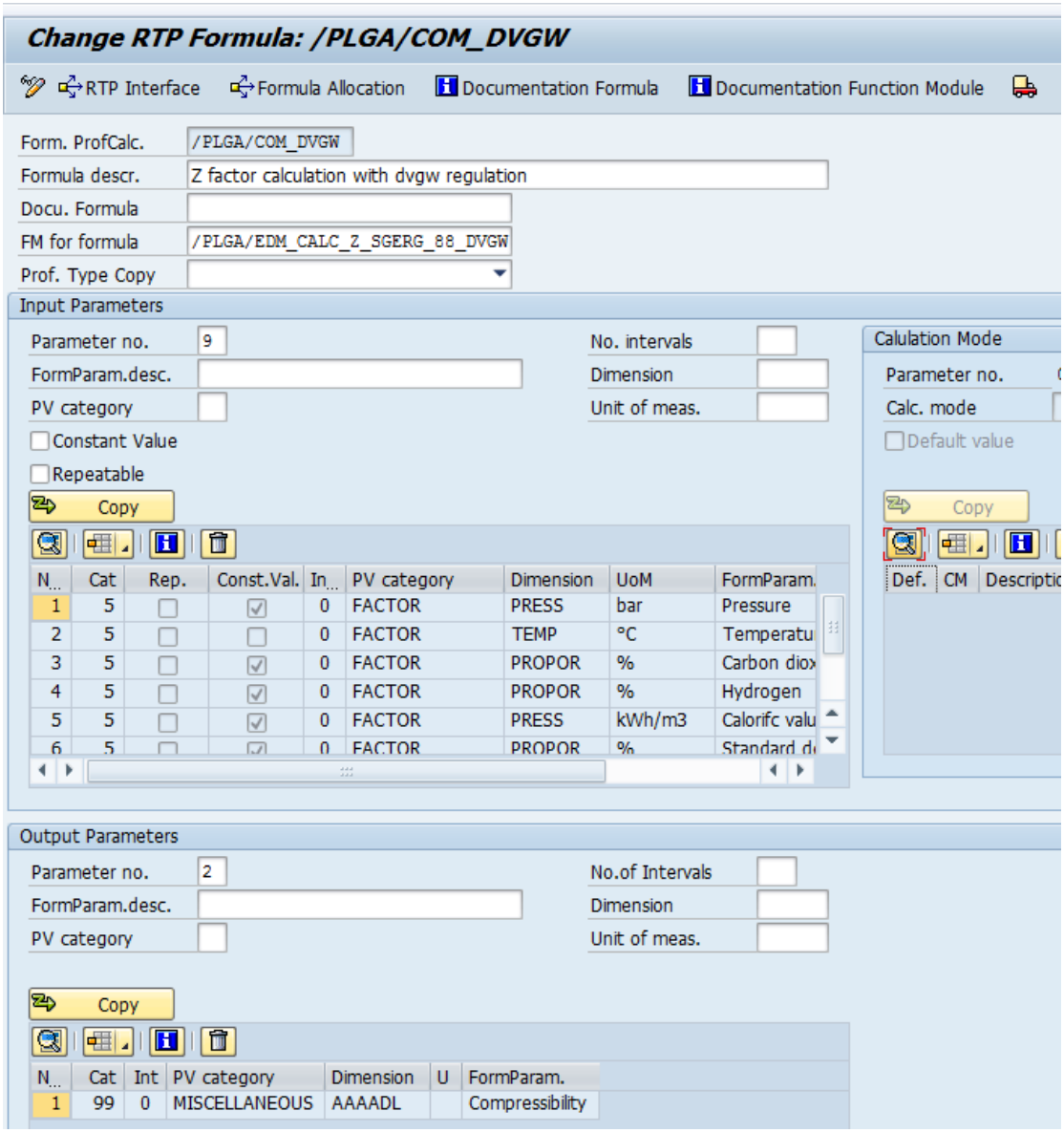


Figure 94: Formula Structure calculation of compressibility

Additionally, the output parameter

- Compressibility

must be entered and transferred according to the previous figure. After entering the data you must save. The following message will appear:

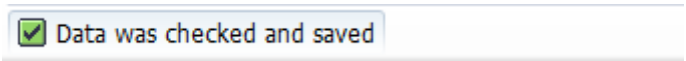


Figure 95: Saved Formula for calculation of compressibility

This formula can be used on user side afterwards.

2.5.3 Create Formula /PLGA/COMP_AGA8 accordance AGA8

For the function module */PLGA/EDM_CALC_Z_AGA8* the following formula will be created using transaction *EEDM_FRM01* and confirmed pressing ENTER.

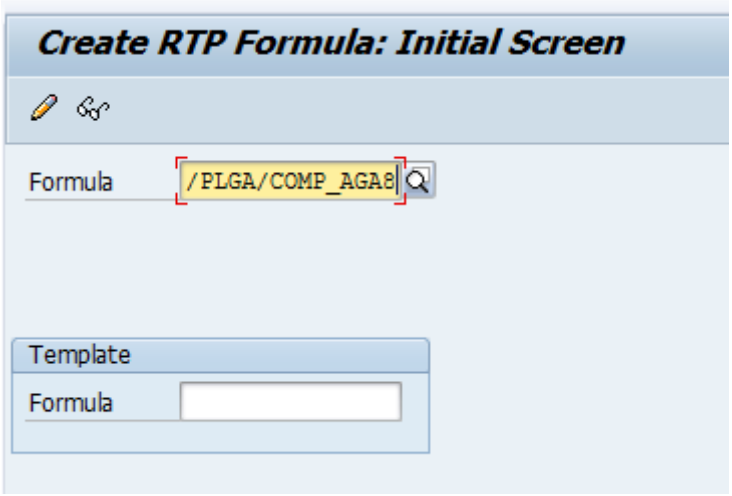


Figure 96: Create Formula for compressibility AGA8

To do so, the parameters

- Gas proportion CH4
- Gas proportion N2
- Gas proportion CO2
- Gas proportion C2H6
- Gas proportion C3H6
- Gas proportion C4H10
- Gas proportion N-C4H10
- Gas proportion I-C4H10
- Gas proportion I-C5H12
- Gas proportion N-C6
- Gas proportion N-C7
- Gas proportion N-C8
- Gas proportion N-C9
- Gas proportion N-C10
- Gas proportion HE
- Gas proportion AR
- Gas proportion H2
- Gas proportion O2
- Gas proportion CO
- Gas proportion H2O
- Gas proportion H2S
- Temperature
- Pressure

must be entered according to the following figureFigure . For each parameter, the calculation mode will be handed over according to the following figure.

The profile value categories, dimensions and units of measurement can be defined in customizing.

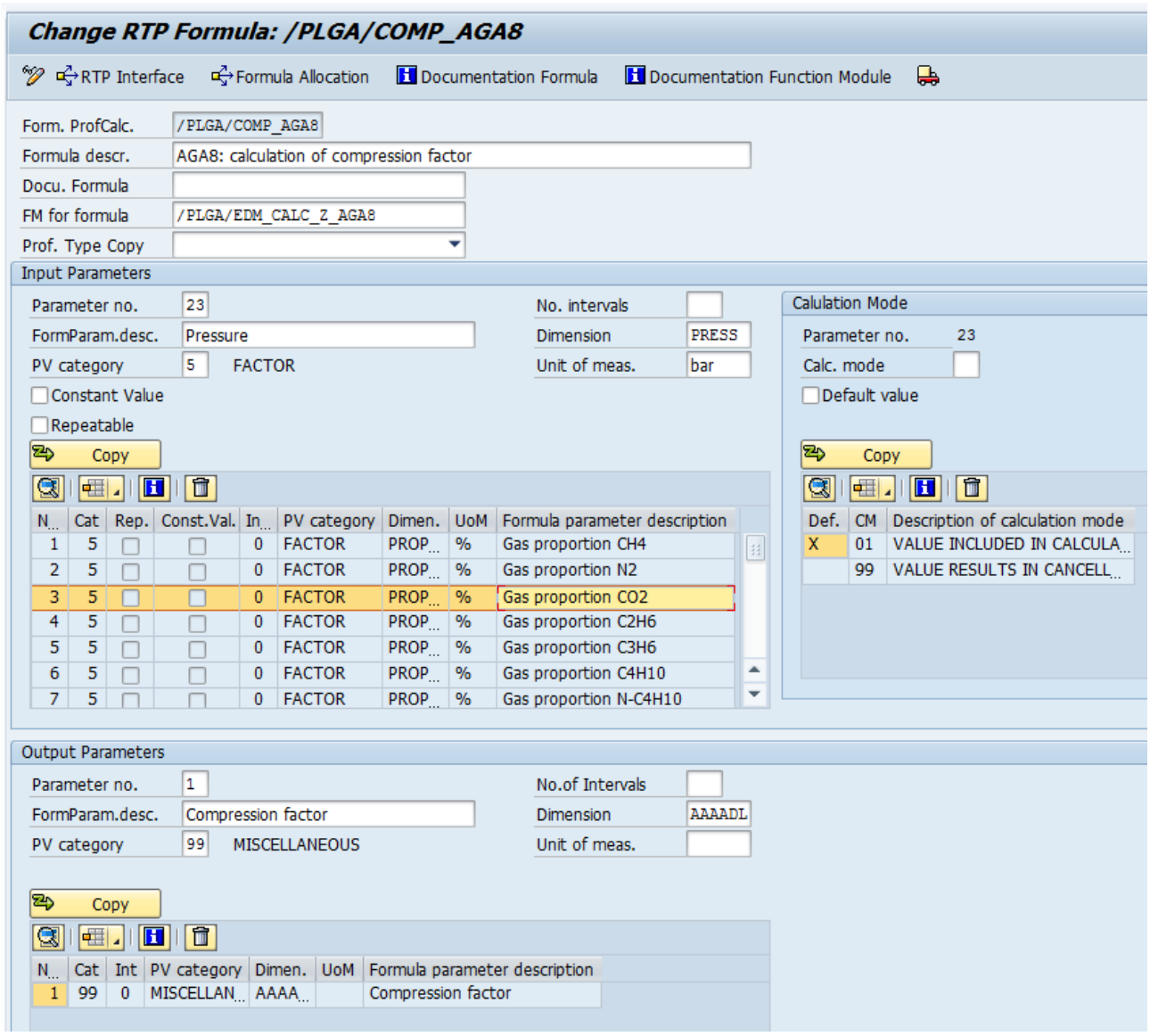


Figure 97: Formula Structure TQ determination according to G685

Furthermore, the output parameter

1. Compression factor

must be entered and transferred according to the previous figure. After entering the data, the settings must be saved. The following message will appear:

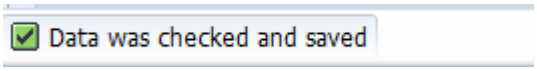


Figure 98: Saved Formula for TQ determination according to G685

Afterwards the formula can be used on user side.

2.5.4 Create Formula /PLGA/SGERG88

This function module implements the calculation of compression factors Implementing "ISO 12213 Part 3: Calculation using physical properties" also known as SGERG-88 equation. It's a fairly accurate conversion of the original FORTRAN code to ABAP. Implementation passes the validation test in ISO 12213-3 Annex C Parameters 1 to 6 are mapped to the input parameters of SGERG-88 with the sole exception of

parameter 5 expecting kwh/m³ instead of MJ/m³. Parameter 7 serves as conversion base for processing percentages based on either 1 or 100. Parameter 8 is needed for controlling the conversion of absolute density measured in kg/m³ to relative density as needed for the calculation.

For the functional module */PLGA/EDM_CALC_Z_SGERG_88* the following formula will be created using transaction *EEDM_FRM01* and confirmed pressing ENTER:

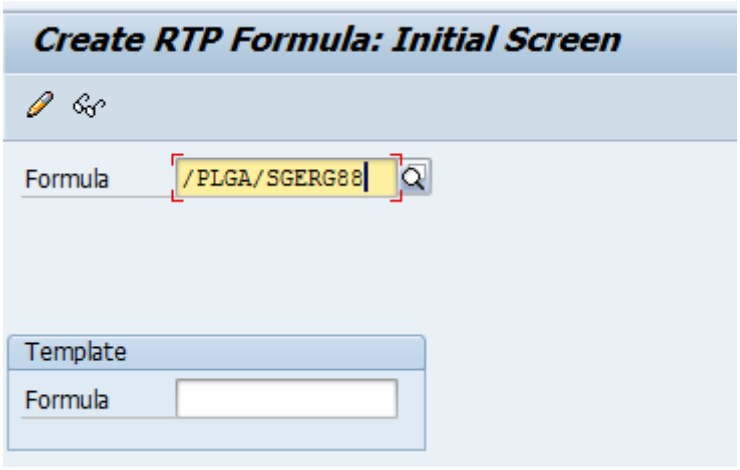


Figure 99: Create Formula

To do that, the nine input parameters

- Pressure,
- Temperature,
- Carbon dioxide,
- Hydrogen,
- Calorific value,
- Standard density,
- % basis,
- Conversion density

must be entered and transferred according to the following figure. For each parameter, the calculation mode will be handed over according to the following figure.

Formula variables settings can be found in the customizing. Profile value categories, dimensions and units of measurement have to be defined in the Customizing of SAP®.

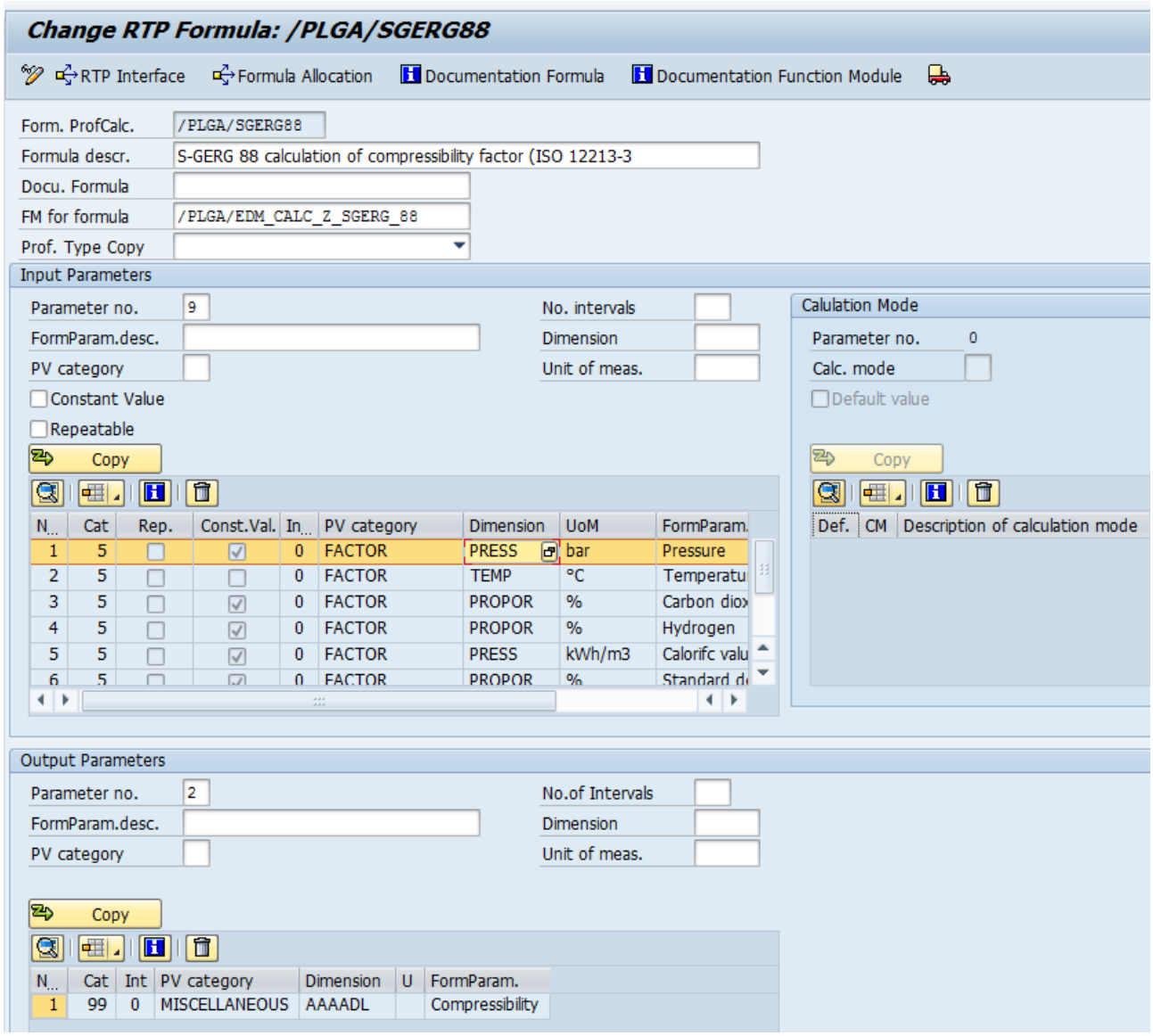


Figure 100: Formula Structure calculation of compressibility

Additionally, the output parameter

- Compressibility

must be entered and transferred according to the previous figure. After entering the data, you must save. The following message will appear:

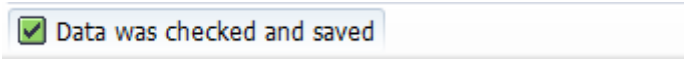


Figure 101: Saved Formula for calculation of compressibility

This formula can be used on user side afterwards.

2.5.5 Create Formula /PLGA/G486

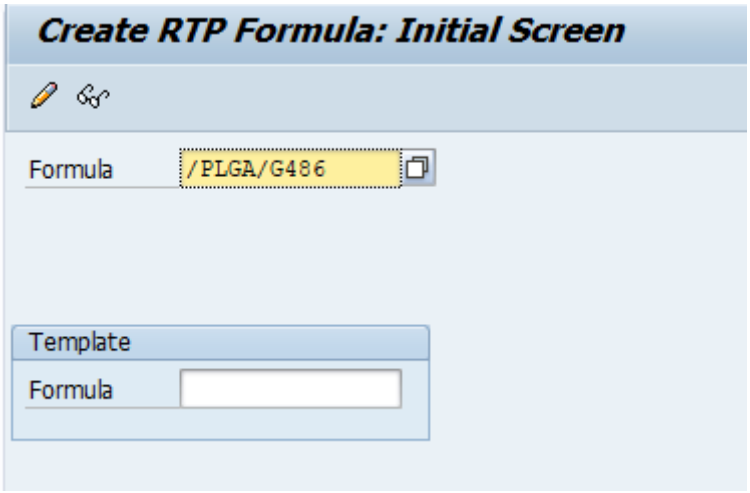


Figure 102: Create Formula

To do that, the four input parameters

- compressibility factor (calculated)
- compressibility factor (metering device)
- Volume
- Switch calculation Mode (this parameter is controlling which threshold needs to be used (0 = 0.001; 1 = 0.1 <= x <= 0.0025) and which type of calculation to be used. See DVGW G486 sections 6.6.2 and 6.6.3 for details). In mode "1" the mean values are calculated based on the calculation period (e.g. month). Otherwise it gets calculated on a daily base.

must be entered and transferred according to the following figure. For each parameter, the calculation mode will be handed over according to the following figure.

Formula variables settings can be found in the customizing. Profile value categories, dimensions and units of measurement have to be defined in the Customizing of SAP®.

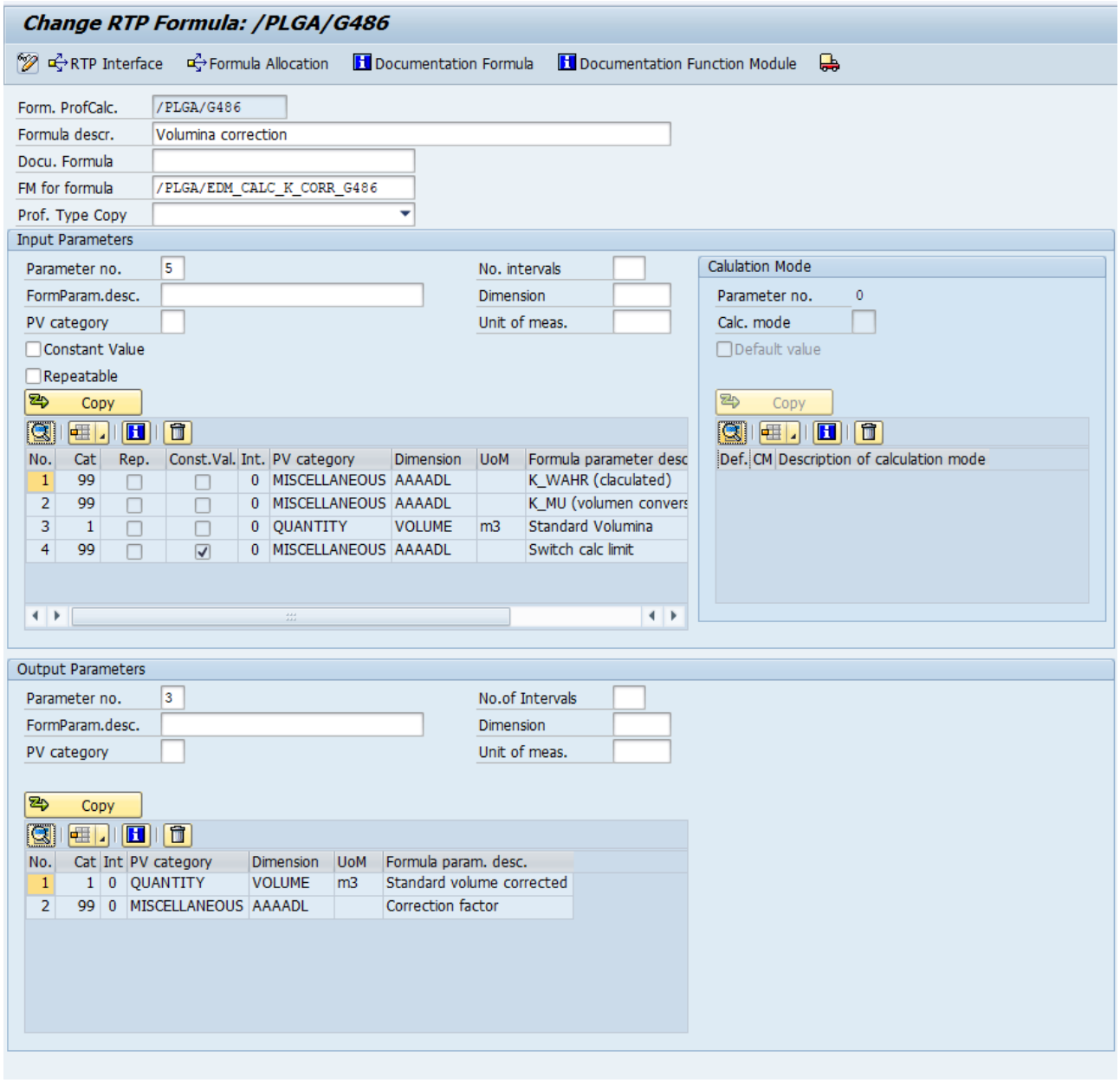


Figure 103: Formula Structure calculation of compressibility correction

Additionally, the output parameter

- Corrected volume
- Correction factor

must be entered and transferred according to the previous figure. After entering the data, you must save. The following message will appear:

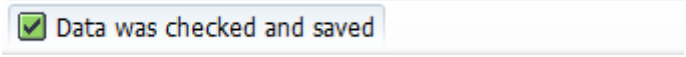


Figure 104: Saved Formula for calculation of compressibility correction

This formula can be used on user side afterwards.

