Business Process Improvement

Customer Exit with ABAP OO in ST-A/PI 01T SP0

October 2018
© 2018 SAP SE or an SAP affiliate company. All rights reserved.

No part of this publication may be reproduced or transmitted in any form or for any purpose without the express permission of SAP SE or an SAP affiliate company.

The information contained herein may be changed without prior notice. Some software products marketed by SAP SE and its distributors contain proprietary software components of other software vendors. National product specifications may vary.

These materials are provided by SAP SE or an SAP affiliate company for informational purposes only, without representation or warranty of any kind, and SAP or its affiliated companies shall not be liable for errors or omissions with respect to the materials. The only warranties for SAP or SAP affiliate company products and services are those that are set forth in the express warranty statements accompanying such products and services, if any. Nothing herein should be construed as constituting an additional warranty.

In particular, SAP SE or its affiliated companies have no obligation to pursue any course of business outlined in this document or any related presentation, or to develop or release any functionality mentioned therein. This document, or any related presentation, and SAP SE’s or its affiliated companies’ strategy and possible future developments, products, and/or platform directions and functionality are all subject to change and may be changed by SAP SE or its affiliated companies at any time for any reason without notice. The information in this document is not a commitment, promise, or legal obligation to deliver any material, code, or functionality. All forward-looking statements are subject to various risks and uncertainties that could cause actual results to differ materially from expectations. Readers are cautioned not to place undue reliance on these forward-looking statements, and they should not be relied upon in making purchasing decisions.

SAP and other SAP products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of SAP SE (or an SAP affiliate company) in Germany and other countries. All other product and service names mentioned are the trademarks of their respective companies. See http://global.sap.com/corporate-en/legal/copyright/index.epx for additional trademark information and notices.

Documentation in the SAP Wiki
You can find this documentation at the following address:
Table of Content

1 INTRODUCTION .................................................................................................................. 5
2 TECHNICAL PREREQUISITES ............................................................................................. 6
3 ROADMAP .......................................................................................................................... 7
4 USE CASE ............................................................................................................................ 8
   4.1 PREPARATION ................................................................................................................. 8
5 DEFINITION OF MONITOR (CUSTOMIZING) ...................................................................... 9
   5.1 CUSTOMIZE PARAMETERS ON MONITOR LEVEL .......................................................... 10
   5.2 DEFINE MONITOR DESCRIPTION TEXT ...................................................................... 12
   5.3 DEFINE KEY FIGURES OF THE MONITOR ................................................................. 13
6 DEFINITION OF KEY FIGURES (CUSTOMIZING) ............................................................... 14
   6.1 DEFINE ATTRIBUTES AND PARAMETERS OF A KEY FIGURE ................................ 15
      6.1.1 Available Parameters ......................................................................................... 21
   6.2 DEFINE DETAIL LIST ............................................................................................... 23
      6.2.1 Define Mapping ................................................................................................. 26
   6.3 DEFINE DESCRIPTION TEXT ON KEY FIGURE LEVEL ............................................ 28
   6.4 DEFINE INSTRUCTION TEXT ON KEY FIGURE LEVEL ............................................ 28
7 CHECKING THE CUSTOMIZING WITH REPORT /SSA/EMR ............................................. 30
8 FUNCTIONS AND MAPPING IN BUSINESS PROCESS ANALYTICS ................................ 33
   8.1 AVAILABLE FUNCTION POOLS FOR BUSINESS PROCESS ANALYTICS ................ 33
   8.2 NECESSITY OF MAPPING FOR BUSINESS PROCESS ANALYTICS ......................... 36
   8.3 MAINTENANCE OF MAPPING FOR BUSINESS PROCESS ANALYTICS ..................... 37
      8.3.1 Postprocessor BPM_PP_ANACUBE .................................................................... 38
      8.3.2 Postprocessor BPM_PP_DETLIST ...................................................................... 40
      8.3.3 Postprocessor BPM_PP_DETLIST_VAL .......................................................... 42
      8.3.4 Postprocessor BPM_PP_ANACUBE_TR ............................................................ 43
         8.3.4.1.1 Example for a throughput key figure without monetary value .................... 43
         8.3.4.1.2 Example for a throughput key figure with monetary value ....................... 44
      8.3.5 Postprocessor BPM_PP_ANACUBE_LT ............................................................ 45
         8.3.5.1.1 Example for a typical lead time key figure .................................................. 45
         8.3.5.1.2 Example for an average count key figure .................................................... 46
   8.4 MAINTENANCE OF MAPPING FOR BUSINESS PROCESS MONITORING .............. 47
      8.4.1 Postprocessor BPM_PP_ALERTING .................................................................. 47
   8.5 MAINTENANCE OF NEW SEMANTIC IDS IN THE SEMANTIC REPOSITORY ................ 48
      8.5.1 Fixed values are maintained for DDIC data element ........................................... 49
      8.5.2 Text table is connected to DDIC data element .................................................... 50
      8.5.3 Text table is connected to DDIC data element but a filter is needed .................... 50
      8.5.4 Table with description is available but not connected to DDIC data element ......... 51
      8.5.5 Sophisticated logic is necessary to get descriptions ............................................ 51
9 DEVELOPMENT OF CUSTOMER EXIT REPORT ........................................................... 53

October 2018 3
<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>9.1</td>
<td>Example Coding for Z_BPM_ECU_COLLECTOR</td>
<td>53</td>
</tr>
<tr>
<td>9.2</td>
<td>Creating GUI Status</td>
<td>62</td>
</tr>
<tr>
<td>9.3</td>
<td>Testing during Development</td>
<td>63</td>
</tr>
<tr>
<td>9.4</td>
<td>Changes in Example Report in Detail</td>
<td>64</td>
</tr>
<tr>
<td>9.4.1</td>
<td>DATA declarations</td>
<td>64</td>
</tr>
<tr>
<td>9.4.2</td>
<td>Class Definition for Monitor</td>
<td>67</td>
</tr>
<tr>
<td>9.4.3</td>
<td>Class Implementation for Monitor</td>
<td>67</td>
</tr>
<tr>
<td>9.4.4</td>
<td>Class Definition for Key figure</td>
<td>68</td>
</tr>
<tr>
<td>9.4.5</td>
<td>Class Implementation for Key figure</td>
<td>68</td>
</tr>
<tr>
<td>9.4.6</td>
<td>Form DC_CUABCDxx (Data Collector)</td>
<td>71</td>
</tr>
<tr>
<td>9.4.7</td>
<td>Form DI_CUABCDxx (Detail information)</td>
<td>72</td>
</tr>
<tr>
<td>9.4.8</td>
<td>Form xxD_USER_COMMAND (Interaction on Detail list)</td>
<td>73</td>
</tr>
<tr>
<td>9.4.9</td>
<td>Form VH_CUABCDxx (Value help for parameters)</td>
<td>75</td>
</tr>
<tr>
<td>10</td>
<td>Further Information</td>
<td>79</td>
</tr>
<tr>
<td>10.1</td>
<td>Setup Business Process Analytics</td>
<td>79</td>
</tr>
<tr>
<td>10.2</td>
<td>Development Procedure in a 3-System Landscape</td>
<td>79</td>
</tr>
<tr>
<td>10.3</td>
<td>Related Documents</td>
<td>80</td>
</tr>
</tbody>
</table>
1 Introduction

SAP provides the functionality of Business Process Analytics in SAP Solution Manager in order to identify and analyze process inefficiencies. A huge number of monitors are already available (see the KPI Catalog).

Since sometimes there are requirements that cannot be fulfilled by the usage of "standard" monitors provided with the ST-A/PI Plug-In, SAP has introduced a customer exit. With this customer exit you can define your own monitors, develop your own data collectors and use the whole functionalities provided by SAP Business Process Analytics, Business Process Operation Dashboards, and Business Process Monitoring framework.

This guide explains how to implement the customer exit for Business Process Analytics and Business Process Monitoring.

The first step is to define your own monitor. This must be done by using the so called “Customizing report” /SSA/EXM. This report is part of the ST-A/PI software component and must be called in the managed system. With this report you can define the customer specific monitor meta data like its name/description, its key figure definitions and the parameters which will be used for selecting the data.

The framework of the customer exit is the program /SSA/ECU which is shipped with the ST-A/PI software component, from where the report Z_BPM_ECU_COLLECTOR is invoked to perform actual data collection. You must develop the coding for the customer exit inside this report (by following strictly some conventions which will be described in this document).
2 Technical Prerequisites

To setup and use Business Process Analytics, the following releases of software components are necessary as an absolute minimum:

- SAP Solution Manager 7.2 SP03 or higher
  This guide is based on SAP Solution Manager 7.2 SP 8.