2.6 Cluster Framework

To perform a check with the EDM Cluster Framework, it is necessary to define a configuration with transaction */PLGA/ECF_CST*.

2.6.1 Defining a Check

Transaction: */PLGA/ECF_CST*

In order to use a Check component in a cluster configuration, it needs to be defined first. Once a Check component is defined, it can be used in any number of cluster configurations.

To define a Check, double click the "Checks" folder on the left and choose "New Entries".

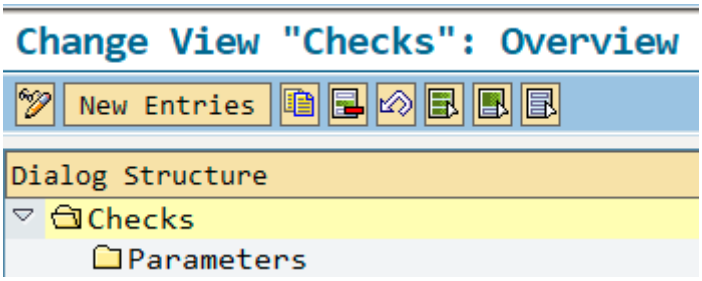


Figure 105: Start Define cluster-check

On the right side of the screen, provide the header data of the Finder.

- **Check ID**: The ID of the Check. The ID is used in cluster configurations to refer to this specific Check.
- **Check implementation**: The ABAP class implementing the business logic of this Check. Only non-abstract classes implementing the interface */PLGA/IF_ECF_CHECK* are suitable. The value help will list all such classes in the system.
- **Business Process Code**: The cluster framework can be used to create BPEM cases, if the relevant Business Process Code is provided here.
- **Check description**: A textual description of the check. This text is display in value helps of the cluster configuration.

When you are finished, save the Check.

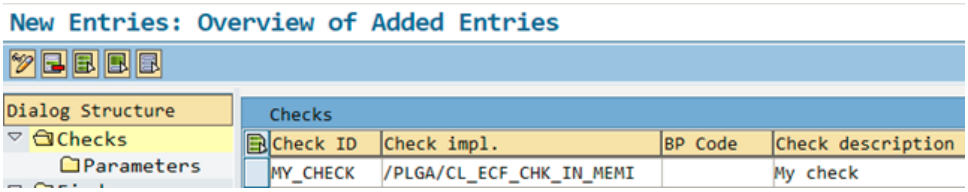


Figure 106: Cluster-Check – define check ID

2.6.1.1 Defining Check parameters

A Check can have an arbitrary number of parameters, for example a number of flags controlling various runtime behaviours. In order to supply the parameters with values in cluster configuration, they first have to be defined for the Check. This definition is used for consistency checks and provides a value help during the cluster configuration itself.

To define parameters of a specific Check, double click the "Checks" folder on the left side, then select the Check on the right side, double-click on the "Parameters" folder on the left side and then choose "New Entries" at the top.

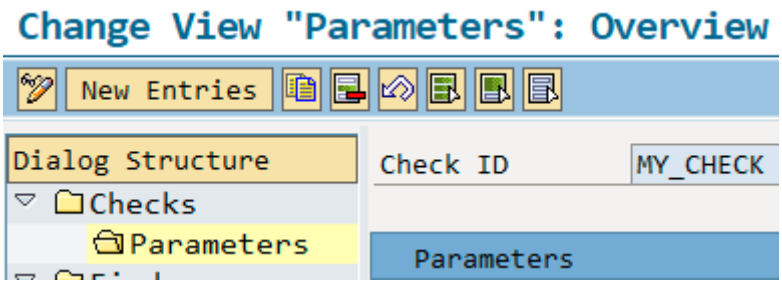


Figure 107: Cluster-Check – define check parameters

One the right side of the screen, provide the details of the Check parameters.

- **Name**: The name of the parameter
- **Type**: Data type of the parameter. This can be any data element in the system. A value help is provided.
- **Optional**: This flag indicates whether this parameter is optional or not. All parameters, which are not optional, must be provided with a value in a cluster configuration.
- **Description**: Free text. Usually it is a good idea to put some information here, on what the parameter does.

When you are finished, save the Check parameters.

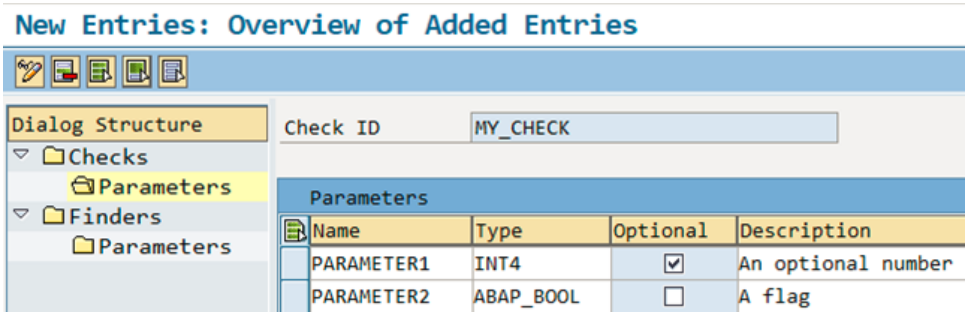


Figure 108: Cluster-Check – define check parameters details

2.6.2 Defining a Finder

Transaction: */PLGA/ECF_CST*

In order to use a Finder component in a cluster configuration, it needs to be defined first. Once a Finder component is defined, it can be used in any number of cluster configurations.

To define a Finder, double-click the "Finder" folder on the left and choose "New Entries".

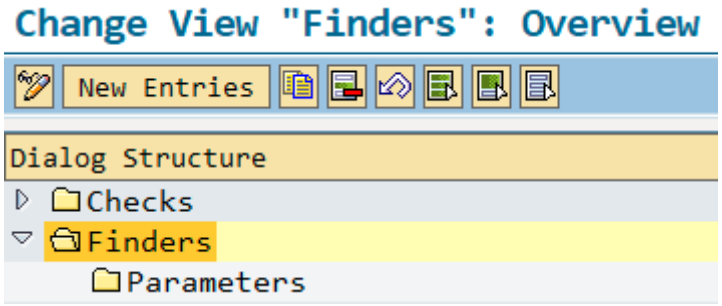


Figure 109: Cluster-Check – define finder

On the right side of the screen, provide the header data of the Finder.

- **Finder ID**: The ID of the Finder. The ID is used in cluster configurations to refer to this specific Finder.
- **Finder implementation**: The ABAP class implementing the business logic of this Finder. Only non-abstract classes implementing the interface */PLGA/IF_ECF_FINDER* are suitable. The value help will list all such classes in the system.
- **Check description**: A textual description of the Finder. This text is display in value helps of the cluster configuration.

When you are finished, save the Finder.

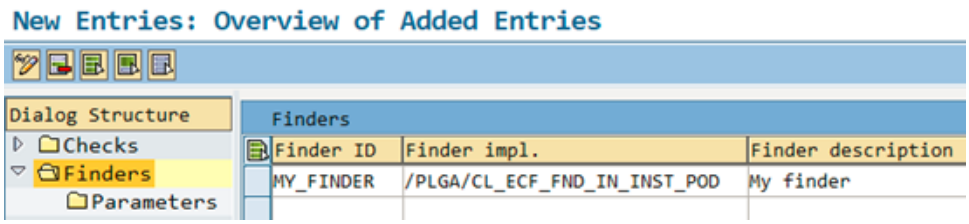


Figure 110: Cluster-Check – define finder details

2.6.2.1 Defining Finder parameters

A Finder can have an arbitrary number of parameters, for example a number of flags controlling various runtime behaviours. In order to supply the parameters with values in cluster configuration, they first have to be defined for the Finder. This definition is used for consistency checks and provides a value help during the cluster configuration itself.

To define parameters of a specific Finder, double click the "Finders" folder on the left side, then select the Finder on the right side, double-click on the "Parameters" folder on the left side and then choose "New Entries" at the top.

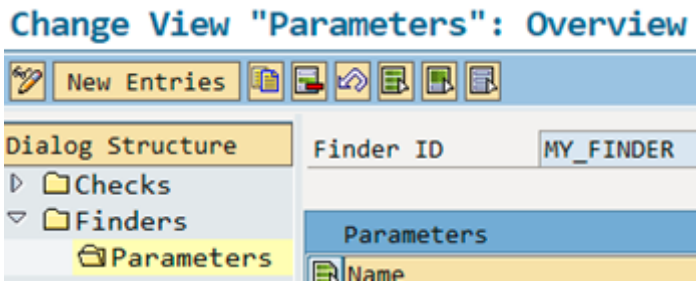


Figure 111: Cluster-Check – define finder

One the right side of the screen, provide the details of the Check parameters.

- **Name**: The name of the parameter
- **Type**: Data type of the parameter. This can be any data element in the system. A value help is provided.
- **Optional**: This flag indicates whether this parameter is optional or not. All parameters, which are not optional, must be provided with a value in a cluster configuration.
- **Description**: Free text. Usually it is a good idea to put some information here, on what the parameter does.

When you are finished, save the Check parameters.

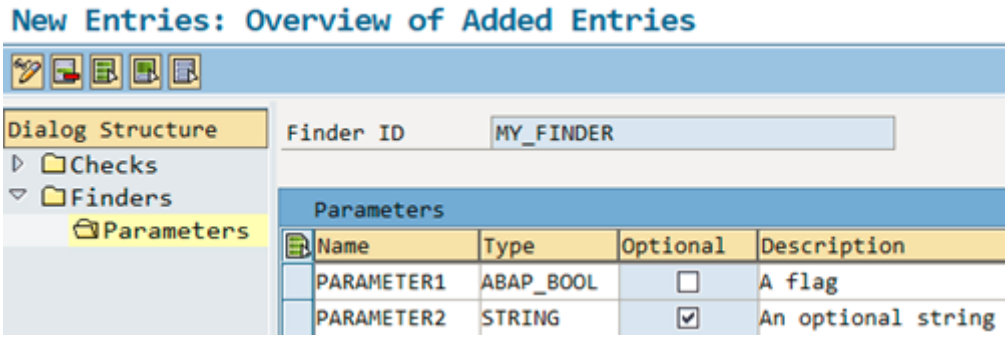


Figure 112: Cluster-Check – define finder details

2.6.3 Delivered content

The EDM Cluster Framework is delivered with a set of Finders and Checks. This content can be categorized as follows:

- Finder/Checks for installations
- Finder/Checks for EDM profiles
- Finder/Checks for load profiles (SLP)

2.6.3.1 Content for Installation - Finder

For checking installations, three Finder and four Checks are delivered. They all work on the same kind of data and can thus be combined in any way.

Finder: IN_INST_POD

- Implementation class: **/PLGA/CL_ECF_FND_IN_INST_POD**
- Description: Finder for installation based on installation number and pod description

Parameter name	Parameter type	is Optional	Description
ANLAGE_AB	ANLAGE	No	Installation from
ANLAGE_BIS	ANLAGE	No	nstallation to
EXT_UI_AB	EXT_UI	Yes	external PoD description from
EXT_UI_BIS	EXT_UI	Yes	external PoD description to

Table 14: Parameter IN_INST_POD

Finder: IN_INST_POD2

- Implementation class: **/PLGA/CL_ECF_FND_IN_INST_POD2**
- Description: Finder for installation based on rate type and division

Parameter name	Parameter type	is Optional	Description
CHECK_FOR_ACTIVE_CONTRACT	ABAP_BOOL	Yes	"X" for check for active contract
CHECK_FOR_SETTLMENT_RELEVANCE	ABAP_BOOL	Yes	"X" for check for active settlement relevance
RATE_CATEGORY	TARIFTYP_ANL	Yes	rate type
DIVISION	SPARTE	Yes	Division

Table 15: Parameter IN_INST_POD2

Finder: IN_POD_ADV

- Implementation class: **/PLGA/CL_ECF_FND_IN_POD_ADV**
- Description: Finder for installation based on rate type, division and active contract

Parameter name	Parameter type	is Optional	Description
CONTRACT_SELECTION	/PLGA/DECF_POD_EXT_CONTR_SEL	Yes	X for active contract, B contract start within time range, E contract ends within time range
CHECK_FOR_SETTLMENT_RELEVANCE	ABAP_BOOL	Yes	"X" for check of settlement relevance
RATE_CATEGORY_ADV_SEL_TYPE	/PLGA/DECF_AS_PAR_TYPE	Yes	rate type
DIVISION	SPARTE	Yes	Division

Table 16: Parameter IN_POD_ADV

2.6.3.2 Content for Installation – Checker

Checker: IN_MEMI

- Implementation class: **/PLGA/CL_ECF_CHK_IN_MEMI**
- Description: Checker for settlement view of installation

Parameter name	Parameter type	is Optional	Description
SETTLVIEW	E_EDMSETTLVIEW	No	Settlement view
CHECK_VIEWADD_ASS_COMPLETE	ABAP_BOOL	Yes	X for run check

Table 17: Parameter IN_MEMI

Checker: IN_SETTL

- Implementation class: **/PLGA/CL_ECF_CHK_IN_SETTL**
- Description: Checker for settlement assignment of installation

Parameter name	Parameter type	is Optional	Description
SERVICE_TYPE	SERCODE	No	Service Type from Service Provider
CHECK_DEADLINE	ABAP_BOOL	Yes	"X" for run check
DEADLINE_NUM_DAYS	I	Yes	Number of Days for calculation for deadline
REVERSE_LOGIC	ABAP_BOOL	Yes	"X" for reverse check

Table 18: Parameter IN_SETTL

Checker: IN_TEMPAREA

- Implementation class: **/PLGA/CL_ECF_CHK_IN_TEMPAREA**
- Description: Checker for temparea assignment of installation

Parameter name	Parameter type	is Optional	Description

PROFROLE	PROFROLE	No	Profrole

Table 19: Parameter IN_TEMPAREA

Checker: IN_VOLTLVL

- Implementation class: **/PLGA/CL_ECF_CHK_IN_VOLTLVL**
- Description: Checker for grid assignment of installation

Parameter name	Parameter type	is Optional	Description
Not relevant	Not relevant	Not relevant	Not relevant

Table 20: Parameter IN_VOLTLVL

2.6.3.3 Content for standard EDM Profile - Finder

Finder: PRD_PROFILE

- Implementation class: **/PLGA/CL_ECF_FND_PRD_PROFILE**
- Description: Finder for profiles based on division and profile role

Parameter name	Parameter type	is Optional	Description
SPARTE	SPARTE	Yes	Division
PROFROLE	PROFROLE	No	Profrole
MATNR	MATNR	Yes	Equipment number

Table 21: Parameter PRP_PROFILE

Finder: PR_BY_NR

- Implementation class: **/PLGA/CL_ECF_FND_PR_BY_NR**
- Description: Finder for profiles based on division and profile number from / to

Parameter name	Parameter type	is Optional	Description
SPARTE	SPARTE	Yes	Division
PROFILE_FROM	E_PROFILE	Yes	profile number from
PROFILE_TO	E_PROFILE	Yes	profile number to
PROFTYPE	PROFTYPE	Yes	Proftype

Table 22: Parameter PR_BY_NR

Finder: PRD_ADV

- Implementation class: **/PLGA/CL_ECF_FND_PRD_ADV**
- Description: Finder for profiles based on advanced selection criteria

Parameter name	Parameter type	is Optional	Description
USE_ADV_SEL_PAR	ABAP_BOOL	Yes	"X" for use advance selection
PROFTYPE	PROFTYPE	Yes	Proftype

Table 23: Parameter PRD_ADV

Finder: PRP_ADV

- Implementation class: **/PLGA/CL_ECF_FND_PRP_ADV**
- Description: Finder for profiles based on advanced selection criteria

Parameter name	Parameter type	is Optional	Description
SPARTE	SPARTE	Yes	Division
PROFROLE	PROFROLE	Yes	Profrole
MATNR	MATNR	Yes	Equipment number
USE_ADV_SEL_PAR	ABAP_BOOL	Yes	"X" for use advance selection

Table 24: Parameter PRP_ADV

2.6.3.4 Content for standard EDM Profile - Checker

Checker: PRD_STAT_O

- Implementation class: **/PLGA/CL_ECF_CHK_PRD_STAT_0**
- Description: Checker of profile based on profile value status or zero values

Parameter name	Parameter type	is Optional	Description
CHECK_ZERO_VALUES	ABAP_BOOL	Yes	"X" for check of zero values
STATUS_1	J_STATUS	Yes	Optional status
STATUS_2	J_STATUS	Yes	Optional status
STATUS_3	J_STATUS	Yes	Optional status

Table 25: Parameter PRD_STAT_0

Checker: PRP_STAT_O

- Implementation class: **/PLGA/CL_ECF_CHK_PRP_STAT_0**
- Description: Checker of profile based on profile value status or zero values

Parameter name	Parameter type	is Optional	Description
CHECK_ZERO_VALUES	ABAP_BOOL	Yes	"X" for check of zero values
STATUS_1	J_STATUS	Yes	Optional status
STATUS_2	J_STATUS	Yes	Optional status
STATUS_3	J_STATUS	Yes	Optional status

Table 26: Parameter PRP_STAT_0

Checker: PRP_STAT_O_C

- Implementation class: **/PLGA/CL_ECF_CHK_PRP_STAT_0_C**
- Description: Checker of profile values (status/zero values) with counting

Parameter name	Parameter type	is Optional	Description
CHECK_ADD_STAT_ONLY	ABAP_BOOL	Yes	"X" for check only optional status
INTERVAL_TYPE	/PLGA/DECF_INTERVAL_TYPE	No	Interval type
INTERVAL_LENGTH	I	No	Interval length

STATUS_1	J_STATUS	Yes	Optional status
STATUS_2	J_STATUS	Yes	Optional status

Table 27: Parameter PRP_STAT_0_C

Checker: PRP_LIM

- Implementation class: **/PLGA/CL_ECF_CHK_PRP_LIM**
- Description: Checker of profile values against limits

Parameter name	Parameter type	is Optional	Description
CHECK_LOWER_LIMIT	ABAP_BOOL	Yes	"X" for run check lower limit
CHECK_UPPER_LIMIT	ABAP_BOOL	Yes	"X" for run check upper limit
REMOVE_FROM_TASKDATA	ABAP_BOOL	Yes	"X" for remove from task data

Table 28: Parameter PRP_LIM

Checker: PRD_LIM

- Implementation class: **/PLGA/CL_ECF_CHK_PRD_LIM**
- Description: Checker of profile values against limits

Parameter name	Parameter type	is Optional	Description
CHECK_LOWER_LIMIT	ABAP_BOOL	Yes	"X" for run check lower limit
CHECK_UPPER_LIMIT	ABAP_BOOL	Yes	"X" for run check upper limit
REMOVE_FROM_TASKDATA	ABAP_BOOL	Yes	"X" for remove from task data

Table 29: Parameter PRD_LIM

Checker: PRD_LIMSUM

- Implementation class: **/PLGA/CL_ECF_CHK_PRD_LIM_SUM**
- Description: Checker of profile values against limits with summation

Parameter name	Parameter type	is Optional	Description
CHECK_LOWER_LIMIT	ABAP_BOOL	Yes	"X" for run check lower limit
CHECK_UPPER_LIMIT	ABAP_BOOL	Yes	"X" for run check upper limit
REMOVE_FROM_TASKDATA	ABAP_BOOL	Yes	"X" for remove from task data
INTERVAL_TYPE	/PLGA/DECF_INTERVAL_TYPE	No	Interval type
INTERVAL_LENGTH	I	No	Interval length

Table 30: Parameter PRD_LIMSUM

Checker: PRP_LIMSUM

- Implementation class: **/PLGA/CL_ECF_CHK_PRP_LIM_SUM**
- Description: Checker of profile values against limits with summation

Parameter name	Parameter type	is Optional	Description
CHECK_LOWER_LIMIT	ABAP_BOOL	Yes	"X" for run check lower limit
CHECK_UPPER_LIMIT	ABAP_BOOL	Yes	"X" for run check upper limit
REMOVE_FROM_TASKDATA	ABAP_BOOL	Yes	"X" for remove from task data
INTERVAL_TYPE	/PLGA/DECF_INTERVAL_TYPE	No	Interval type
INTERVAL_LENGTH	I	No	Interval length

Table 31: Parameter PRP_LIMSUM

Checker: PRP_METER

- Implementation class: **/PLGA/CL_ECF_CHK_PRP_METER**
- Description: Checker of profile values against meter reading

Parameter name	Parameter type	is Optional	Description
MISSING_MR_IS_ERROR	ABAP_BOOL	Yes	"X" for red flag for missing value
REMOVE_FROM_TASKDATA	ABAP_BOOL	Yes	"X" for remove from task data
UNIT_OF_MEASUREMENT	E_MASSREAD	Yes	Unit of measurement

Table 32: Parameter PRP_METER

Checker: PR_UNROLLED

- Implementation class: **/PLGA/CL_ECF_CHK_PR_UNROLLED**
- Description: Checker of profile values against initial status

Parameter name	Parameter type	is Optional	Description
LOG_INDIVIDUAL_TS	ABAP_BOOL	Yes	"X" for run check