- The connected managed system must have at least ST-A/PI 01S and ST-PI 2008_1 SP06 installed.
  This guide is based on ST-A/PI 01T SP0 and it is recommended to implement this version.

The corresponding note for SAP Solution Manager 7.2 SP03 or higher is 2324106.
3 Roadmap

The development of a customer specific monitor with the BP Improvement customer exit framework consists of the following steps (all steps will be explained in detail in this document):

1. Define a Monitor with the customizing report /SSA/EXM in the managed system (development system)
2. Create the report Z_BPM_ECU_COLLECTOR in the managed system
3. Copy the coding template into the report Z_BPM_ECU_COLLECTOR (see chapter 9.1)
4. Adapt the coding template to your needs
5. Transport both the report Z_BPM_ECU_COLLECTOR and the customizing for project ECU which you defined via report /SSA/EXM from the development system to the test system
6. Trigger an update of the analytics objects repository into the SAP Solution Manager system to make the new monitor available (reload of monitors)
7. Test the new monitor
8. Transport everything from the test system to the productive system
4 Use Case

A customer is using the standard monitor ‘SD Invoices (AR)’ with the key figure ‘06: Billing due sales documents’. The available selection criteria provided with the standard collector are not sufficient for the customer.

As an additional field the ‘Destination country’ is required to be shown in the result list and it is also required to be enabled as a grouping category for Business Process Analytics. So does the whole functionality of Business Process Analytics including Advanced Benchmarking and Value Benchmarking.

In the following example, it will be explained how to develop a customer exit which contains the standard coding for the monitor ‘SD Invoices (AR)’ and how to extend this monitor to fulfill the customer requirement.

- The used standard Monitor is ‘SD Invoices (AR)’.
  Technical name is SD0002, report /SSA/EKP
- Additional field for selection is Destination country.
  Technical name is LLAND from table VKDFS

4.1 Preparation

For the given example it would be necessary to identify the standard coding which is collecting the data for the key figure 06 of the monitor ‘SD Invoices (AR)’.

The coding part of this monitoring object can be found by debugging the collector for the key figure 06 locally in the managed system with transaction ST13.

To save your time to find the coding for this specific monitoring object we attached the complete coding in chapter 9.1.
5 Definition of Monitor (Customizing)

The definition of the monitor and the customizing is done with the report /SSA/EXM. This report is delivered with the software component ST-API and it must be executed in the managed system on which the monitoring should be implemented. Please mind that the report is not only used for Business Process Analytics, therefore not all functions within the report are necessary and won’t be explained in this guide. Execute the report /SSA/EXM via transaction SE38. There are two parameters in the start screen of this report with prefilled default values ‘ECU’ and ‘Display Mode’. Choose ‘Change Mode’ instead as shown in the following screenshot.

![Figure 1](image1.png)

**Figure 1** Entry screen Customizing Report with default parameters

After pressing the “Execute” button you will see the main screen. You must maintain all settings for defining a Customer Monitor. The first time you start the report you will see one monitor named CUST00 without a description text. This entry is delivered with the report but does not contain any actual functionality. It only shows as an example how the naming of the monitor can be done and it is recommended to delete this entry before starting the actual development.

There is no naming convention for the monitoring object. This guide follows a name convention like this: ABCD plus a 2-digit number. Alternatively, you could use an abbreviation of your company, or e.g. CUST plus a 2-digit number as the above-mentioned name. This technical name of the monitor (which is later evaluated in the customer-exit framework) must be entered in the first column. You should put the description of the monitor in the second column. This is the name of the monitor which will be shown later in the setup session for Business Process Analytics in the SAP Solution Manager.

In case a customer exit is developed on different systems respectively products which are all connected to the same SAP Solution Manager it is very important not to use the same technical names on more than one system, e.g. you should not have one monitor named ABCD01 on an ERP and another monitor with the same name on a CRM system. Only in case the monitors and all key figures are identical, the same names might be used. Otherwise errors will appear when using Business Process Analytics as specific attributes of a monitor are used for different monitors (see also chapter 9.3).

By default, the name of the ABAP report is set to Z_BPM_ECU_COLLECTOR. The name of the report can be changed to any customer report name fitting to your naming convention. In this example, we will just use the proposed name Z_BPM_ECU_COLLECTOR.