Table 33: Parameter PR_UNROLLED

Checker: PRP_CMP

- Implementation class: **/PLGA/CL_ECF_CHK_PRP_CMP**
- Description: Checker of profile values against other profile values

Parameter name	Parameter type	is Optional	Description
REF_PROFILE_MUST_EXIST	ABAP_BOOL	Yes	"X" for check of reference profile
NUM_DAYS_FOR_REF_CALC	I	Yes	Number of days
USE_AVERAGE	ABAP_BOOL	Yes	"X" for use average
CHECK_ABSOLUTE_DEVIATION	ABAP_BOOL	Yes	"X" for use absolute deviation

Table 34: Parameter PRP_CMP

Checker: PRP_CMP_SUM

- Implementation class: **/PLGA/CL_ECF_CHK_PRP_CMP_SUM**
- Description: Checker of profile values against other profile values summation

Parameter name	Parameter type	is Optional	Description
REF_PROFILE_MUST_EXIST	ABAP_BOOL	Yes	"X" for check of reference profile
NUM_DAYS_FOR_REF_CALC	I	Yes	Number of days
USE_AVERAGE	ABAP_BOOL	Yes	"X" for use average
CHECK_ABSOLUTE_DEVIATION	ABAP_BOOL	Yes	"X" for use absolute deviation

Table 35: Parameter PRP_CMP_SUM

Checker: PRP_CMP_SU_A

- Implementation class: **/PLGA/CL_ECF_CHK_PRP_CMP_SU_A**
- Description: Checker of profile values against other prof. val. sum. and time references

Parameter name	Parameter type	is Optional	Description
COMPARE_AGAINST_REF_PROF	ABAP_BOOL	No	"X" for check of reference profile
REF_PERIOD_TYPE	/PLGA/DECF_INTERVAL_TYPE	Yes	Reference period
REF_PERIOD_LENGTH	I	Yes	Reference period length
SUM_TYPE	/PLGA/DECF_INTERVAL_TYPE	No	Sum type
USE_AVERAGE	ABAP_BOOL	Yes	"X" for use average

Table 36: Parameter PRP_CMP_SU_A

Checker: PRP_CMP_SU_A2

- Implementation class: **/PLGA/CL_ECF_CHK_PRP_CMP_SU_A2**
- Description: Checker of profile values against other prof. val. sum. and time references

Parameter name	Parameter type	is Optional	Description
COMPARE_AGAINST_REF_PROF	ABAP_BOOL	Yes	"X" for check against reference profile
REF_PERIOD_TYPE	/PLGA/DECF_INTERVAL_TYPE	Yes	Reference period (D=Day, m= Month, Y=Year)
REF_PERIOD_LENGTH	I	Yes	Reference period length
SUM_TYPE	/PLGA/DECF_INTERVAL_TYPE	Yes	Sum type (D=Day, M=Month, Y=Year)
CORRECT_REF_DATE	ABAP_BOOL	Yes	"X" for check against reference date
FACTORY_CALENDAR	WFCID	Yes	Factory calendar

Table 37: Parameter PRP_CMP_SU_A2

2.6.3.5 Content for SLP-Profile - Finder

Finder: SLP_CG_AR

- Implementation class: **/PLGA/CL_ECF_FND_SLP_CG_AR**
- Description: Finder of SLP profile based on profrole, contract and customer group

Parameter name	Parameter type	is Optional	Description
PROFROLE	PROFROLE	Yes	Profrole
CHECK_CONTRACT	ABAP_BOOL	Yes	"X" for check active contract
GRID_OPERATOR	/PLGA/DECF_GRID_OPERATOR	Yes	Network operator
CUSTOMER_GROUP	/PLGA/DECF_CUST_GRP	No	Customer group

Table 38: Parameter SLP_CG_AR

Finder: SLP_RC

- Implementation class: **/PLGA/CL_ECF_FND_SLP_RC**
- Description: Finder of SLP profile based on profrole actual or forecaset, contract and rate category

Parameter name	Parameter type	is Optional	Descritption
PROFROLE_IST	PROFROLE	No	Profrole actual
CHECK_CONTRACT	ABAP_BOOL	No	"X" for check active contract
PROFROLE_PROG	PROFROLE	Yes	Profrole forecast
RATE_CATEGORY	TARIFTYP_ANL	Yes	Rate category

Table 39: Parameter SLP_RC

Finder: SLP_CG_MS

- Implementation class: **/PLGA/CL_ECF_FND_SLP_CG_MS**
- Description: Finder of SLP profile based on network operator, contract and division

Parameter name	Parameter type	is Optional	Description
GRID_OPERATOR	/PLGA/DECF_GRID_OPERATOR	Yes	Network operator
CHECK_CONTRACT	ABAP_BOOL	No	"X" for check active contract
DIVISION	SPARTE	No	Division

Table 40: Parameter SLP_CG_MS

Finder: SLP_CG

- Implementation class: **/PLGA/CL_ECF_FND_SLP_CG**
- Description: Finder of SLP profile based on network operator, contract and customer group

Parameter name	Parameter type	is Optional	Description
GRID_ID	GRID_ID	Yes	Grid

CHECK_CONTRACT	ABAP_BOOL	Yes	"X" for check active contract
GRID_OPERATOR	/PLGA/DECF_GRID_OPERATOR	Yes	Network Operator
CUSTOMER_GROUP	/PLGA/DECF_CUST_GRP	Yes	Customer group

Table 41: Parameter SLP_CG

Finder: SLP_CGG_MS

- Implementation class: **/PLGA/CL_ECF_FND_SLP_CGG_MS**
- Description: Finder of SLP profile based on network operator, contract, division and grid

Parameter name	Parameter type	is Optional	
GRID_OPERATOR	/PLGA/DECF_GRID_OPERATOR	Yes	Network operator
CHECK_CONTRACT	ABAP_BOOL	No	"X" for check active contract
DIVISION	SPARTE	No	Division
GRID_ID	GRID_ID	Yes	Grid ID

Table 42: Parameter SLP_CGG_MS

2.6.3.6 Content for SLP-Profile – Checker

Checker: SLP_UF_ISTPR

- Implementation class: **/PLGA/CL_ECF_CHK_SLP_UF_ISTPR**
- Description: Checker of customer value form actual and forecast SLP profiles

Parameter name	Parameter type	is Optional	Description
NUM_DAYS_FOR_REF_TS	I	No	Number of days
REMOVE_FROM_TASKDATA	ABAP_BOOL	Yes	"X" for remove from task data
USE_EXTENDED_TRIGGER	ABAP_BOOL	Yes	"X" for use advanced trigger

Table 43: Parameter SLP_UF_ISTPR

Checker: SLP_UF_LIMIT

- Implementation class: **/PLGA/CL_ECF_CHK_SLP_UF_LIMIT**
- Description: Checker of limits of customer value from SLP profile

Parameter name	Parameter type	is Optional	Description
REMOVE_FROM_TASKDATA	ABAP_BOOL	Yes	"X" for remove from task data
CHECK_UPPER_LIMIT	ABAP_BOOL	Yes	"X" for check upper limit
CHECK_LOWER_LIMIT	ABAP_BOOL	Yes	"X" for check lower limit
PROFROLE	PROFROLE	Yes	Profrole

Table 44: Parameter SLP_UF_LIMIT

Checker: SLP_UF_PRV_P

- Implementation class: **/PLGA/CL_ECF_CHK_SLP_UF_PRV_P**
- Description: Checker of limits of customer value from SLP profile with advanced trigger

Parameter name	Parameter type	is Optional	Description
NUM_DAYS_FOR_REF_TS	I	No	Number of reference days
REMOVE_FROM_TASKDATA	ABAP_BOOL	Yes	"X" for remove from task data
PROFROLE	PROFROLE	Yes	Profrole
USE_EXTENDED_TRIGGER	ABAP_BOOL	Yes	"X" for use advanced trigger

Table 45: Parameter SLP_UF_PRV_P

2.6.4 Simple Process Flow connector

A cluster of the Cluster Framework can be simply integrated into the Simple Process Flow as a stand-alone node.

Each node represents a selection variant for the report /PLGA/ECF_EXECUTE_CLUSTER:

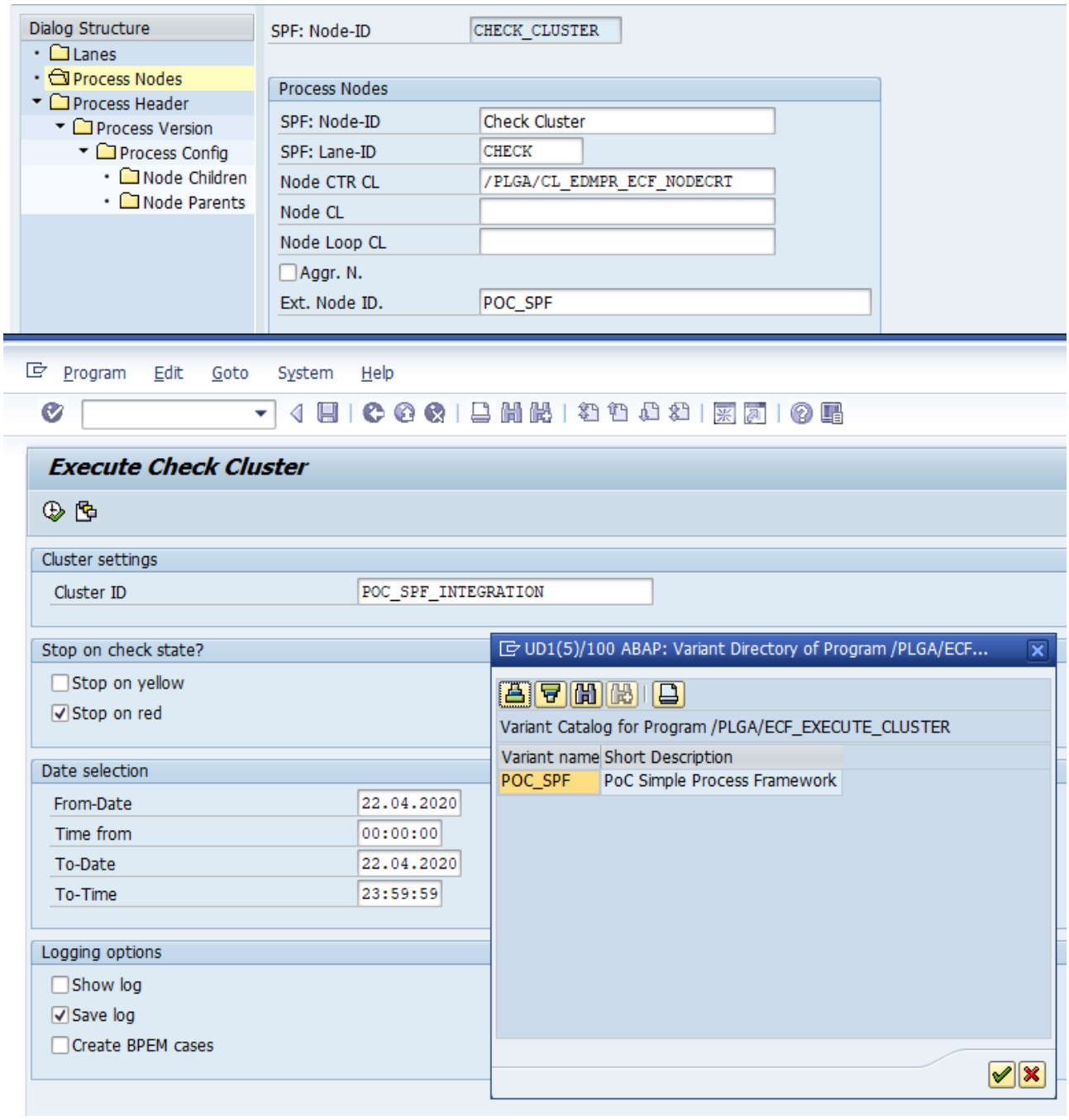


Figure 113: Example SPF connector Cluster Framework

Notes on Parameters (Maintain according SPF node maintenance):

Lane-ID: The selected lane identification should be compatible to stand-alone nodes (e.g. lanes with Lane-Class */PLGA/CL_EDMPR_NOOP_LANECTR*)

The node control class is */PLGA/CL_EDMPR_ECF_NODECRT*

The External Node identification shall contain the technical name of a variant of the report /PLGA/ECF_EXECUTE_CLUSTER.

2.7 Generic CSV/XML Importer

The Generic CSV/XML Importer is used to write XML or CSV Data into a Profile. The corresponding configuration can be done within the SAP GUI in the transaction *SPRO (SAP Reference IMG)*. The following path leads to the view cluster for the dedicated configuration:

SAP Customizing Implementation Guide ->Extension for Utilities by PROLOGA ->Energy Data Management -> Generic CSV/XML Importer -> Configure customizing for generic XML/CSV Importer

This configuration is used to assign the Import ID to a parse and builder class. The Import ID is used as unique identifier for one Import. The parser and builder classes define which method is used to Import a File. The following figure depicts a typical setting for an XML Import.

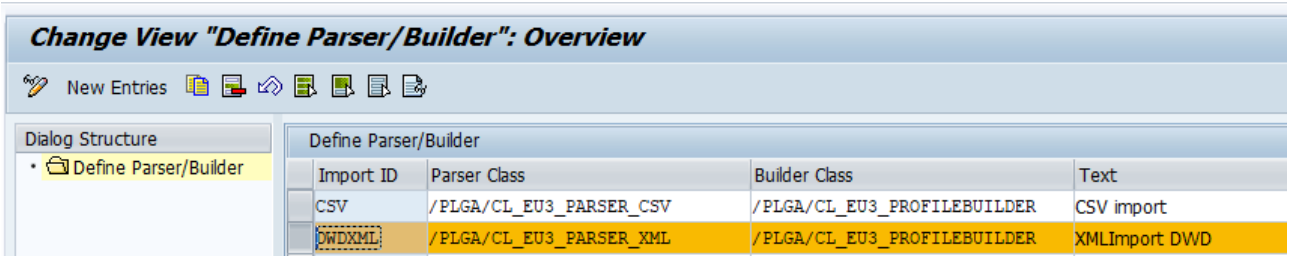


Figure 114: CSV/XML Importer: Parser/Builder configuration

The following table lists each Element of the customizing table and describes the Element and its usage within the Generic CSV/XML Importer.

ID	Description
Import ID	Defines import method
Parserclass	Defines which parser class to use for the Import
Builderclass	Defines which builder class to use for the Import
Text	Text to use as comment

Table 46:Table definition for /PLGA/TTUTIPB

2.7.1 Enhancement Spot /PLGA/ES_EU3_FILE

The enhancement spot currently consists of one business add-in, which is described in the following chapter.

2.7.1.1 BAdI /PLGA/BD_EU3_FILEHANDLER

The BAdI definition /PLGA/BD_EU3_FILEHANDLER is used to import the file, validate the filenames, get the data from CSV-files and manage the history table.

Method /PLGA/IF_EU3_FILEHANDLER~IMPORT_FILE

The method is used to validate the specified filename or import directory that was specified in the Report and check the history table for already existing importruns for the filename and importid. Every entry that isn't found within the history table is added to the internal table that is returned and specifies which files will be imported. The method has the following Signature parameters:

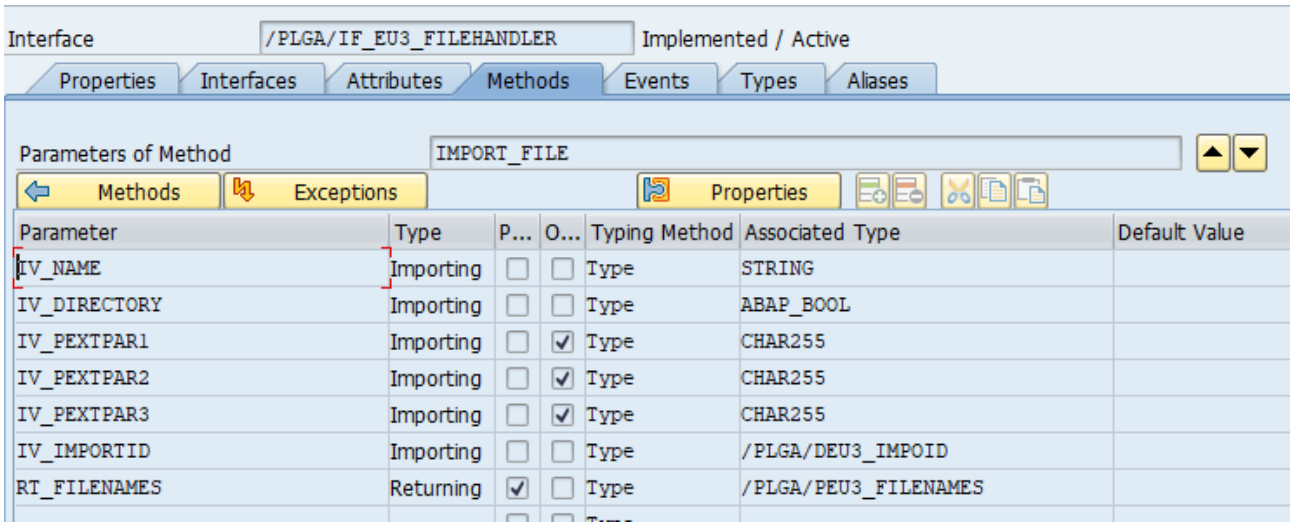


Figure 115: BAdI Method Import_File

Parameter	Dataelement	Description
IV_NAME	→ STRING	Filename or directoryname specified within the report as Import target
IV_DIRECTORY	→ABAP_BOOL	Defines if it's a directory import or Fileimport
IV_PEXTPAR1	→CHAR255	Parameter for File transaction
IV_PEXTPAR2	→CHAR255	Parameter for File transaction
IV_PEXTPAR3	→CHAR255	Parameter for File transaction
IV:IMPORTID	→/PLGEDM/DUTI_IMPOID	ID for current importrun (Import ID)
RT_FILLENAMES	←/PLGA/PEU3_FILENAMES	Table returned containing all the valid Filenames for the current importrun

Table 47: BAdI Method Import_File parameter

Method /PLGA/IF_EU3_FILEHANDLER~WRITE_IMPHIST

The method is used to insert Lines into the Table /PLGEDM/TTUTIIH with successfully imported Filenames of the current import. The method has the following Signature parameters:

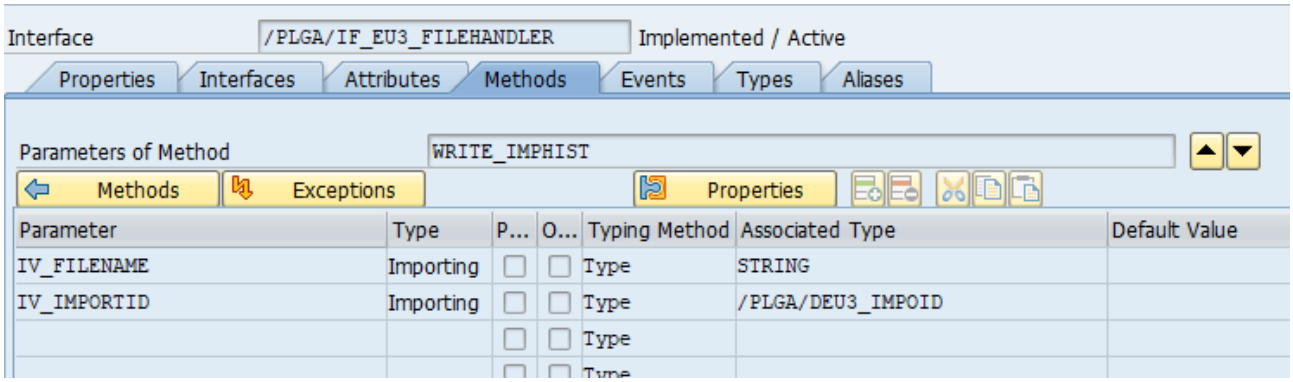


Figure 116: BAdI Method Write_Imphist

Parameter	Dataelement	Description
IV_FILENAME	→ STRING	Filename to be inserted into table
IV_IMPORTID	→ /PLGEDM/DUTI_IMPOID	ID for current Importrun (Import ID)

Table 48: BAdI Method Write_Imphist parameter

Method /PLGA/IF_EU3_FILEHANDLER~GET_CSV_DATA

The method is used to validate the defined Filenames and extract the CSV-Data into the returning table which is used for the Import. The method has the following Signature parameters:

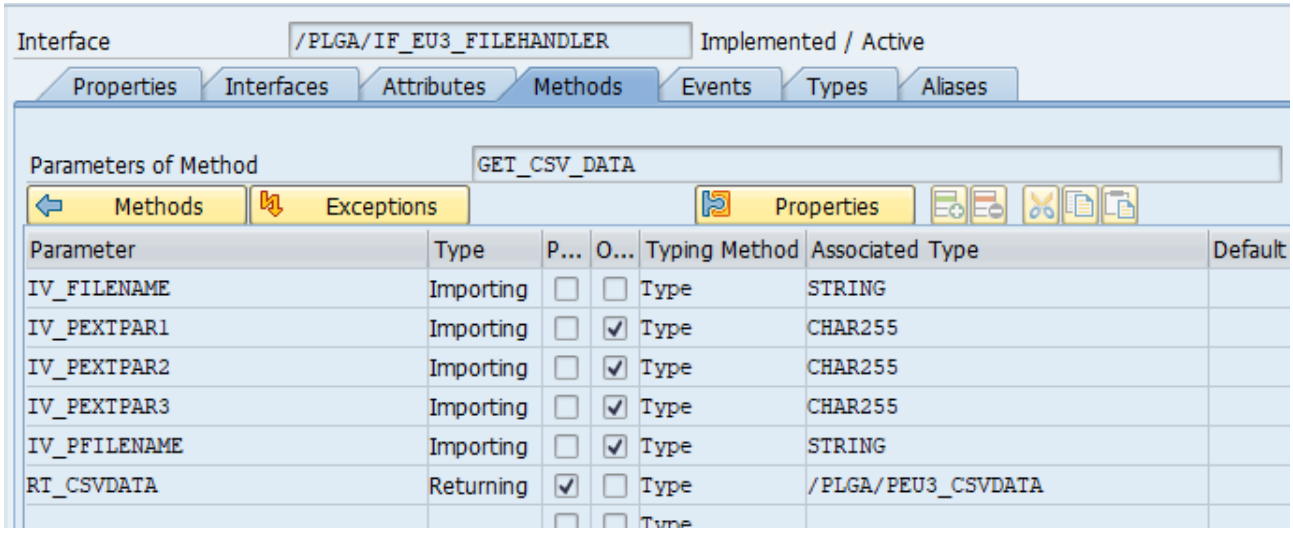


Figure 117:BAdI Method Get_CSV_Data

Parameter	Dataelement	Description
IV_FILENAME	→ STRING	Filename used to check if the File is UTF8 or not
IV_PEXTPAR1	→CHAR255	Parameter for File transaction
IV_PEXTPAR2	→CHAR255	Parameter for File transaction
IV_PEXTPAR3	→CHAR255	Parameter for File transaction
IV_PFILENAME	→STRING	Filename used to Validate the Filename and get the Filename from Filetransaction
RT_CSVDATA	←/PLGA/PEU3_CSVDATA	Returning table containing CSV-Data