![Figure 2](image2.png)

**Figure 2** Define Monitor “Sales Documents (Customer Exit)”
As shown in Figure 2 the technical name of the monitor in our example is ABCD01, the description is ‘SD Invoices (AR) (Customer Exit)’. The suffix ‘Customer Exit’ is added to make it easier to separate standard monitors from customer developed monitors during setup of Business Process Analytics in the SAP Solution Manager. Save the monitor (CTRL+S) after you defined the name and description for this monitoring object.

5.1 Customize Parameters on Monitor Level

A click on the function button ‘Customize Parameter’ (shown in Figure 2, marked with the number 1) will bring you to the screen as shown in Figure 3. The boxes which are marked with a red cross must be activated.
In our example, we need the capability to assign the monitor to a business process step (for Business Process Monitoring) and to a logical component group (for Business Process Analytics). And we activate all three available flags for the BW Enablement. Furthermore, it is mandatory to activate the flag for the field ‘TBI-OO Framework’ as the customer exit report is developed in ABAP OO. After activating this flag and pressing ENTER a new input field will be available named ‘Monitor Class’ where the class name for the monitor must be named. There is no predefined naming convention, that has to be followed, but the name entered here has to be used later in the coding of the customer exit.

![Figure 4](image)

Click on the tab ‘Application Areas’ and you will see the following maintenance screen:

![Figure 5](image)

Under the tab ‘Application Areas’ there is a table where you have to select an application area (using the button “Add”) to which this monitor will be assigned to. The chosen application area is used later in the setup as a filter on the local catalog, and in ST13 you also find the monitor below that application area (compare Figure 55).
Either choose an application area starting with BPO_ or the one specifically created for customer specific monitors.

Only one application area should be selected, otherwise in Business Process Analytics the related key figures will be shown multiple times. Monitors developed in the customer exit can be assigned to the application area ‘Customer Defined’, but in case your authorization concept is based on different application areas (e.g. employees are only authorized to see key figures for ‘Procurement’) it is recommended to assign the monitors to the standard areas starting with ‘BPO_’.

5.2 Define Monitor Description Text

Via button “Monitor Description” (Figure 2, marked with number 2) you may enter a monitor description. The description text should describe the monitor very roughly in just one sentence.

In Solution Manager, when creating a new key figure data collection from the local catalog in the Solution Documentation the text will be displayed when clicking on a monitor:
5.3 Define Key Figures of the Monitor

To define the key figure for a monitor you need to click on the function button ‘Define Key figures’ (shown in Figure 2, number 3). The meaning of the key figures and their customizing is explained in the following chapter.
6 Definition of Key Figures (Customizing)

When you click the button ‘Define Key figures’ in the monitor overview you will see the following screen:

![Define Key Figures for Monitor ABCD01 (SD Invoices (AR)) (Customer Exit)](Figure 9 Define Key Figure of Monitor)

If it is your first entering this screen, you will see an empty list not containing any key figure.

Enter the settings as shown in Figure 9 and save.

In the column ‘Type’ you choose the type of the key figure.

![Available types for key figures](Figure 10 Available types for key figures)

Types ‘L1’, ‘L0’, ‘L1’, ‘L2’, ‘L3’, and ‘L4’ allow the usage of more than one counter for this key figure when using the key figure for Business Process Monitoring, whereas types ‘KZ’, ‘K1’, ‘K2’, ‘K3’, and ‘K4’ are restricted to only one counter. Most of the SAP standard monitors delivered with ST-A/PI are based on type ‘LI’. Compared with ‘L1’, the types ‘L1’, ‘L2’, ‘L3’, and ‘L4’ have the advantage that it is not necessary to define the threshold details in the attributes (e.g. text “Threshold for YELLOW”). For the example described in this document we use the type ‘L1’. Alternatively, ‘L2’ could be used.

The flag ‘DCC relevant?’ is related to data consistency check tools and has no impact for the functionality of a monitor in Business Process Analytics.

Finally enter the description of the key figure, in this example it is ‘Billing due sales documents’ and save the settings.