Table 49: BAdI Method Get_CSV_Data parameters

To create an own implementation of the BAdI you can go to *SPRO -> SAP Customizing Implementation Guide -> Extension for Utilities by PROLOGA -> Energy Data Management -> Generic CSV/XML Importer -> BAdI for Fileimport and Historytable*

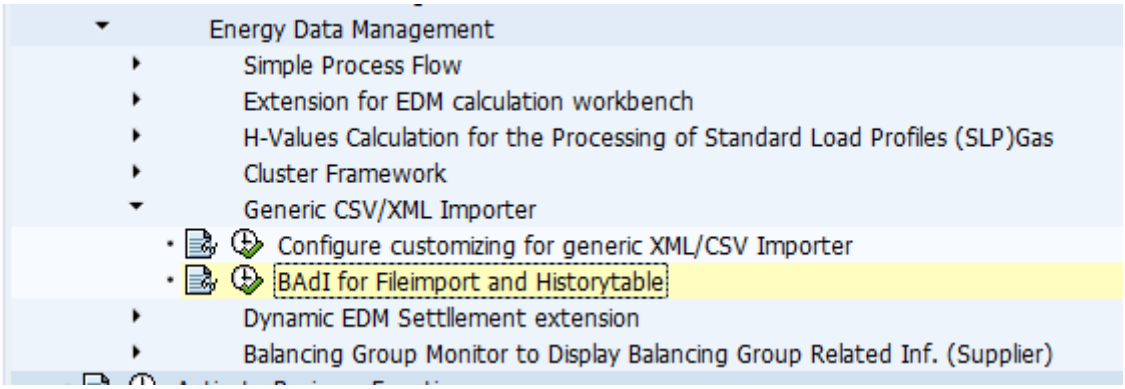


Figure 118: EDM Calculation Extension: SPRO-path for BAdI implementation

2.7.2 Simple Process Flow connector

The file importer can simply be integrated into the Simple Process Flow as a stand-alone node:

The selection screen parameter must be saved as a variant of the report /PLGA/RP_EU3_FILEIMPORT and a node has to be maintained according SPF node maintenance:

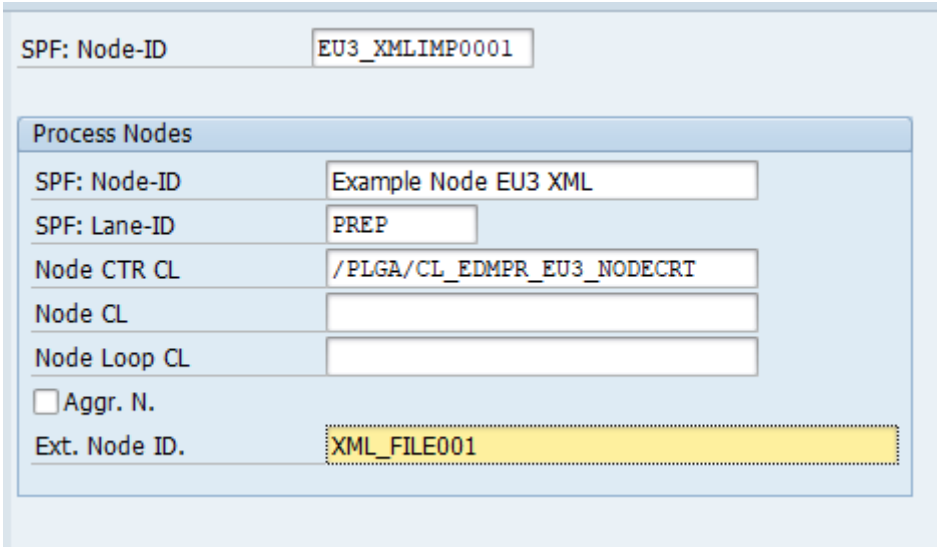


Figure 119: Example EU3 SPF connector

Lane-ID: The selected lane identification should be compatible to stand-alone nodes (e.g. lanes with Lane-Class */PLGA/CL_EDMPR_NOOP_LANECTR*)

The node control class is */PLGA/CL_EDMPR_EU3_NODECRT*

The External Node identification shall contain the technical name of a variant of the report /PLGA/RP_EU3_FILEIMPORT.

2.8 Dynamic EDM Settlement extension

This extension uses the *Simple Process Flow* to provide the possibility to create and integrate SAP EDM settlement processes only with customization efforts.

2.8.1 General Customizing

To support the manual creation of settlement documents, the *Simple Process Flow* engine needs a temporarily number object. This number range must be set via transaction *SNUM* in the object */PLGA/PRNO*. The number range "DM" has to be specified according to the following example:

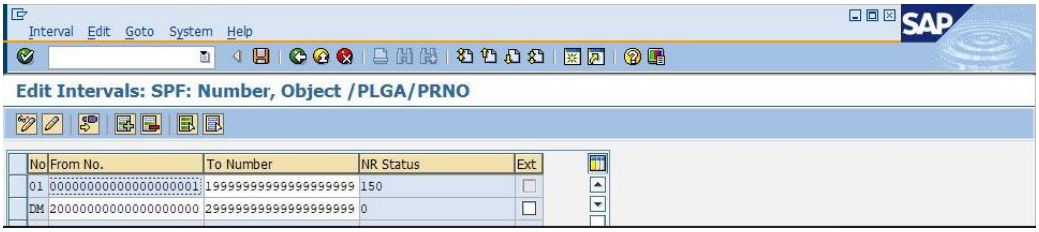


Figure 120: Number range SPF dynamic settlment

2.8.2 Usage requirements

To use this extension the following requirements must be fulfilled and prepared before starting the configuration:

- Completely customized settlement procedure with all dependencies (profile roles, parameters, selection types, settlement steps)
- Assignment of the "Program Name" */PLGA/RP_EDMSET_SETTLMENT* in the head data of the settlement procedure:

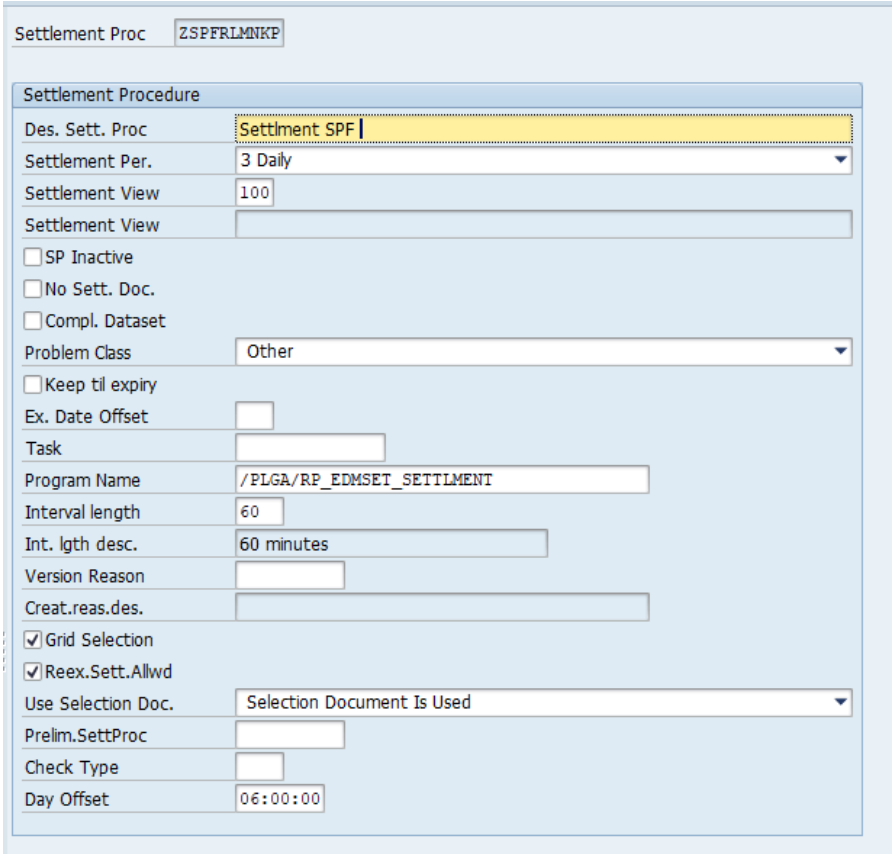


Figure 121: Example head data of settlement procedure

The settlement steps should only determine their data via input parameters. Within the dynamic settlement extension, the calculation results within the process document will be filtered accordingly to the customized input parameters of the specific settlement step.

2.8.3 Simple Process Flow configuration details

Shipped within the addon is the following lane controller class to manage the dynamic settlement extension in the *Simple Process Flow*:

Class: /PLGA/CL_EDMSET_LANE_CTRL

Method GET_INSTANCE

Instantiate itself and saves the process head data

Method PREPARE_LANE

Gets the instance of the current *Simple Process Flow* instance, and initially prepares the settlement document according to the setting in table */PLGA/TEDMSETCRT.*

Additionally, the process instance is bind via table /PLGA/TEDMSETIND

Method GET_AGGREGATION_DATA

This method builds an internal table of all following, aggregated nodes, which have a consecutive number

Method RUN_AGGREGATED_NODES

The method organises and calls every node which was identified in method GET_AGGREGATION_DATA.

Method POST_LANE_PROCESS

This method gracefully ends the settlement document.

Method DESTRUCT

The implementation returns "abap_false" to destroy the instance.

This class can be used to define a lane within the customizing of the *Simple Process Flow*:

Lanes

SPF: LID	SPF: Lane-ID Text	Pos.	Lane Class	SAP-Icon
CALC	Calculation	1	/PLGA/CL_EDMPR_NOOP_LANECTR	sap-icon://kpi-corporate-performance
CHECK	Check	2	/PLGA/CL_EDMPR_NOOP_LANECTR	sap-icon://kpi-corporate-performance
DATEX	Data Exchange	4	/PLGA/CL_EDMPR_NOOP_LANECTR	sap-icon://share
PREP	Preparation		/PLGA/CL_EDMPR_NOOP_LANECTR	sap-icon://order-status
SETTLMENT	Settlement	3	/PLGA/CL_EDMSET_LANE_CTRL	sap-icon://performance
SLEEP	Waiting Lane	5	/PLGA/CL_EDMPR_DLINE_LANECTR	sap-icon://future

Figure 122: Example Settlement Lane

To configure the correct progression of the settlement steps within the settlement procedure, every settlement step must be configured within the *Simple Process Flow* customizing view (Maintain Nodes) according to the following example:

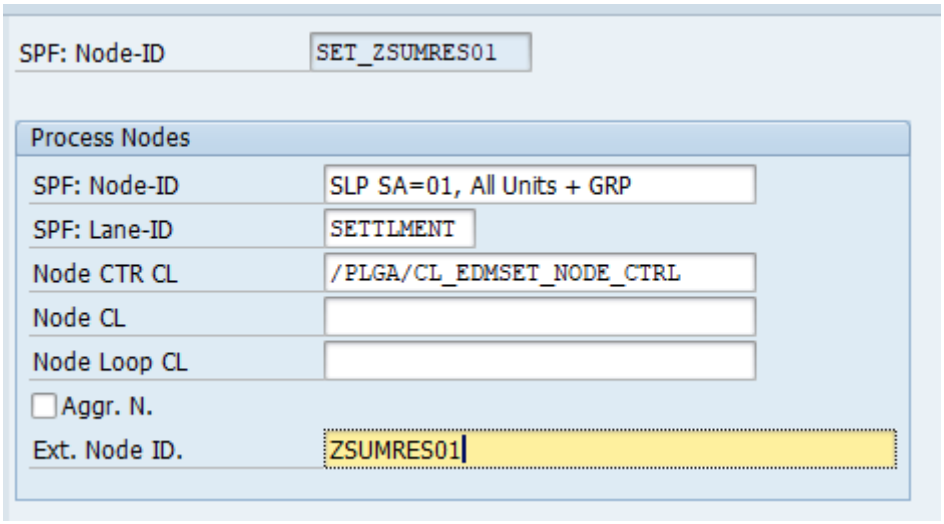


Figure 123: Settlement Step Node - Example

Class: /PLGA/CL_EDMSET_NODE_CTRL

The node controller class gets the customizing of the EDM settlement step and prepares the input parameters, as configured within the EDM settlement customization views.

The field "Ext. Node ID." Contains the settlement step name.

When the check box "Aggr. N." (aggregated node) is checked, the step will be looped over all settlement units within the selection of the settlement document.

2.8.4 Additional functionalities

This chapter shows additional functions shipped with the addon.

2.8.4.1 Stand-alone processing of settlement procedures

Every settlement procedure configured via the *Simple Process Flow* can run independently from a process instance, e.g. started from *EEDMSETTLCREATE.*

In this case, the process can not be monitored via the SPF monitoring application or will be logged via the SPF application log.

2.8.4.2 Settlement step: CDS-View-based grid-level aggregation

With the EDM settlement class */PLGA/CL_EDMSET_STEP_HDB_GLVL* the addon gives the opportunity to use CDS-View based aggregations of interval metered profiles with the resolution of the grid level (attribute to point of deliveries).

The following figure shows an example of a settlement step, which summarizes points of delivery with grid level "G1" in one parameter and grid level "G2" and "G3" in another settlement parameter:

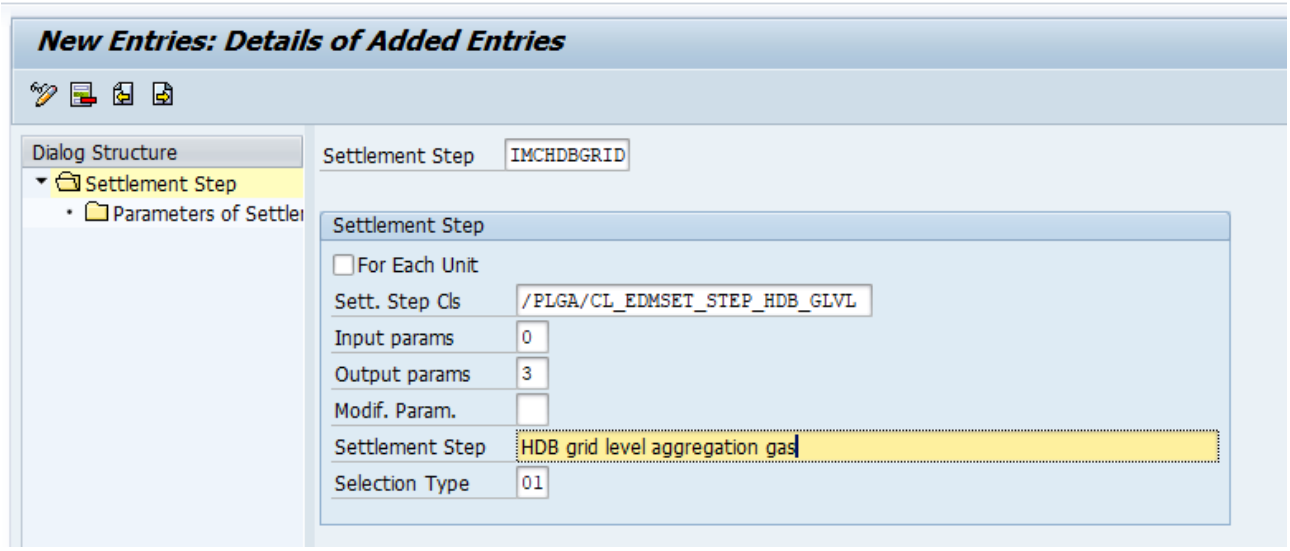


Figure 124: HDB GLVL Settlstep head

The first parameter is always the complete sum of all grid level specific parameters, therefore we have three output parameters in our example.

The association of parameter an grid level identification hat to be done in the following SPRO-Path:

SAP Customizing Implementation Guide -> Energy Data Management -> Dynamic EDM Settllement extension -> EDM Settlement Extension: Gridlevel Aggregation

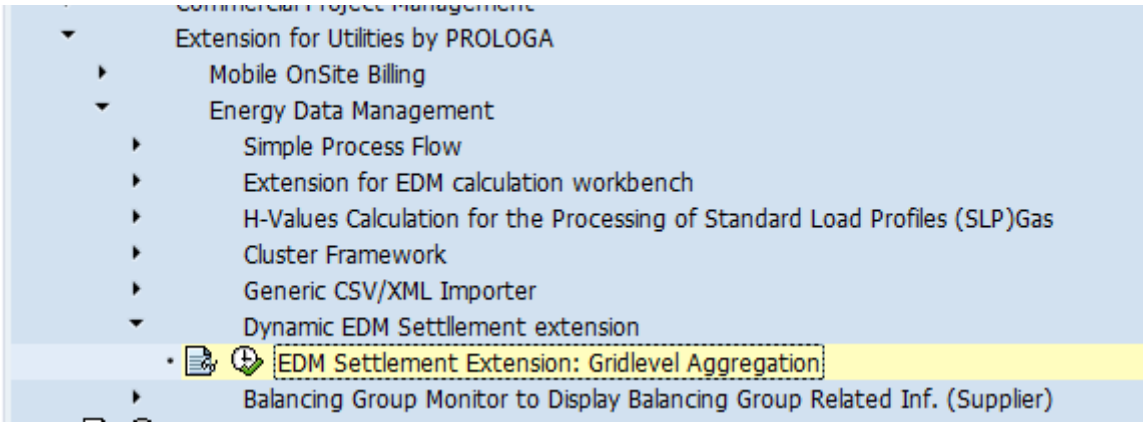


Figure 125: HDB GLVL Settlstep configuration

The following data must be entered to fulfil the current example:

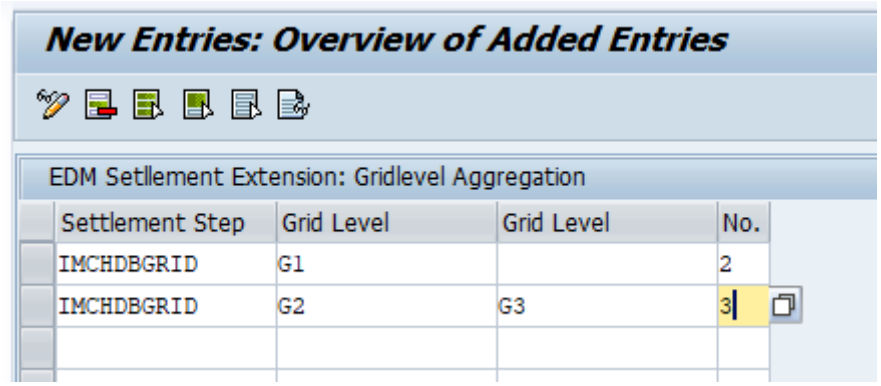


Figure 126: HDB GLVL Settlstep configuration example entries

The settlestep can be used in any settlement procedure and is not limited to the usage within the dynamic settlement extension.

2.9 Balancing Group Monitor to Display Balancing Group Related Information from Supplier's Point of View

The customizing options for the balancing group monitor are described in the following.