One monitor may consist of one or more key figures. The data collection and alerting are done (in most cases) on key figure level, so does the example in this document. But in case of data collectors developed in object oriented TBI framework, in principle it is also possible to do the data collection on monitoring object level. This makes sense when the data collection for the key figures assigned to a monitoring object is always done in the same manner and only the result list is evaluated differently for the different key figures. Then the data collection can be done on monitoring object level which means the coding for data collection is done in the method ‘Collect Data’ of CLASS ABCD01_MONOBJ. In this case the customizing of the selection criteria is also done on the monitoring object level (and not on key figure level as it will be shown here in this document).
6.1 Define Attributes and Parameters of a Key Figure

To define the attributes and parameters on key figure level click the function button ‘Customize Parameters’ (see Figure 9, number 1). Here you define some attributes of this key figure and whose selection parameters will be used later during the setup of the Business Process Analytics data collection setup session.

![Define attributes of Key Figure](image)

**Figure 11** Define attributes of Key Figure

On the tab ‘Attributes’, the setting for ‘Keyfigure Category (Cloud Catalog)’ is not relevant for customer exit key figures as these can be only chosen from the local catalog during setup in the Solution Manager. Therefore, you first must define the technical category of the key figure. For typical backlog key figures choose category ‘BL’ (Backlog), for throughput key figures select ‘TH’ (Throughput).

The key figure category of a Percentage key figure, a Lead time key figure, respectively an Average Count key figure is ‘AC’ (Average Count). In the case of an Average Count key figure the measured value in Business Process Analytics is calculated by dividing a certain value (stored in 0SMD_SUM) by e.g. the number of rows of the detail list (stored in 0SMD_COUN). The following table gives an overview about which fields of the BW cube must be filled by the mapping depending on the key figure categories.
Customer Exit in Business Process Analytics

<table>
<thead>
<tr>
<th>Key figure category</th>
<th>Relevant field in the BW cube</th>
</tr>
</thead>
<tbody>
<tr>
<td>TH (Throughput)</td>
<td>0SMD_SUM</td>
</tr>
<tr>
<td>BL (Backlog)</td>
<td>0SMD_SUM</td>
</tr>
<tr>
<td>AC (Average Count)</td>
<td>0SMD_SUM and 0SMD_COUN</td>
</tr>
<tr>
<td>AP (Average Period)</td>
<td>0SMD_SUM and 0SMD_PERI</td>
</tr>
<tr>
<td>MX (Maximum Measured value)</td>
<td>0SMD_MAX</td>
</tr>
</tbody>
</table>

Table 1 Key figure categories and mandatory fields for the BW cube

The Function Pool summarizes a combination of functions in Business Process Analytics. Those functions are for example Benchmarking, Advanced Benchmarking (incl. value based analysis), Trend Analysis (incl. value based trend analysis), Detail List, or Age Analysis. Depending on the selected Function Pool, different functions will be available later on in BP Analytics.

Together with the Function Pool the Postprocessing Indicator determines how the selected data from the detail list will be transformed to the BW Cube.

<table>
<thead>
<tr>
<th>Postprocessing Indicator</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TBI_LT</td>
<td>Lead Time</td>
</tr>
<tr>
<td>TBI_STD</td>
<td>Standard</td>
</tr>
<tr>
<td>TBI_TR</td>
<td>Time Range</td>
</tr>
</tbody>
</table>

Table 2 Postprocessing Indicators

For all key figures of category ‘BL’ the Postprocessing Indicator is typically ‘TBI_STD’, for key figures of category ‘AC’ typically it must be “TBI_LT”.

Postprocessing Indicator “TBI_LT” ensures that no document is written more than once to the BW cube. This is important as e.g. Lead Time key figures collect data for a certain reference period. When using a reference period, it is probable that one document is collected in more than one collector runs. Therefore, it is important to ensure that this document is only written once to the BP Analytics InfoCube.

In case of key figures of category ‘TH’ it must be distinguished if the data collector runs only for one day or if a time range is analyzed. If time ranges are allowed, the Postprocessing Indicator must be ‘TBI_TR’, otherwise ‘TBI_STD’ is correct. Postprocessing Indicator “TBI_TR” ensures that no document is written more than once to the BW cube. When using a time range, it is probable that one document is collected in more than one collector runs.

In the area ‘Additional functionalities’ you can steer if additional function buttons in the Setup session in the SAP Solution Manager are shown or not. Besides that, the functionality behind these buttons must be programmed in the customer exit.