Customizing for the balancing group monitor will be started within the SAP® Reference IMG by using the transaction *SPRO.* The maintenance for SLP is performed using the path: *SAP® Customizing Implementation Guide -> Extension for Utilities by Prologa -> Energy Data Management -> Balancing Group Monitor to Display Balancing Group Related Inf. (Supplier)*

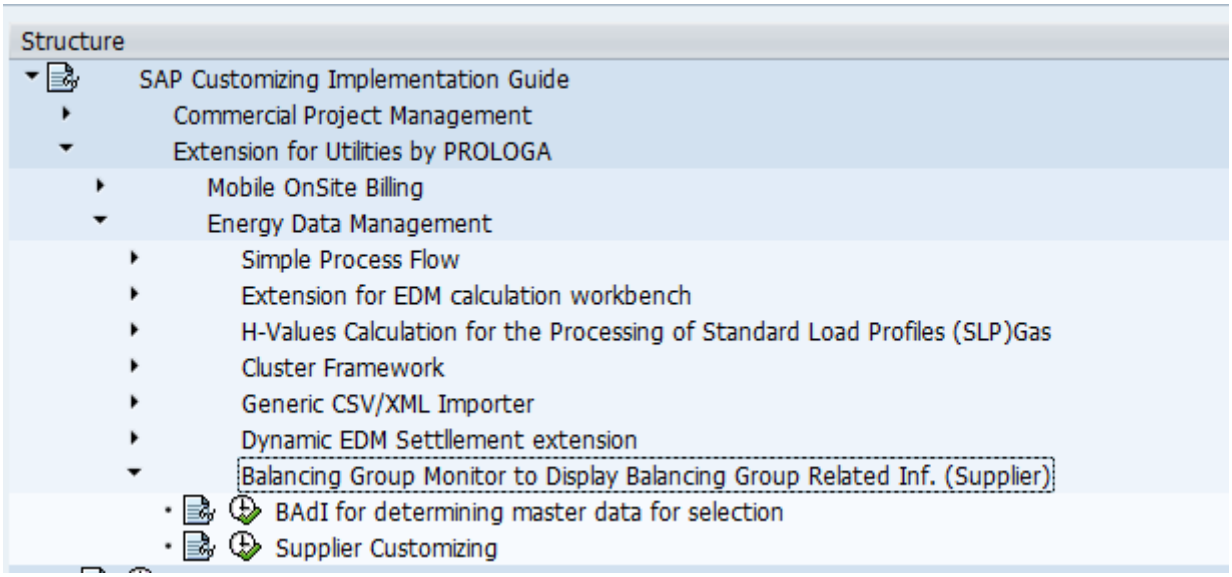


Figure 127: Customizing path for Extension for Utilities by Prologa

The maintenance of the Balancing Group Monitor is performed by using the path *SAP® Customizing Implementation Guide -> Extension for Utilities by Prologa -> Energy Data Management -> Balancing Group Monitor to Display Balancing Group Related Inf. (Supplier) -> Supplier Customizing*

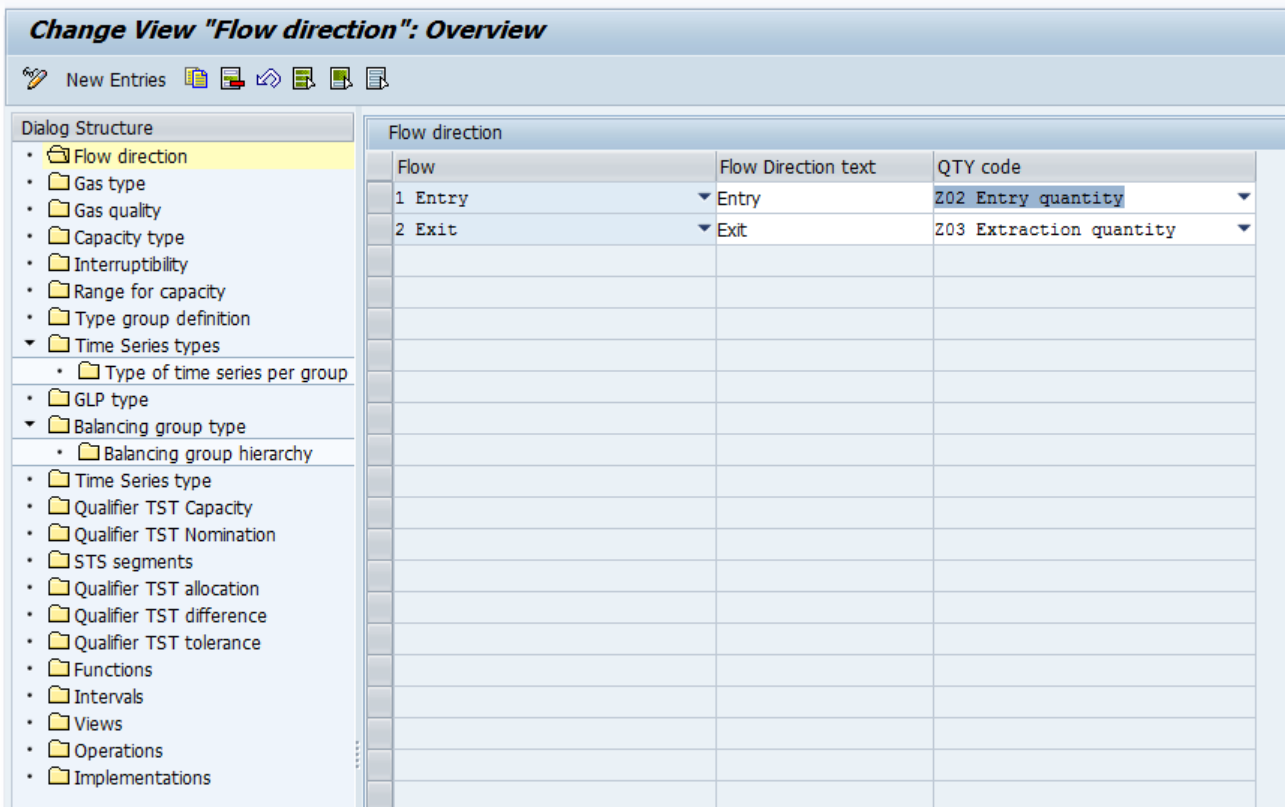


Figure 128: Dialog structure Balancing Group Monitor View Cluster

The configurations are described in detail in the following.

2.9.1 Flow Direction

Table: */PLGA/TTESLF*

Text Table: */PLGA/TTESLFT*

Description: Flow direction

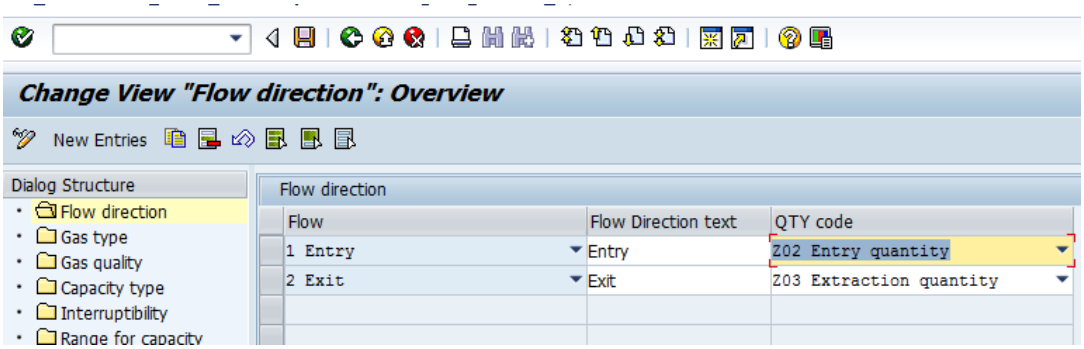


Figure 129: Customizing Example: Flow Direction

Field	Data element	Description
Flow	/PLGA/DESL_FLOW_ID	Flow direction key
Flow Direction text	/PLGA/DESL_CAP_FD_TEXT	Text field for flow direction
Code QTY	/PLGA/DESL_QUAL_MENGE	EDIFACT Code for qualifying the quantity type

Table 50: Flow Direction

2.9.2 Gas Type

Table: */PLGA/TTESLB*

Text Table*: /PLGA/TTESLBT*

Description: Gas type

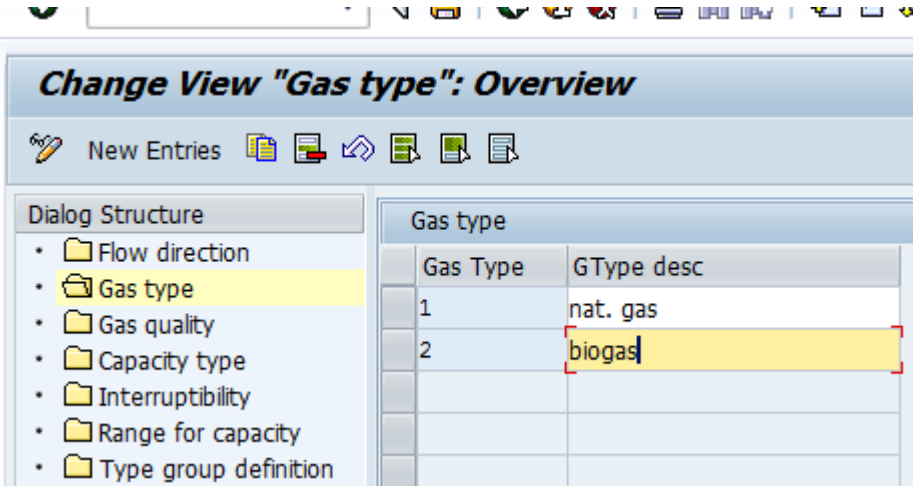


Figure 130: Customizing Example: Gas Type

Field	Data element	Description
Gas Type	/PLGA/DESL_ART_ID	Gas type key
GType desc	/PLGA/DESL_CAP_GA_TEXT	Gas Type Text

Table 51: Gas Type

2.9.3 Gas Quality

Table: */PLGA/TTESLH*

Text Table: */PLGA/TTESLHT*

Description: Gas Quality

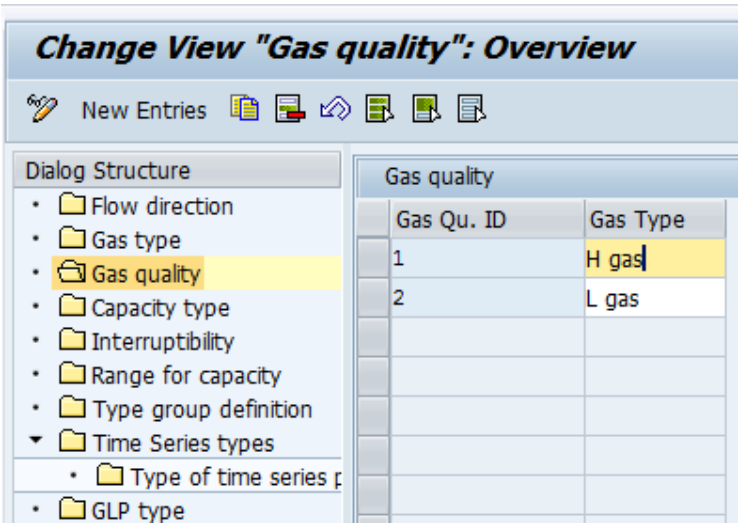


Figure 131: Customizing Example: Gas quality

Field	Data element	Description
Gas qu. ID	/PLGA/DESL_QUAL_ID	Gas quality key

Gas Type	/PLGA/DESL_CAP_GT_TEXT	Text field for gas type

Table 52: Gas quality

2.9.4 Capacity Type

Table: */PLGA/TTESLA*

Text Table: */PLGA/TTESLAT*

Description: Capacity Type

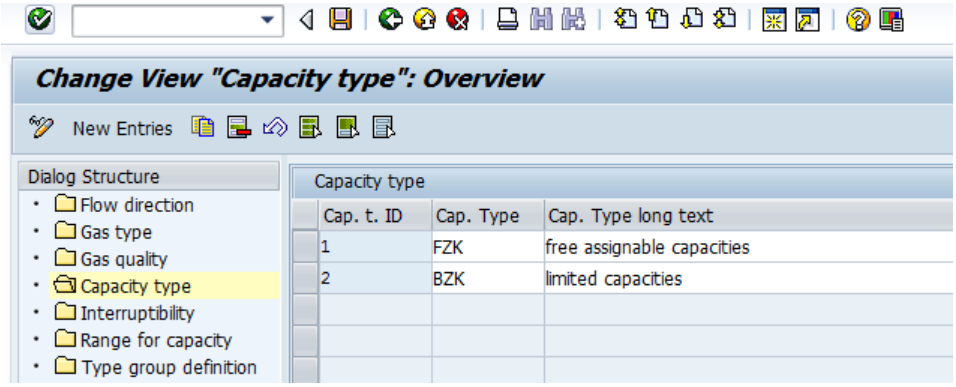


Figure 132: Customizing example: Capacity type

Field	Data element	Description
Cap ID	UGAS/CS_CAPART_ID	Flow direction key
Cap. Type	/CS4UGAS/CS_CAP_CA_TEXT	Abbreviation for capacity type
Cap. Type long text	/CS4UGAS/CS_CAP_LANGTEXT	Description for capacity type
BGM segm	/PLGA/DESL_BGM_CHACAP	BGM segment CHACAP
STS segm.	/PLGA/DESL_STS_CHACAP	STS segment CHACAP
IMD segm.	/PLGA/DESL_IMD_CHACAP	IMD segment CHACAP

Table 53: Capacity Type

2.9.5 Interruptibility

Table: */PLGA/TTESLI*

Text Table: */PLGA/TTESLIT*

Description: Interruptibility

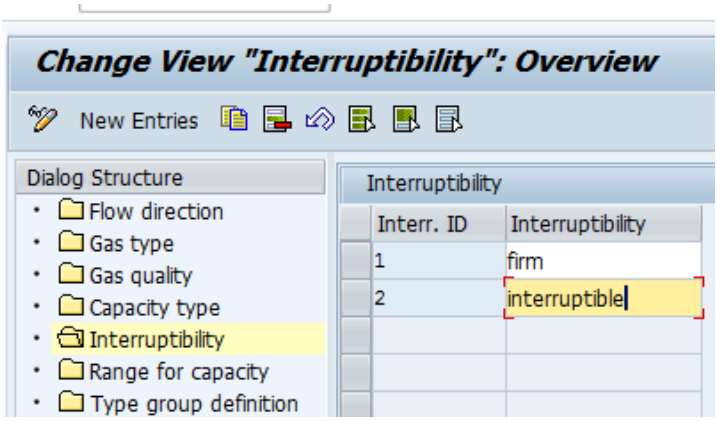


Figure 133: Customizing Example: Interruptibility

Field	Data element	Description
Interrupt ID	/PLGA/DESL_INT_ID	Interruptibility key
Interruptibility	/PLGA/DESL_CAP_I_TEXT	Text field for interruptibility

Table 54: Interruptibility

2.9.6 Interval Length of Capacity Products

Table: */PLGA/TTESLP*

Text table: */PLGA/TTESLPT*

Description: Interval length for capacity

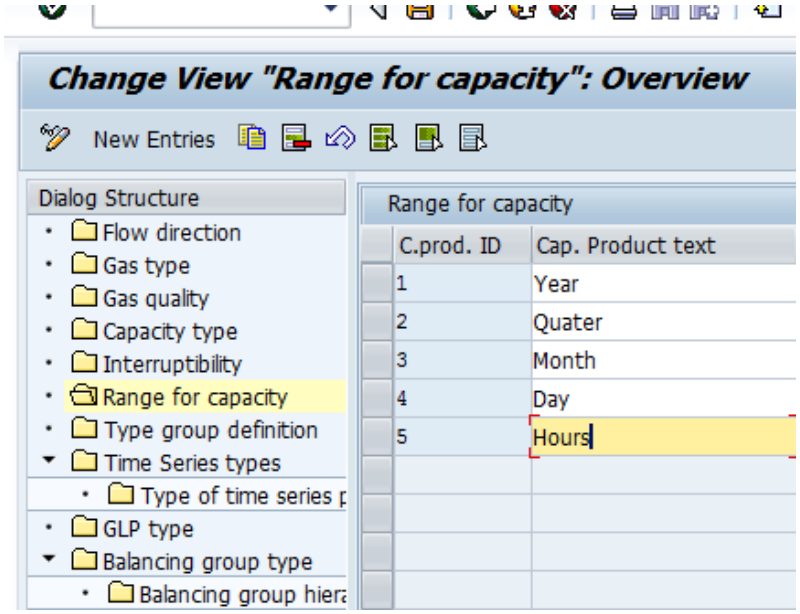


Figure 134: Customizing Example: Interval Length of the Capacity Product

Field	Data element	Description
Product ID	/PLGA/DESL_PROD_ID	Interval length key
Capacity product Text	/PLGA/DESL_CAP_CP_TEXT	Text field for interval length

Table 55: Interval Length of Capacity Products

2.9.7 Interval Length of Capacity Products

Table: */PLGA/TTESLG*

Text table: */PLGA/TTESLGT*

Description: Type groups

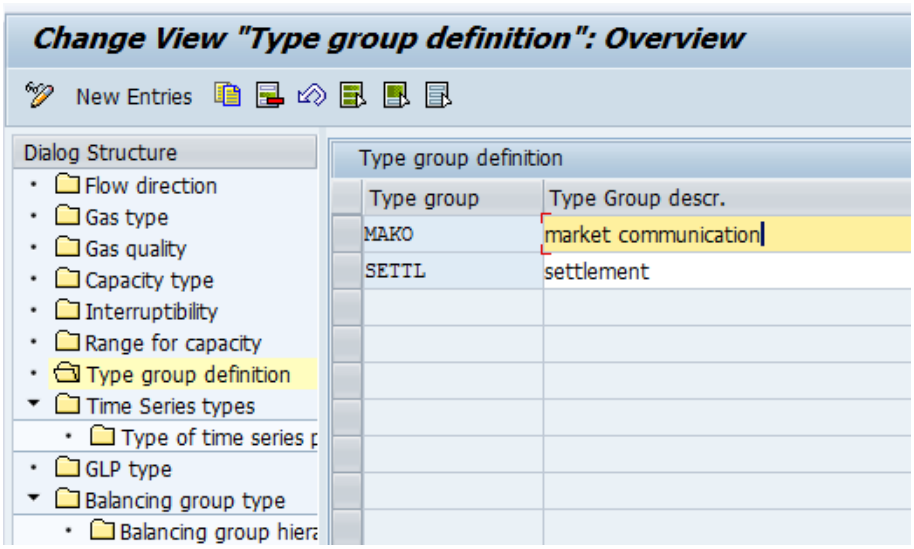


Figure 135: Customizing Example: Interval Length of the Capacity Product

Field	Data element	Description
Type group	/PLGA/DESL_GROUPID	Type group
Type Group descry.	/PLGA/DESL_GROUPID_TEXT	Type Group description

Table 56: Interval Length of Capacity Products

2.9.8 Type of Time Series

Table: */PLGA/TTESLR*

Text Table: */PLGA/TTESLRT*

Description: Type of time series

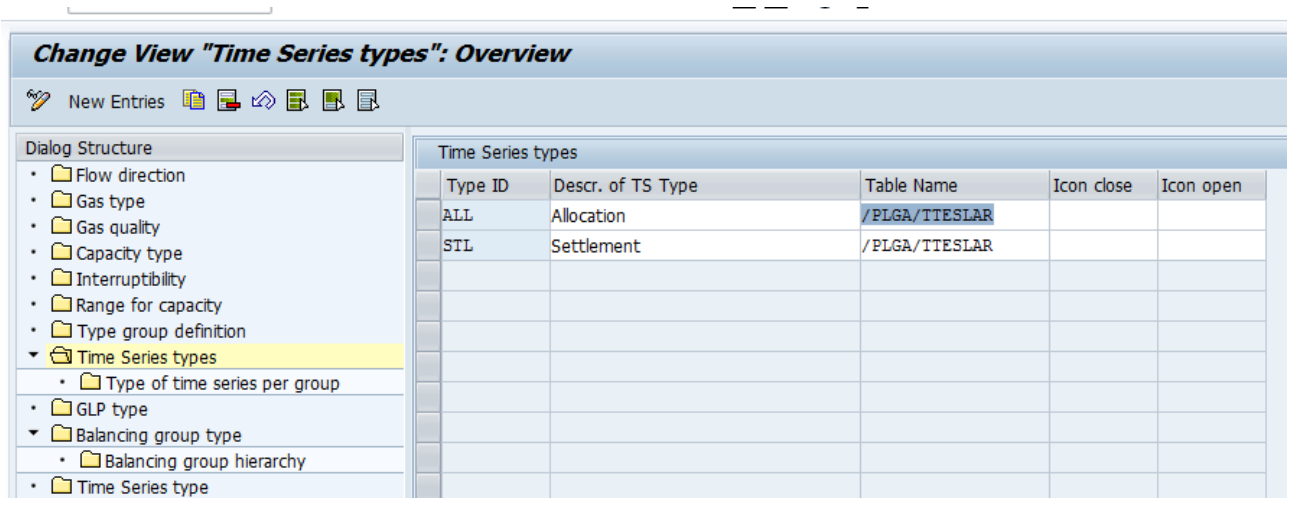
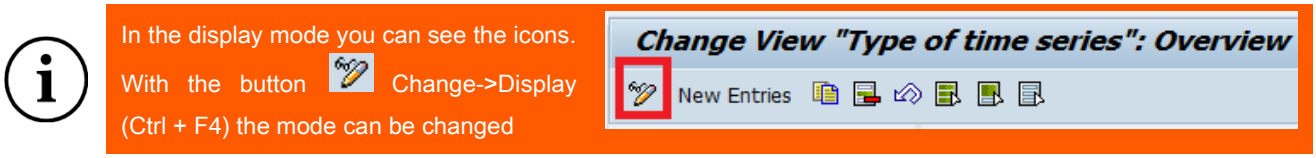


Figure 136: Customizing Example: Type of Time Series (Changing Mode)

Field	Data element	Description
Type ID	/PLGA/DESL_RTYPE	Type of ZRT
Description of TS Type	/PLGA/DESL_TYPE_ZRT_TEXT	Description of the Type ZRT
Table name	/PLGA/DESL_TABLE_NAME	The qualifier table of the type
Icon close	/PLGA/DESL_ICON_CLOSED	Icon for closed nodes in the balancing group screen of the balancing group monitor

Icon open	/PLGA/DESL_ICON_OPEN	Icon for opened nodes in the balancing group screen of the balancing group monitor. Search help see icon close **If icon opened is not maintained, the icon closed is selected automatically**

Table 57: Type of Time Series



2.9.9 Type of time series type per group

Table: */PLGA/TTESLG_T*

Description: Assignment from time series types to type group

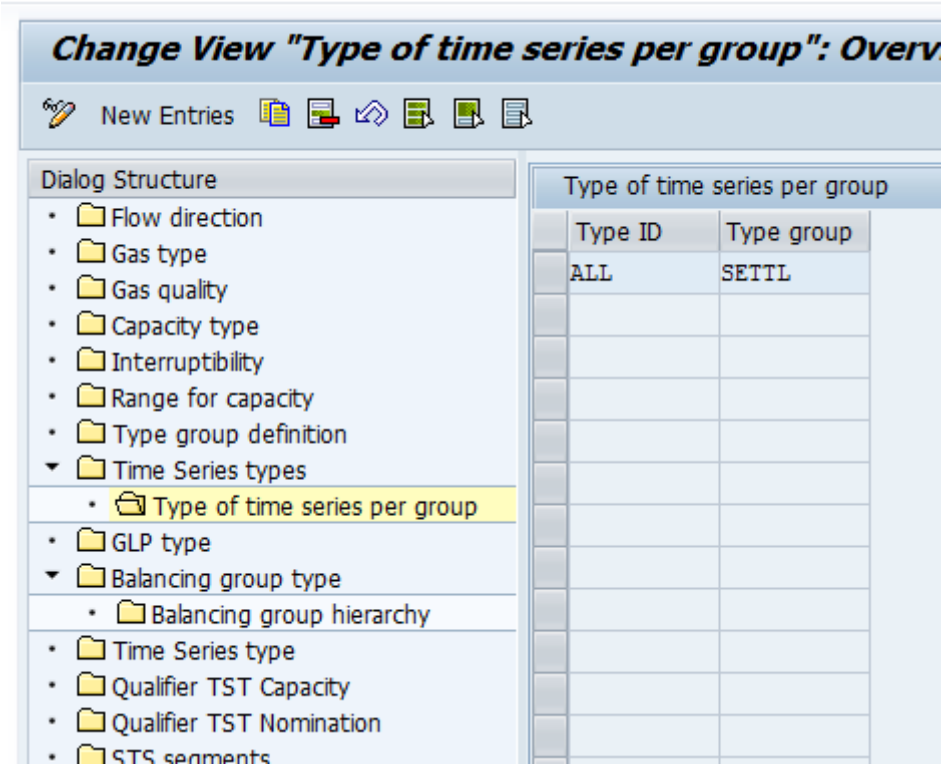


Figure 137: Customizing Example: Type of time series per group

Field	Data element	Description
Type ID	/PLGA/DESL_RTYPE	Type of time series type
Type Group	/PLGA/DESL_GROUPID	Type group

Table 58: Interval Length of Capacity Products

2.9.10 Type of the Grid Linking Point

Table: */PLGA/TTESLK*

Text Table*: /PLGA/TTESLKT*

Description: Type of grid link point

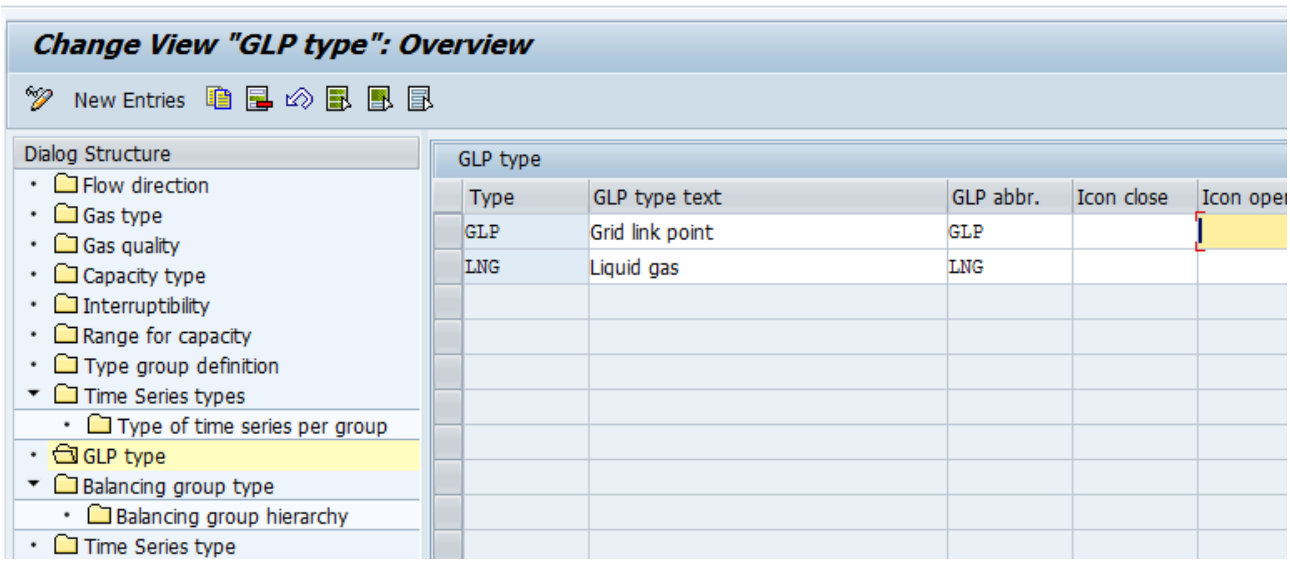


Figure 138: Customizing Example: Type of GLP

Field	Data element	Description
Type	/PLGA/DESL_NKP_TYPE_ID	Key field for GLP type
GLP Type text	/PLGA/DESL_CAP_NA_TEXT	Description of Types of GLP
GLP abbrev.	/PLGA/DESL_NKP_TYPE_SHORT	Abbreviation of the type of GLP **The shortcut can be identical with the key for the first customizing, but for further languages it will differ from the key.**
Icon Close	/PLGA/DESL_ICON_CLOSED	Icons for closed nodes in the balancing group screen of the balancing group monitor
Icon Open	/PLGA/DESL_ICON_OPEN	Icons for opened nodes in the balancing group screen of the balancing group monitor. Search help see icon close. **If icon opened is not maintained, the icon closed is selected automatically**

Table 59: Type of GLP

2.9.11 Balancing Group Type

Table: */PLGA/TTESLY*

Text Table: */PLGA/TTESLYT*

Description: Balancing Type

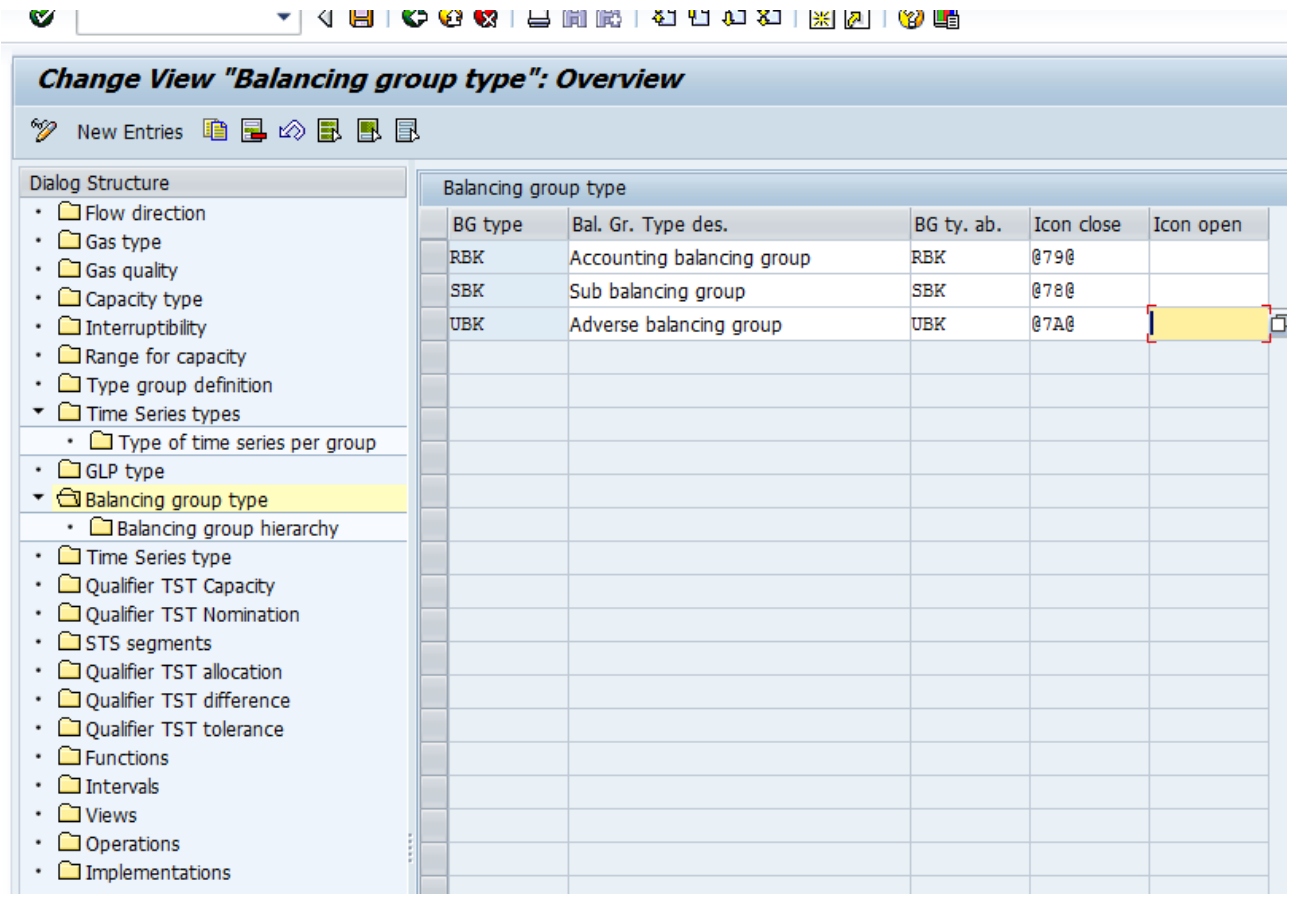


Figure 139: Customizing Example: Balancing Group Type

Field	Data element	Description
BG Type	/PLGA/DESL_BK_TYPE_ID	Key field for balancing group key
Descr. BG Type	/PLGA/DESL_BKTYPE_TEXT	Description of the type of the GLP
Abbrev.	/PLGA/DESL_BK_TYPE_SHORT	Abbreviation of the type of GLP **The shortcut can be identical with the key for the first customizing, but for further languages it will differ from the key.**
Icon Close	/PLGA/DESL_ICON_CLOSED	Icons for closed nodes in the balancing group screen of the balancing group monitor
Icon Open	/PLGA/DESL_ICON_OPEN	Icons for opened nodes in the balancing group screen of the balancing group monitor. Search help see icon close. **If icon opened is not maintained, the icon closed is selected automatically**

Table 60: Balancing Group Type

2.9.12 Balancing Group Type Hierarchy

Table: */PLGA/TTESLYH*

Description: Balancing Group Type Hierarchy

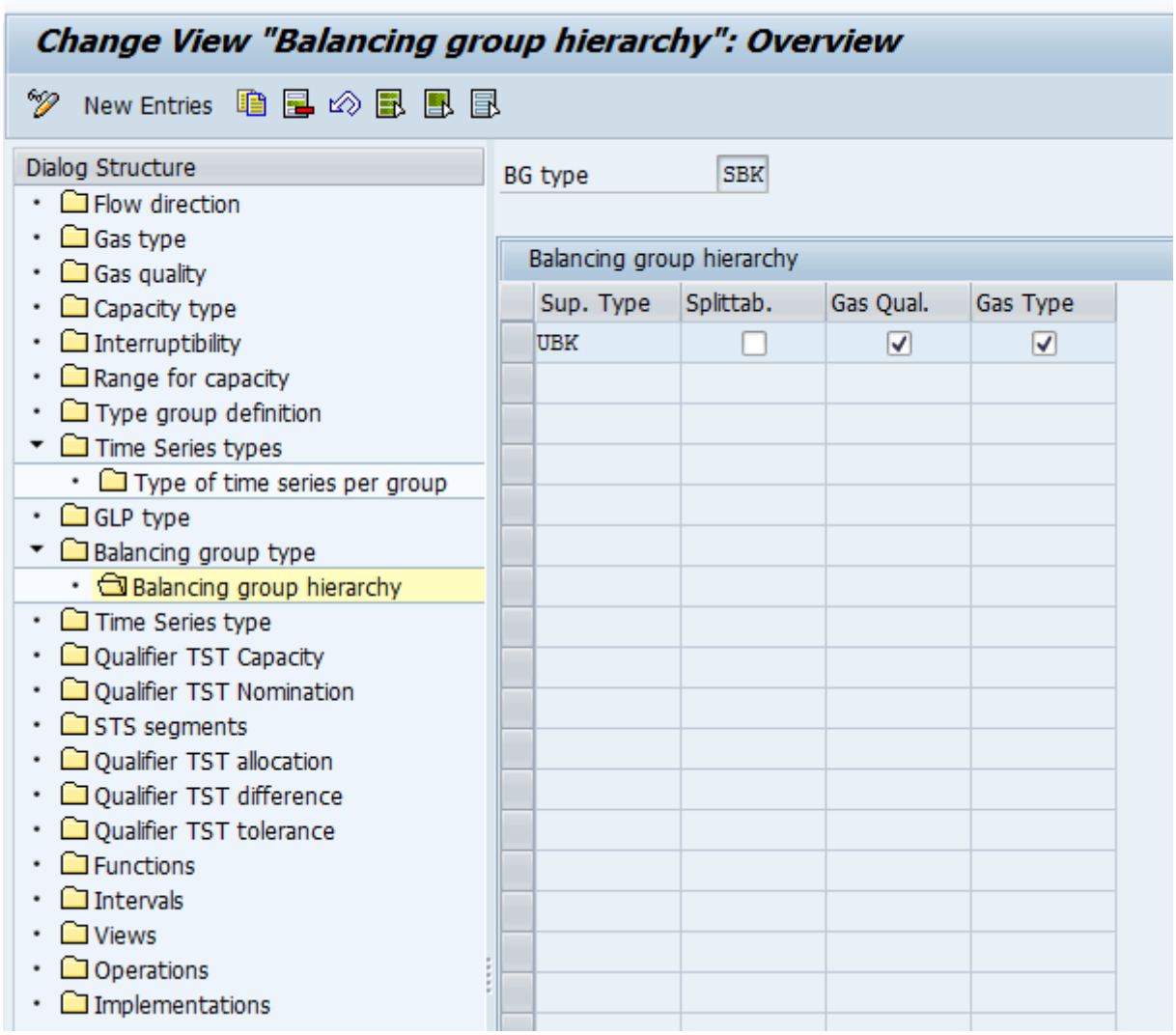


Figure 140: Customizing Example: Balancing Group Type Hierarchy

Field	Data element	Description
Sup. Type	/PLGA/DESL_BK_TYPE_PID	Upper level balance group type
Splittable	/PLGA/DESL_IS_ANTEIL	Indicator for percentage allocation
Gas Qual.	/PLGA/DESL_IS_NK_CHECK	Check gas quality (H, L) for the BK compound
Gas type	/PLGA/DESL_IS_NK_CHECK_ART	Check gas type (Bio, natural gas, ...) for the BG compound

Table 61: Balancing Group Type Hierarchy

The following entries are currently necessary for the German market:

- RBK cannot be subordinated (no entry)
- UBK can be subordinated to RBK and UBK

For RBK: a proportional allocation is possible, the gas quality has to match, and the gas type has to match.

For UBK: a proportional allocation is not possible, the gas quality does not have to match, and gas type does not have to match

SBK can be subordinated to UBK

Proportional allocation is not possible, however: the gas quality

and type must match

2.9.13 Time Series Type

Table: */PLGA/TTESLZ*

Text Table: */PLGA/TTESLZT*

Description: Time series types

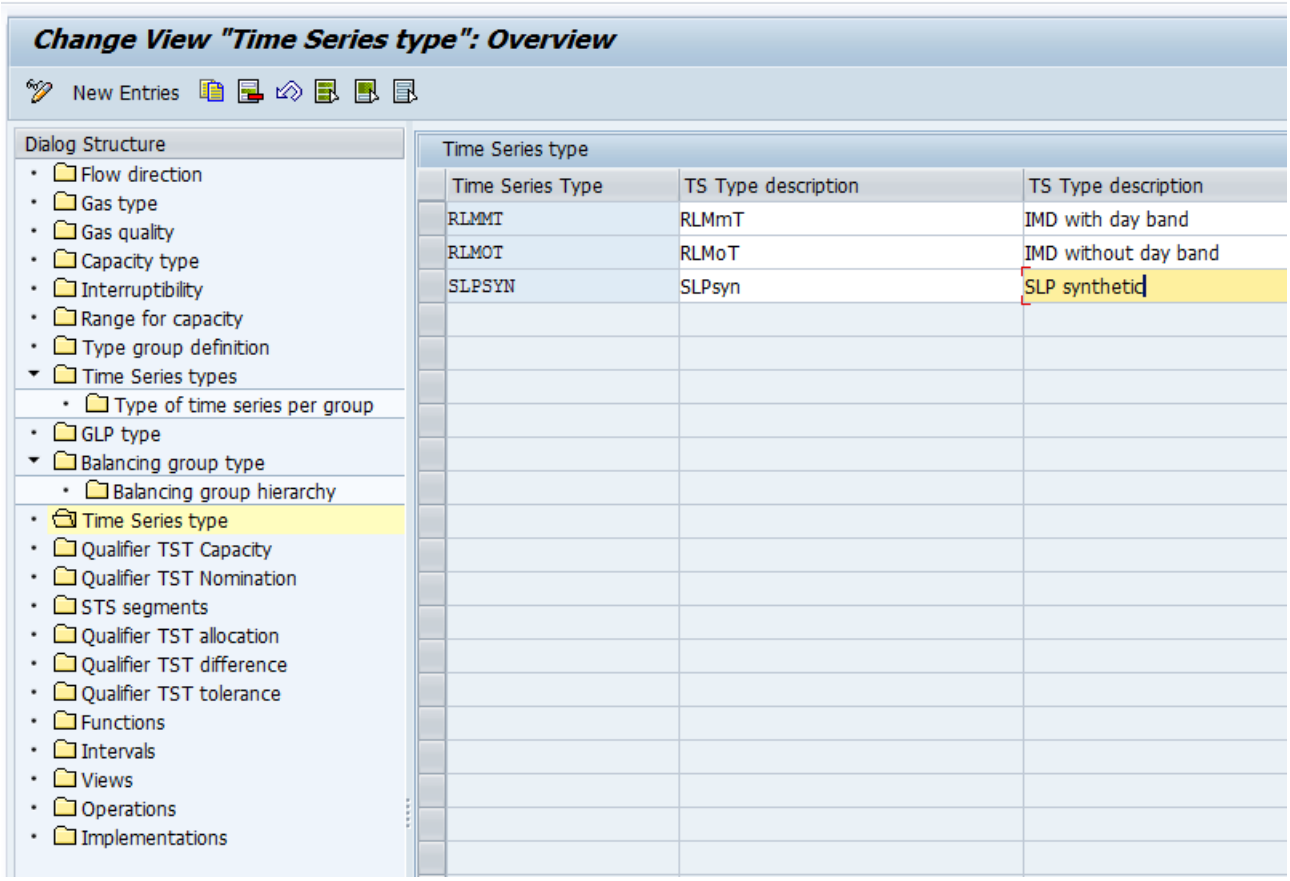


Figure 141: Customizing Example: Time Series Type

Field	Data element	Description
Time series type	/PLGA/DESL_ZRT	Time series type key
Time series type	/PLGA/DESL_ZRT_TEXT	Description for TST
Description TST	/PLGA/DESL_ZRT_LONG	Description for TST

Table 62: Time Series Type

Except for the usual TST for market communication, further TST for aggregated quantities are needed for balancing the group monitor.

2.9.14 Qualifier TST Capacity

Qualifiers are characteristics that describe time series type dependents with more details.

Table: */PLGA/TTESLCR*
Description: Qualifier ZRT Capacity

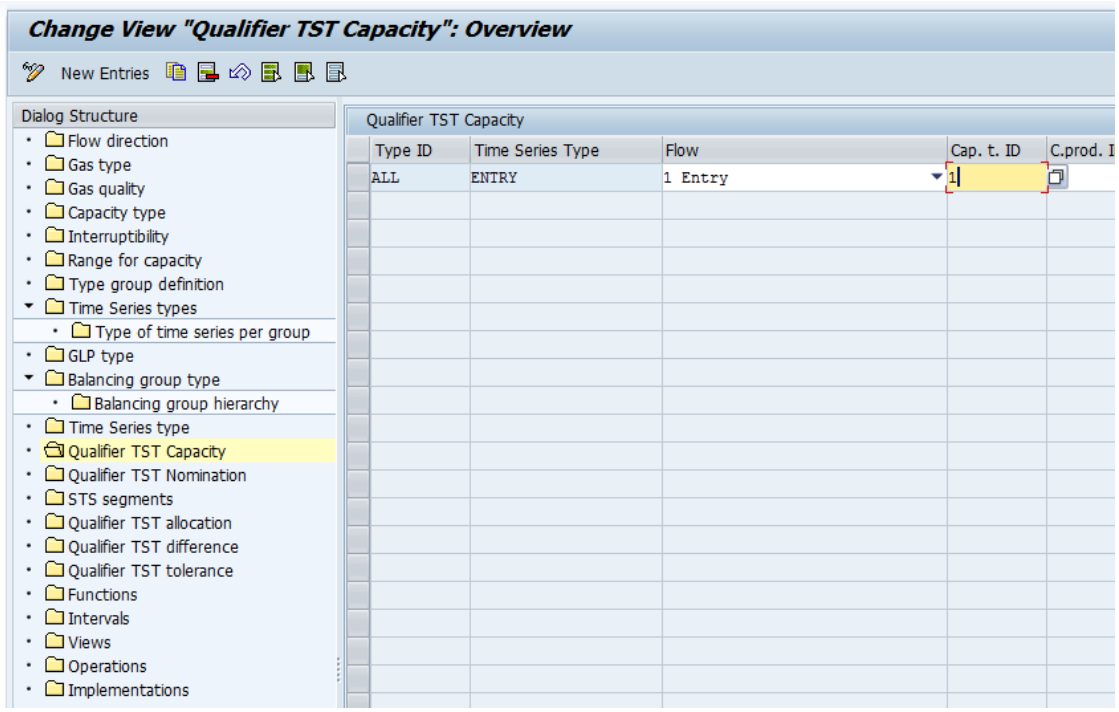


Figure 142: Customizing Example: Qualifier TST Capacity

Field	Data element	Description
Type ID	/PLGA/DESL_RTYPE	Type of TST
Time series type	/PLGA/DESL_ZRT	Time series type
Flow	/PLGA/DESL_FLOW_ID	Flow direction (Entry, Exit)
Cap ID	/PLGA/DESL_CAPART_ID	Capacity type (FZK, BZK, …)
Product ID	/PLGA/DESL_PROD_ID	Product interval length (Year, Quarter, …)
Interrupt ID	/PLGA/DESL_INT_ID	Interruptibility
T/F	/PLGA/DESL_IS	Nomination responsibility (X)
Position	/PLGA/DESL_POS	Order per flow direction from down to up for the stacked graphical view **TST with different hourly values (e.g. without daily band) should be the last ones**