- Validation check: when active, an additional function button will be shown in SAP Solution Manager during setup of the key figure. The button will be named ‘Validation check’. It can be used to process a validation of the entered data for the parameters.
- Default Value: a function button with the name ‘Default Value’ will be shown during setup of the key figure. It can be used to fill some of the selection parameter with default values.
- Detail Info: A button with the name ‘Detail Info’ will be shown in Business Process Analytics and in the Business Process Monitoring session on key figure level. It can be used to show a detail list of objects when marking an alert.
When you want to use the key figure for Business Process Monitoring it is very important to enter a name for the Threshold 1 and 2 as you see it in the Figure 12 (the naming here is ‘Threshold for YELLOW’ and ‘Threshold for RED’). If you miss this here there will be no possibility to enter the alert threshold values during Business Process Monitoring setup. Alternatively you can choose a key figure type like ‘L2’ to which the thresholds are directly connected.

In case of backlog key figures, you need two threshold values (threshold 1 for the alert level from green to yellow, threshold 2 for the alert from yellow to red).

![Figure 12 - Naming of Threshold in Case of a 2-Step Rating](image)

In case of a 2-step rating you enter only the naming for the thresholds 1 and 2; in case of a 4-step rating (usually the case for throughput key figures) you also enter the threshold names for threshold 3 and 4. Please see the following Figure 13 where the naming of the 4 thresholds is shown.

![Figure 13 - Naming of Threshold in Case of a 4-Step Rating](image)

The default unit of a key figure is ‘Document’. Therefore, in case the result of a key figure is based on documents nothing needs to be entered in field ‘Unit’. In other cases use the value help to choose other units as ‘Items’, ‘Materials’, etc. This unit will be shown in Business Process Analytics and in ST13.

![Figure 14 - In case the key figure counts documents, no unit needs to be entered](image)

It is recommended to enter the name of the key figure class directly in the customizing. If this is done, it is not necessary to redefine method GET_KF_CLASSNAME in the class of the monitoring object (compare source code of report Z_BPM_ECU_COLLECTOR).

![Figure 15 - Implementation details of the key figure](image)

Under the tab ‘Parameters the selection parameters are defined for the customer exit (see next Figure 16).
In Figure 16 the complete list of selection parameters is shown as it is needed for this example. The parameters contain the selection parameter as they are used for the standard collector and in addition the new parameter ‘Destination country’ is added.

Remark: As this use-case is to extend an existing key figure with a new parameter for selection it is also possible to copy the settings from the existing key figure and only to extend the copy with the additional parameter.

To copy the setups from an existing monitor, do the following steps:

1. Start the report /SSA/EXM
2. Enter EKP into the field P_PROJID (replace ECU)
   Choose ‘Display Mode’ (to avoid un-intended changes on the standard customizing)
   EKP is the name of the project where many monitoring objects for an ERP system are grouped together.
3. The technical name of the monitor is SD0002. Place the cursor in it and use the button ‘Define Key figures’.
4. Place the cursor into key figure 06 (Billing due sales documents) and use the button ‘Customize Parameters’
5. Click on the tab ‘Parameters’ and the list of parameters for this key figure is shown and can be copied into the customer exit key figure. To copy all columns in one step, narrow some columns.
6. It is recommended to copy the fields Type, Description, Tech Name 1, Tech Name 2, BI Para, Semantic ID.
   The field Parameter ID has no meaning regarding the code of the customer exit key figures in report Z_BPM_ECU_COLLECTOR (so you may enter e.g. enter numbers starting from 1), but to complete and save the customizing it must be filled uniquely for one key figure. For reasons of simplicity it is recommended to copy the content from column ‘Tech Name 2’ to column ‘Parameter ID’.

   The flags cannot be copied, they must be activated manually.
7. Jump back into the overview list of all key figures and jump to the definition of the detail list of key figure 6 by using the button ‘Detail List.’
Now you can also copy the definition of the detail list from here into the detail list definition for the customer exit key figure (see chapter 6.2).