Table 63: Qualifier TST Capacity

2.9.15 Qualifier TST Nomination

Table: */PLGA/TTESLNR*
Description: Qualifier TST Nomination

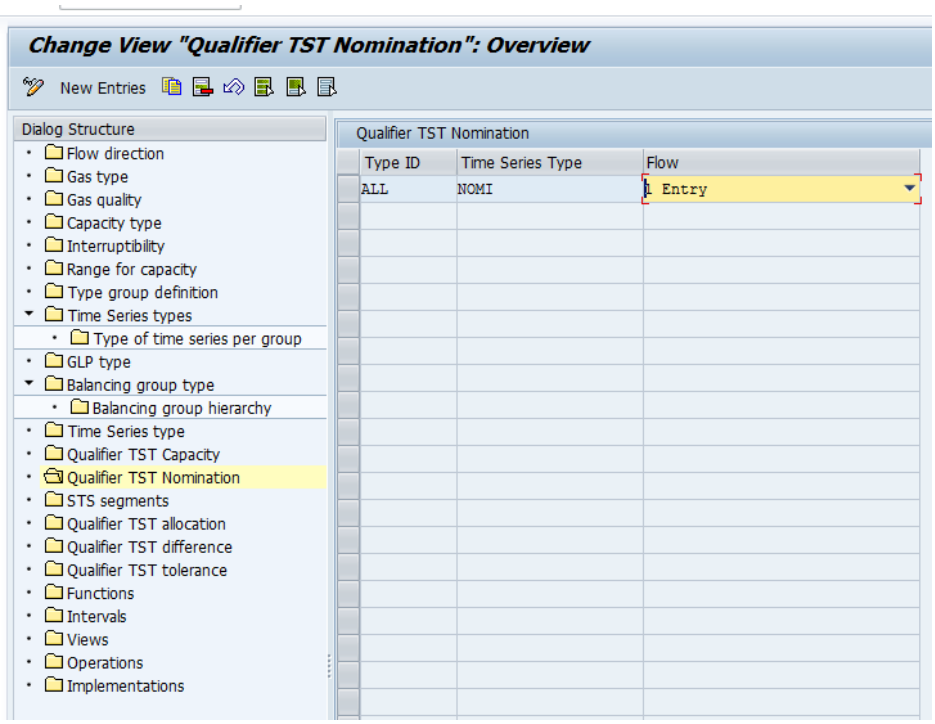


Figure 143: Customizing Example: Qualifier TST Nomination

Field	Date element	Description
Type ID	/PLGA/DESL_RTYPE	Type of TST
Time series type	/PLGA/DESL_ZRT	Time series type
Direction	/PLGA/DESL_FLOW_ID	Flow direction (Entry, Exit)

Table 64: Qualifier TST Nomination

2.9.16 STS segments

Table: */PLGA/TTESLSTS*

Text Table: */PLGA/TTESLSTST*

Description: STS segments

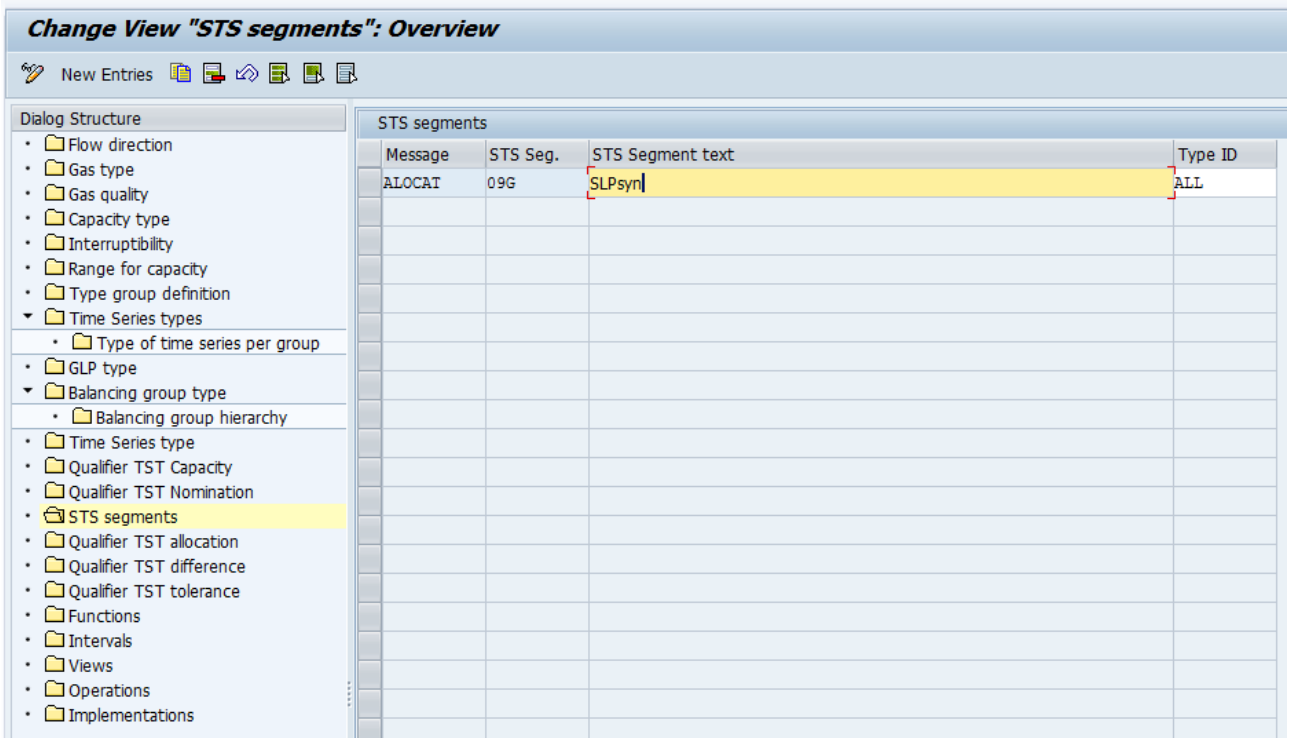


Figure 144: Customizing Example: STS segments

Field	Date element	Description
Message	/PLGA/DESL_MSG_TYPE	Message type
STS Seg.	/PLGA/DESL_STS	STS Segment
STS Segment text	/PLGA/DESL_STSTEXT	STS segment text
Type ID	/PLGA/DESL_RTYPE	Type ID

Table 65: Qualifier TST Nomination

2.9.17 Qualifier TST Allocation

Table: */CS4UGAS/CS_AR*

Description: Qualifier TST Allocation

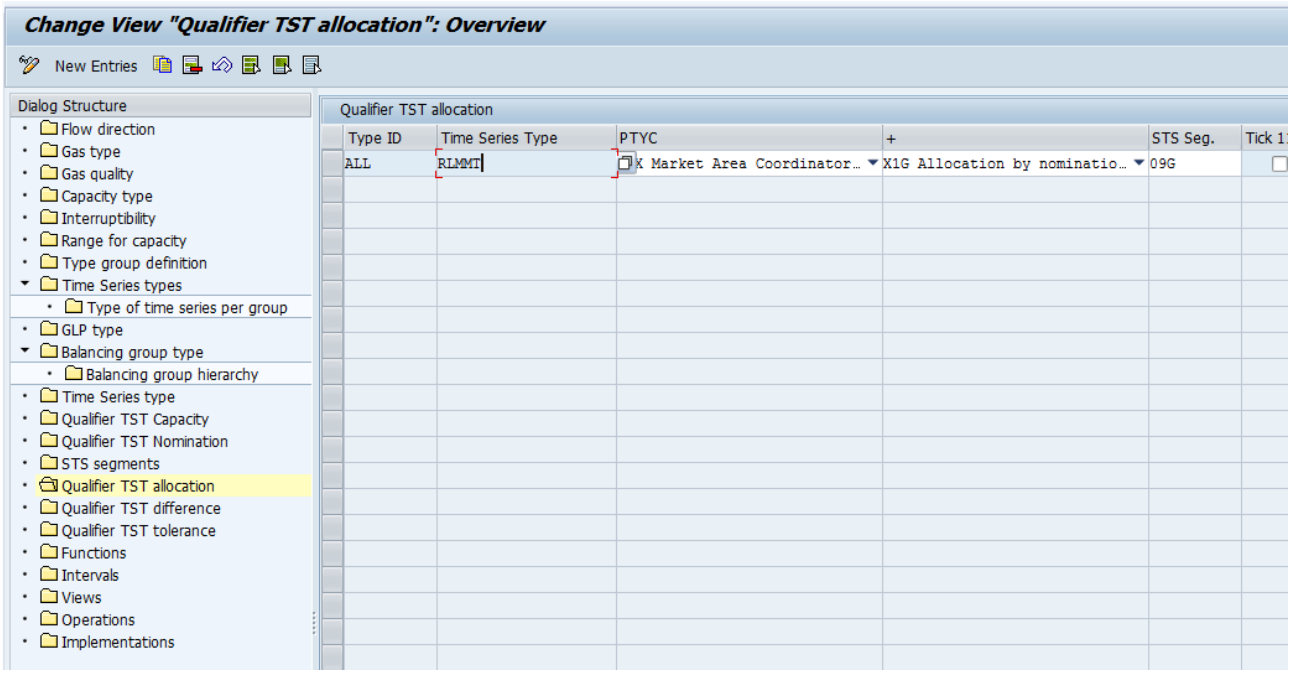


Figure 145: Customizing Example: Qualifier TST Allocation

Field	Data element	Description
Type ID	/CS4UGAS/CS_ZRT	Time series type
Time Series Type	/CS4UGAS/CS_STSQUAL	Data element for STS segment ALOCAT
PTYC	/CS4UGAS/CS_FLOW_ID	Flow direction (Entry, Exit)
+	/CS4UGAS/CS_POS	Order per flow direction from down to up for the stacked graphical view **TST with different hourly values (e.g. without daily band) should be the last ones**
STS Seg.		
Tick 11G		
Flow		
Position		

Table 66: Qualifier TST Allocation

2.9.18 Qualifier TST Difference

Table: */PLGA/TTESLDR*

Description: Qualifier TST Difference

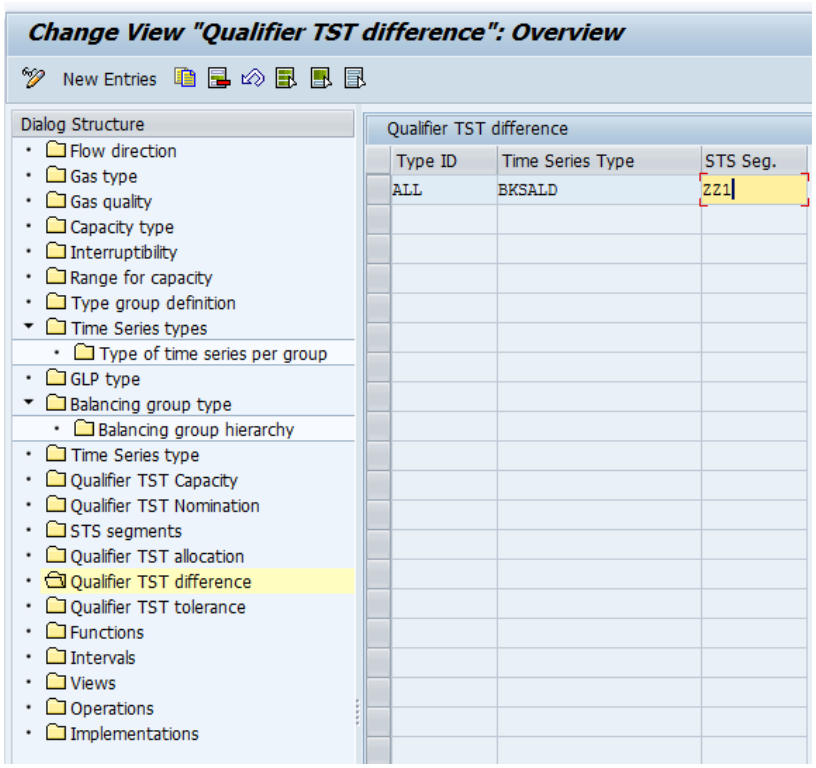


Figure 146: Customizing Example: Qualifier TST difference

Field	Data element	Description
Type ID	/PLGA/DESL_RTYPE	Type ID
Time Series Type	/PLGA/DESL_ZRT	Time Series Type
STS Seg	/PLGA/DESL_STS	STS Segment

Table 67: Qualifier TST difference

2.9.19 Qualifier TST Tolerance

Table: */PLGA/TTESLTR*

Description: Qualifier TST Tolerance

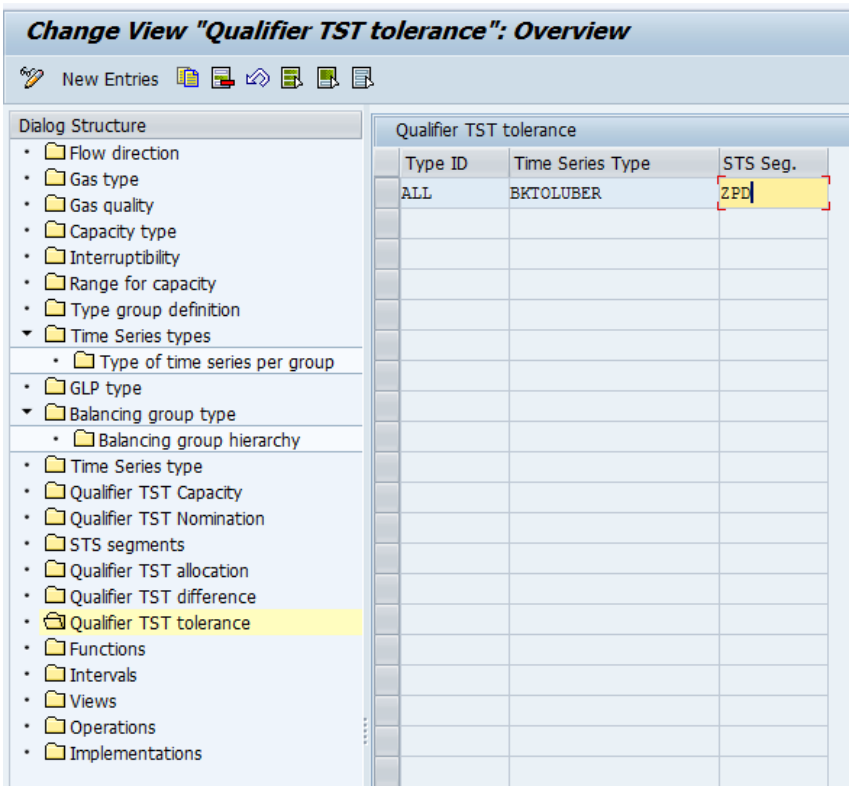


Figure 147: Customizing Example: Qualifier TST Tolerance

Field	Data element	Description
Type ID	/PLGA/DESL_RTYPE	Type ID
Time Series Type	/PLGA/DESL_ZRT	Time Series Type
STS Seg	/PLGA/DESL_STS	STS Segment

Table 68: Qualifier TST tolerance

2.9.20 Functions

Table: */PLGA/TTESLFU*

Text table*: /PLGA/TTESLFUT*

Description: Functionalities for time series display

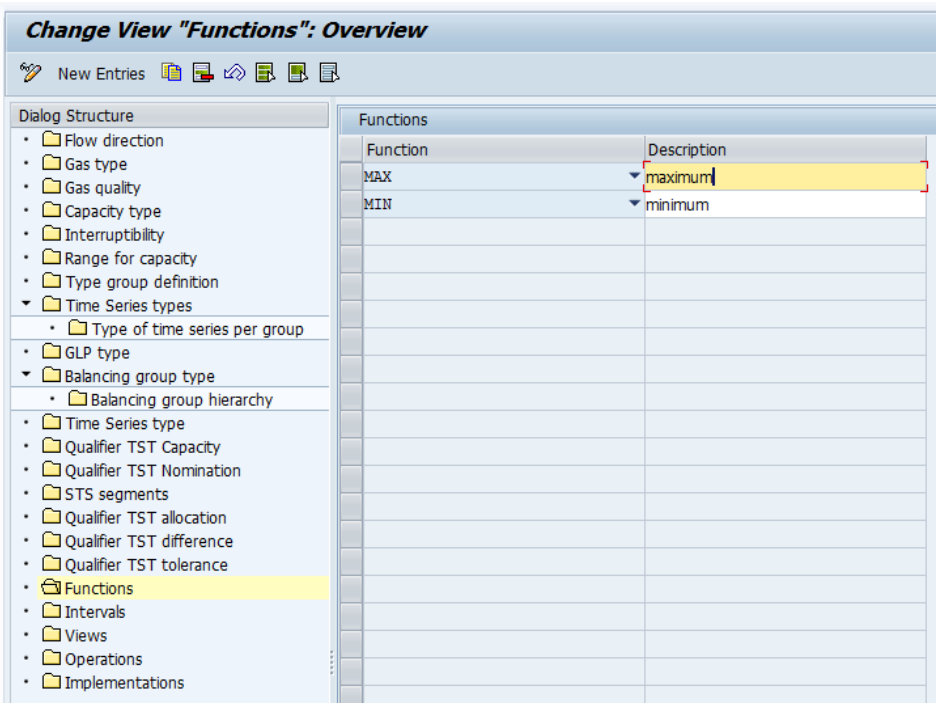


Figure 148: Customizing Example: Functions

Field	Data element	Description
Funct.	/PLGA/DESL_PROF_FUNCTION	Functionality key
Description	/PLGA/DESL_PROF_FUNCDESC	Description of functionality

Table 69: Functions

2.9.21 Interval Length

Table: */PLGA/TTESLIN*

Text table: */PLGA/TTESLINT*

Description: Interval for time series display

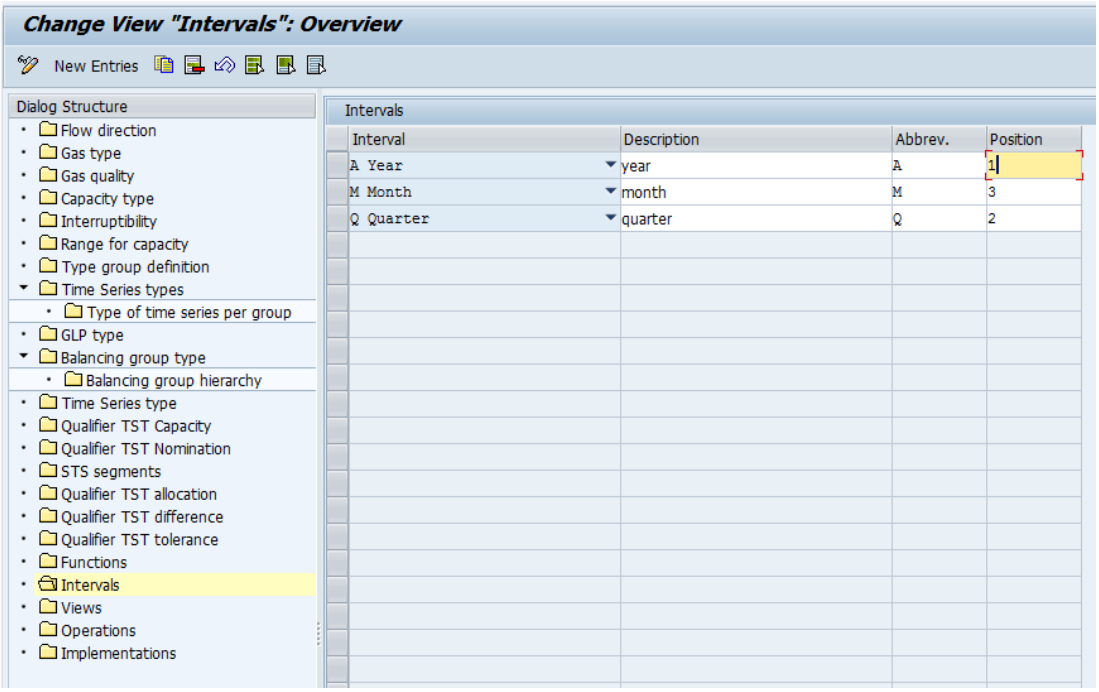


Figure 149: Customizing Example: Interval Length

Field	Data element	Description
Interval	/PLGA/DESL_PROF_INTSIZE	Interval key
Description	/PLGA/DESL_PROF_INTDESC	Description of an interval length
Abbrev.	/PLGA/DESL_PROF_INTSHORT	Abbreviation of an interval length
Position	/PLGA/DESL_POS	Position in the combination field

Table 70: Interval Length

2.9.22 Views

Table: */PLGA/TTESLVIEW*

Text table: */PLGA/TTESLVIEWT*

Description: Definition of views

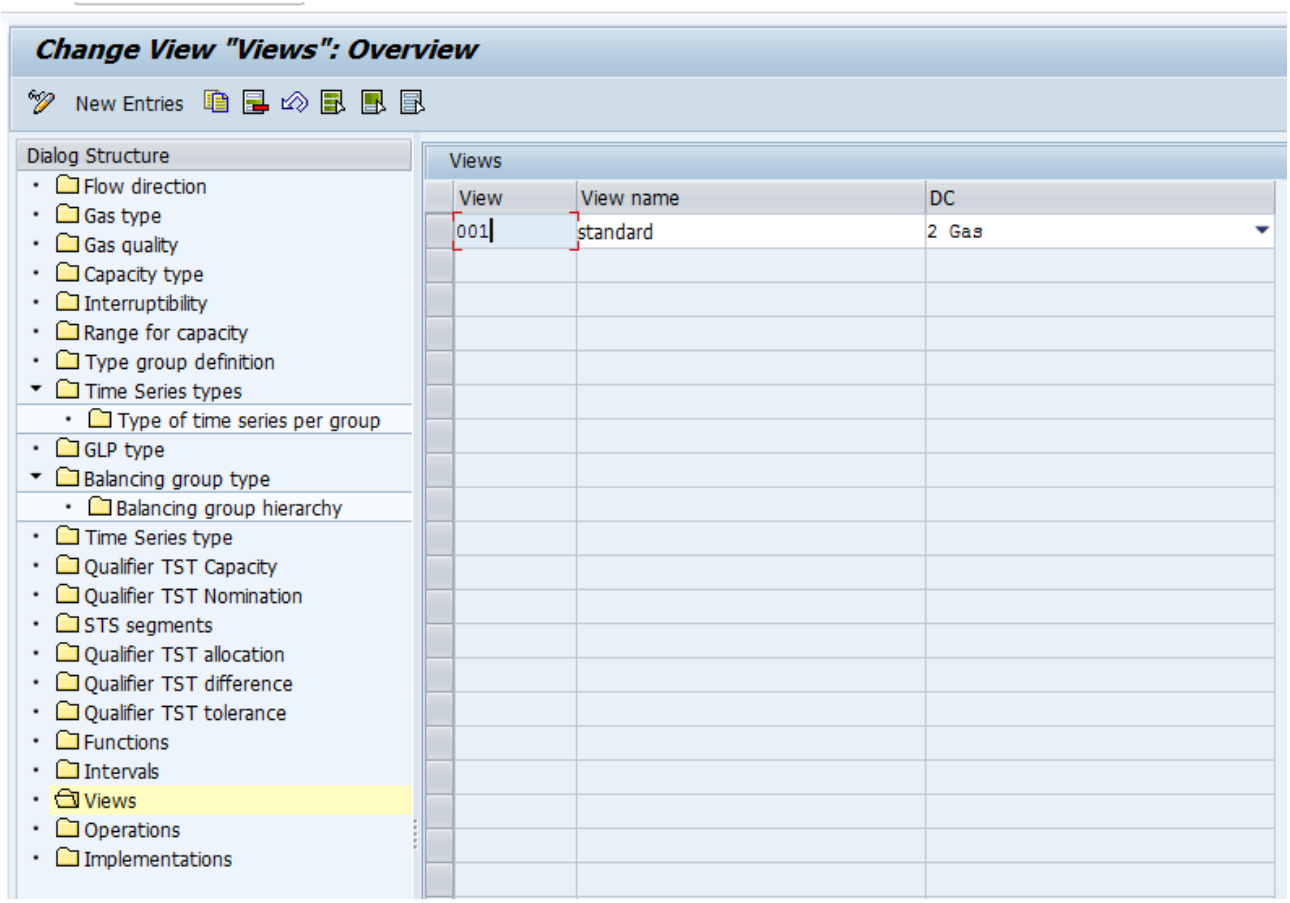


Figure 150: Customizing Example: Views

Field	Data element	Description
View	/PLGA/DESL_VIEW	Interval key of view
View name	/PLGA/DESL_VIEWTEXT	Description of view
Division category	SPARTYP	Division for view

Table 71: Views

2.9.23 Operations

Table: */PLGA/TTESLOPS*

Text table: */PLGA/TTESLOPST*

Description: Definition of operations

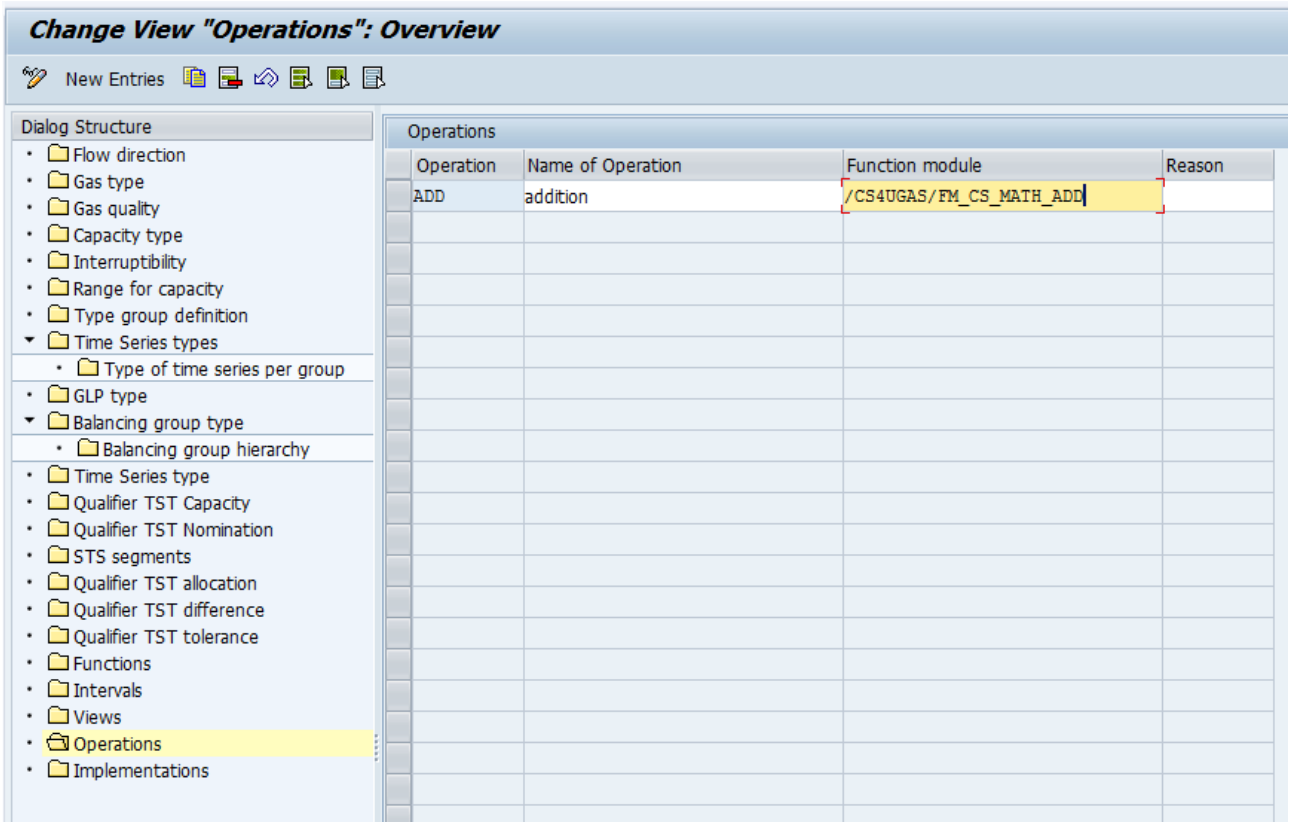


Figure 151: Customizing Example: Operations

Field	Data element	Description
Operation	/PLGA/DESL_OPERATION	Interval key of operation
Name of Operation	/PLGA/DESL_OPTEXT	Description of operation
Function module	/PLGA/DESL_FUNCNAME	Function module of operation
Reason	E_VERSREASON	Reason for creating profile version

Table 72: Operations

2.9.24 Implementations

Table: */PLGA/TTESLCLAS*

Description: Definition of operations

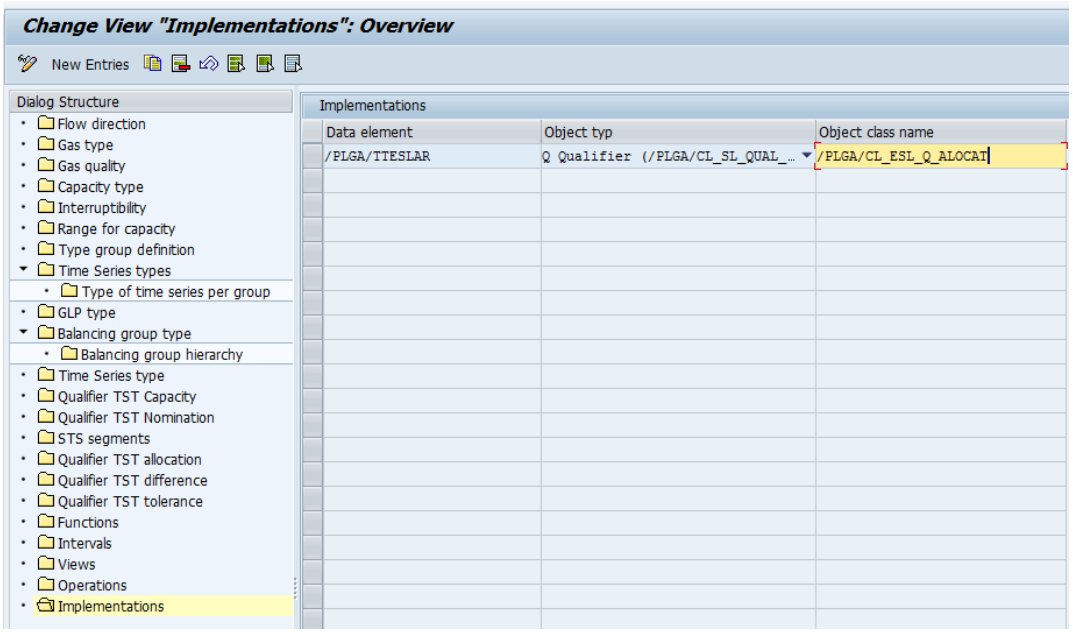


Figure 152: Customizing Example: Implementations

Field	Data element	Description
Data element	ROLLNAME	Data element (semantic domain)
Object type	/PLGA/DESL_CLASS_TYPE	Object type
Object class name	/PLGA/DESL_CLASS_NAME	Name of object class

Table 73: Implementations

2.10 Master data extensions

No configuration is needed for executing the current master data helper reports:

- /PLGA/EDM_CREATE_SLP Create SLP profiles (TUM)
- /PLGA/EDM_MASTER_DATA_CHG Change master data

3 Authorization objects

3.1 Product specific objects

3.1.1 /PLGA/EDMP - Authorization object for Simple Process Flow

The authorization object provides additional authentication for the Simple Process Flow. See details in field description.

Defined fields

The field "ACTION" supports the differencing between creation of process instances and triggering

A: Allows the usage of Report /PLGA/RP_EDMPR_CREATEPI

C: Allows global execution of the SPF, see field /PLGA/EDMP for details

The field "/PLGA/EDMP" refers to the process identification. This field will be check additionally to "ACTION" when the process is logically identified.

The authorization object is mandatory when executing processes with the Simple Process Flow and the dynamic settlement extension.

3.1.2 /PLGA/CFW - Authorization object for Cluster Framework

This authorization object provides additional authentication for the Cluster Framework. See details in field description.

Defined fields

The field /PLGA/ECFA supports differentiating between customizing Checks and Finders, configuring a Cluster and executing a Cluster.

Possible values:

· CST: Allows Finder and Check customization (via transaction /PLGA/ECF_CST)

· CFG: Allows Check Cluster configuration (via transaction /PLGA/ECF_CFG and /PLGA/ECF_CHK)

· RUN: Allows Check Cluster execution (via transaction /PLGA/ECF_RUN)

The field ACTVT supports activities 02 (Change) and 03 (Display) for actions CST and CFG and 16 (Execute) for action RUN.

The authorization object is mandatory when using the Cluster Framework.

3.2 Additional objects

Generally, every transaction within this product is shipped with default values. These values match the authorization checks made within the process.

Additional authorizations may be added from IS-U and EDM standard to allow certain functionalities.

4 Transactions

The following table lists all transaction available within the product.

Transaction code	Description	Associated module
/PLGA/ECA_AUTOCALCWB	EDM automatic calculation workbench	Extension for EDM calculation workbench
/PLGA/ECF_CFG	ECF: Configuration	Cluster Framework
/PLGA/ECF_CHK	ECF: Check settings	Cluster Framework
/PLGA/ECF_CST	ECF: Customizing	Cluster Framework
/PLGA/ECF_LOG	ECF: Diplay logs	Cluster Framework
/PLGA/ECF_RUN	ECF: Run Check Cluster	Cluster Framework
/PLGA/ECV_MAINTBILL	Maintain billing relevant tables	Calorific value extension
/PLGA/ECV_PROF_ST	EDMCV: Maintain profiles to CV area	Calorific value extension
/PLGA/ECV_REF_PV	Calculate reference factor profile	Calorific value extension
/PLGA/EDMSET_CRT_ST	EDM Settlment Ext. maintain control	Dynamic EDM Settlement extension
/PLGA/EDM_CREATE_SLP	Create SLP profiles	Master data extensions
/PLGA/EDM_MD_CHANGE	Change Master Data	Master data extensions
/PLGA/EDM_SLP_TUMX	Configure own TUM SLP types	h-value calculation
/PLGA/EDM_TAPROF	Maintenance Temperature area/profile	Global settings
/PLGA/EU3_CST	EU3: Customizing	Generic CSV/XML Importer
/PLGA/EU3_FILE	EU3 Fileimport	Generic CSV/XML Importer
/PLGA/EU3_MSD	EU3: Master data	Generic CSV/XML Importer
/PLGA/SPF_CREATE	SPF: Create process instance	Simple Process Flow

Table 74: Overview of transaction

5 Webservices

The product offers the following webservices. Some entities are designed to support bundled so-called Business Server Pages Applications (BSP Applications) and are therefore limited to only support the bundled UIs.

The following web services are shipped within the addon:

Technical Service Name	Usage
/PLGA/ESL_SPF_SRV	Configuration/monitoring of "Simple Process Flow" processes/process instances
/PLGA/ESL_BILAMON_HCP_SRV	Display data from "Balancing Group Monitor"
/PLGA/TSO_MASTERDATA_SRV	Determine information for technical connection points (TSO master data)

Table 75: List of webservices

5.1 Adding a service

This chapter shows exemplarily the configuration and propagation of a service for an activated service gateway within the SAP Gui.

Within the transaction */IWFND/MAINT_SERVICE* The button "Add Service" hat to be clicked:

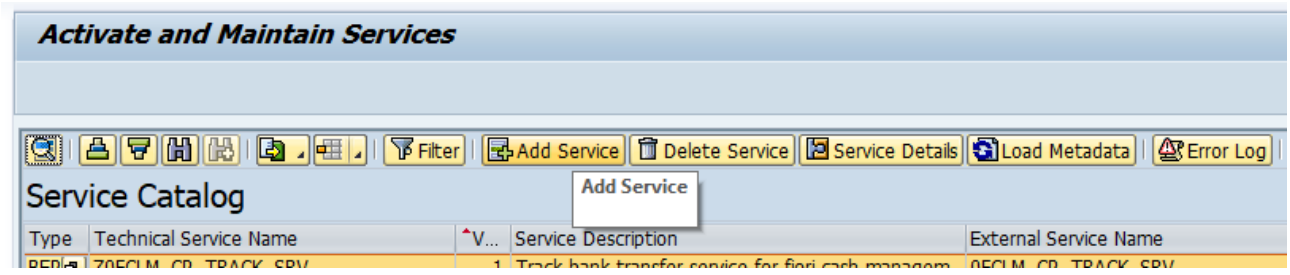


Figure 153: add a service

After selecting the correct "System Alias" and entering the "Technical Service Name", mentioned above, the underlined text can be clicked in the following screen:

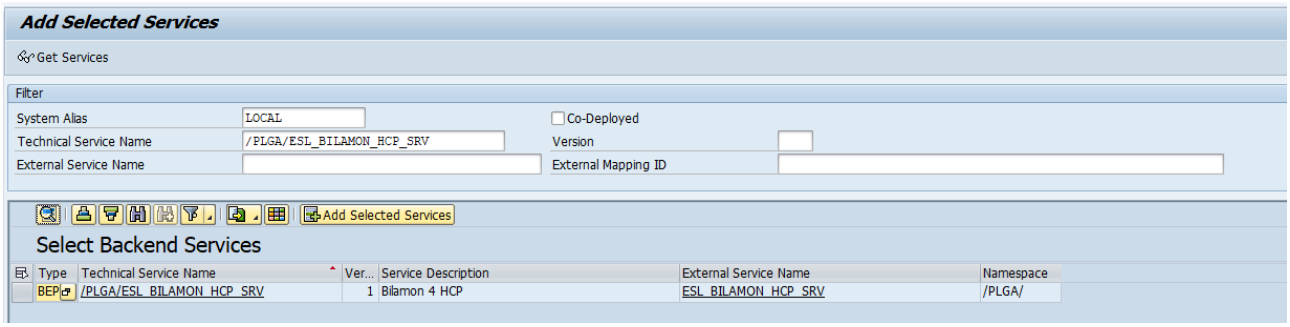


Figure 154: selecting a service to be added

This starts a wizard to save the relevant activation objects. The service can be used afterwards. The following figures show the dependant steps:

This figure shows the general setting of all services mentioned above. You can alter the "Package Assignment" and editable namespaces to your needs.

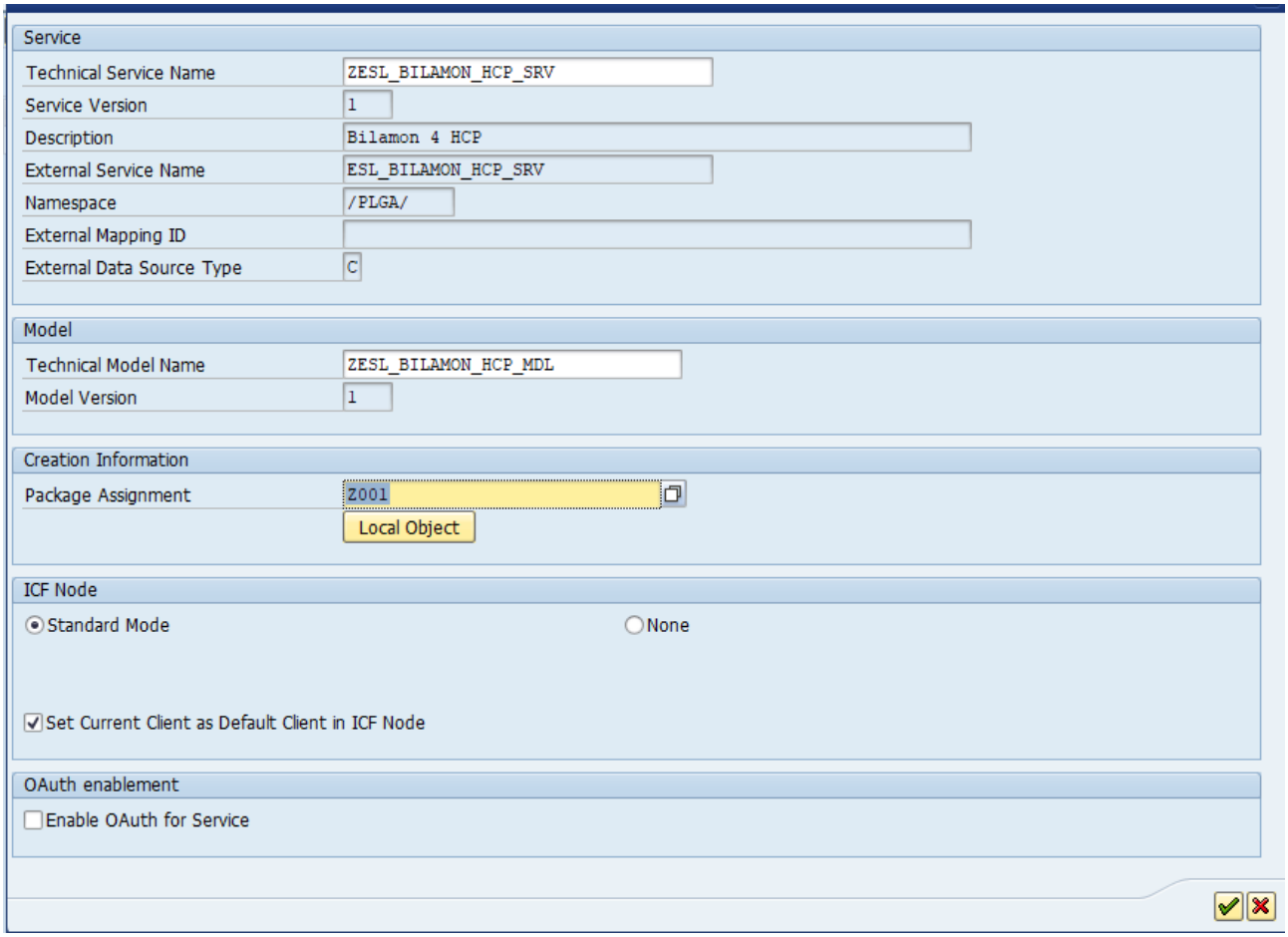


Figure 155: Activating a service, step 1

Selecting a nonlocal package assignment will lead into a transport request selection:

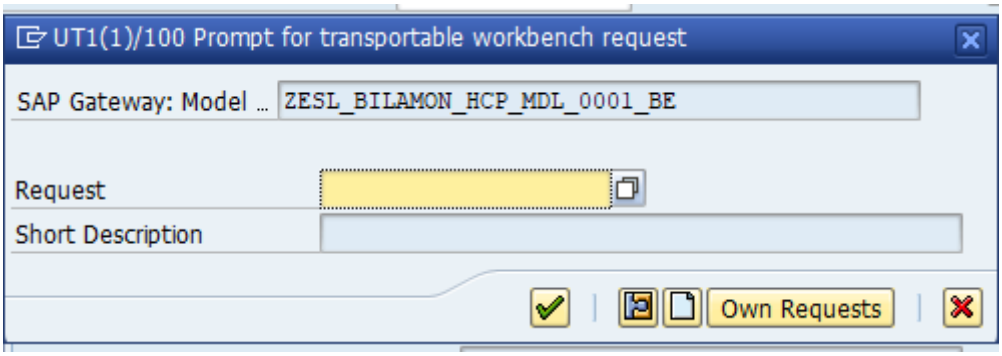


Figure 156: Activating a service, step 2

Afterwards the selected service will be shown in the maintenance monitor with a green ICF node a connected "System Alias", like shown in the following figure:

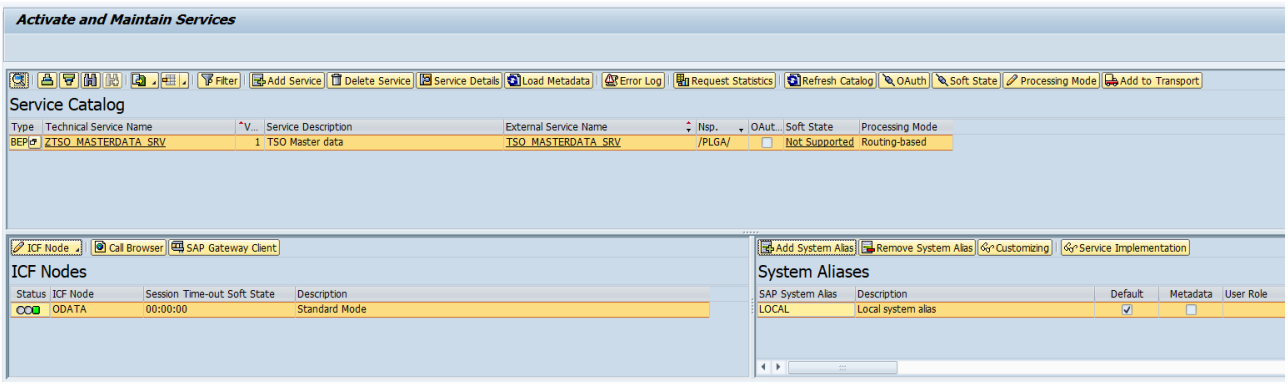


Figure 157: Activating a service, step 3

5.2 Webservice ODATA description

Please refer to the metadata description within the transaction */IWFND/GW_CLIENT* after selecting the choses service.