The meanings of the columns of the table shown in Figure 16 are:

- Column ‘No.’: this column is filled with an ongoing numbering automatically by the system after saving.
- The field ‘Parameter ID’ has no meaning regarding the code of the customer exit key figures, but to save the customizing it must be filled uniquely for one key figure. For reasons of simplicity it is recommended to copy the content from column ‘Tech Name 2’ to column ‘Parameter ID’, as for all select options which are defined as a group-by dimension, the Parameter ID needs to be entered also in the definition of the detail list (see Figure 22, column ‘Parameter ID’).
- Column ‘Type’: data type of the parameter (can be chosen by an F4-Help)
- Column ‘Description’: free text for describing the parameter. This is the name for the parameter as it is shown later in the Business Process Analytics data collection setup session.
- Columns ‘Tech Name 1’, ‘Tech Name 2’: these columns are especially important if you want to provide an F4-Help for a parameter which is based on the existing F4-Helps in the data dictionary. Then you must enter in ‘Tech Name 1’ the name of the table and in ‘Tech Name 2’ the name of the table field from where you want to use the F4-Help. In case a special value help is necessary ‘Tech Name 1’ must be initial (compare chapter 9.4.9).
  The name of “Tech Name 2” must be identical to the parameter used in the code (compare chapter 9.4.4). Furthermore “Tech Name 2” must be used uniquely for one key figure.
- Column ‘Default Values’: can be used to maintain single default values or ranges which appear as default value in ST13 resp. in the setup of Business Process Analytics. To enter a default value, click on button ‘Maintain’.

![Change Default Values for Parameter VBTYP (SD Document Cat.)](image)

**Figure 17**  Example for the maintenance of a default value

As soon as a default value is maintained for a parameter, the displayed name of the button is ‘Change’ instead of ‘Maintain’.

![Maintained default values can be recognized by a changed button name](image)

**Figure 18**  Maintained default values can be recognized by a changed button name

- Column ‘Mandatory’: can be used when you want to make a parameter mandatory during the Business Process Analytics data collection setup session. When the flag is set for parameters offering a fix value F4 help, in the setup session initially the first fix value is displayed.
- Column ‘Check against F4-Help’: must be activated if you want to provide a F4-Help, either a dynamic one or with fix values.
- Column ‘No dyn. F4-Help’. must be activated if you want to provide a value help with fix values (this makes sense e.g. when there is no data dictionary F4 Help assigned to the field, or in case of flags). To provide a self-defined value help table you have to press the button ‘Fix F4-Help Values’ and maintain your value table there (in the above shown example this should be done for the parameter ‘Exclude internal customer’). You will see a table where you can enter the value and a description for
this value you want to provide with your F4-Help. It is important that you activate the flag in column ‘No dyn. F4-Help’ and the flag ‘Check against F4-Help for this parameter.
By setting these two flags in combination with only the two fix values ‘ ’ and ‘X’, for this parameter a checkbox is shown in ST13.

![Figure 19 Values for fixed F4 help](image)

- Column ‘Selection Option’: If this flag is active, then during Business Process Analytics setup the parameter is shown as a range parameter (otherwise there would only the possibility to maintain a single value as the selection option is displayed on the selection screen without a ‘to’ field).

Generally, all parameter fields used as select options should be shown in the detail list. For those selection fields which are used for BW aggregation it is mandatory that they are defined there.

Scroll the table to the right to see some more customizing fields as shown in the following Figure 20.

![Figure 20 Customizing of Key Figure (Parameters): part 2](image)

The meanings of the remaining columns of the table shown in Figure 20 are:

- Column ‘Group By Only’: If the flag is set, a parameter is not available in the setup session, but it is possible to activate it as a ‘Group By’ parameter for Benchmarking in Business Process Analytics.
- Column ‘No Extension’: If the flag is set, you can only make an entry on one line. Calling the additional "Multiple Selection" screen is not supported and no pushbutton for this appears on the selection screen. This makes sense e.g. when a date range is entered.

- Column ‘Case Sensitive’: If the flag is set, the entered value is not converted to upper case and remains as entered. This is only relevant in a few cases as e.g. in RFC monitoring for the RFC destination. Usually the flag is not set, and it is not considered when running the key figure in ST13.

- Column ‘BI Para’: To use the key figure in Business Process Analytics you must define at least one parameter to be used for aggregation in the BW of the SAP Solution Manager. You do so by entering a number starting from 1 into the column ‘BI Para’. If you want to enable a second parameter, you have to enter the number 2 and so on. Those parameters which are used for aggregation are called “group-by fields”.

If not at least one parameter is maintained as a group-by field, the key figure cannot be setup for Business Process Analytics.

Don’t change the ‘BI Para’ numbers for a key figure which is already used productively because then inconsistencies will occur in the BW cube!

- Column ‘Semantic ID’: For all fields which are used as a group-by field for Business Process Analytics it is mandatory that the Semantic ID is entered. The assignment of a Semantic ID allows uploading the language dependent text description of a selection criterion into BW.

The F4 help offers Semantic IDs for the most important fields used in ERP. According to the naming convention, the name of the Semantic ID contains the name of the data element to which it refers. If you want to enable selection parameters for BW choose the Semantic ID from the list which fits to the data element you use for the definition of your selection criteria.

6.1.1 Available Parameters

Usually not only select options are defined as parameters but also parameters as “Older than x day”, “No status details”, or “Use creation date”.

All types which can be used to specify those parameters are defined in structure TS_CHDR_STD (Include /SSA/IEF within report Z_BPM_ECU_COLLECTOR).

To limit the number of fields in this structure on the one hand, but to be as flexible as possible on the other hand there are also generic fields as NUMBER_1 which is an integer or CHECKBOX_1 as a character. The following table contains all fields of structure TS_CHDR_STD with the typical description text.

Of course, every field of the structure can be used with a different description as originally used for a standard key figure.
<table>
<thead>
<tr>
<th>Field name</th>
<th>Type</th>
<th>Typical usage / Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CHECKBOX_1</td>
<td>C</td>
<td>For generic usage</td>
</tr>
<tr>
<td>CHECKBOX_2</td>
<td>C</td>
<td>For generic usage</td>
</tr>
<tr>
<td>CHECKBOX_3</td>
<td>C</td>
<td>For generic usage</td>
</tr>
<tr>
<td>CHECKBOX_4</td>
<td>C</td>
<td>For generic usage</td>
</tr>
<tr>
<td>CHECKBOX_5</td>
<td>C</td>
<td>For generic usage</td>
</tr>
<tr>
<td>CHECKBOX_6</td>
<td>C</td>
<td>For generic usage</td>
</tr>
<tr>
<td>CHECKBOX_7</td>
<td>C</td>
<td>For generic usage</td>
</tr>
<tr>
<td>CHECKBOX_8</td>
<td>C</td>
<td>For generic usage</td>
</tr>
<tr>
<td>CHARACTER_4</td>
<td>C(4)</td>
<td>For generic usage</td>
</tr>
<tr>
<td>NUMBER_1</td>
<td>I</td>
<td>For generic usage</td>
</tr>
<tr>
<td>NUMBER_2</td>
<td>I</td>
<td>For generic usage</td>
</tr>
<tr>
<td>NUMBER_3</td>
<td>I</td>
<td>For generic usage</td>
</tr>
<tr>
<td>NUMBER_4</td>
<td>I</td>
<td>For generic usage</td>
</tr>
<tr>
<td>OTXDAY</td>
<td>I</td>
<td>Older than x days</td>
</tr>
<tr>
<td>YTXDAY</td>
<td>I</td>
<td>Younger than y days</td>
</tr>
<tr>
<td>OTXHOUR</td>
<td>I</td>
<td>Older than x hours</td>
</tr>
<tr>
<td>YTXHOUR</td>
<td>I</td>
<td>Younger than y hours</td>
</tr>
<tr>
<td>OTXMIN</td>
<td>I</td>
<td>Older than x minutes</td>
</tr>
<tr>
<td>YTXMIN</td>
<td>I</td>
<td>Younger than y minutes</td>
</tr>
<tr>
<td>ODSNC</td>
<td>I</td>
<td>Overdue since</td>
</tr>
<tr>
<td>ODLESS</td>
<td>I</td>
<td>Overdue less</td>
</tr>
<tr>
<td>MTXPER</td>
<td>I</td>
<td>More than x periods</td>
</tr>
<tr>
<td>LTYPEPER</td>
<td>I</td>
<td>Less than y periods</td>
</tr>
<tr>
<td>REFPER</td>
<td>I</td>
<td>Ref. period (x days)</td>
</tr>
<tr>
<td>PREV_DAY</td>
<td>C(1)</td>
<td>Data from prev. day</td>
</tr>
<tr>
<td>USE_ERDAT</td>
<td>C(1)</td>
<td>Use creation date</td>
</tr>
<tr>
<td>USE_CHANGE_DATE</td>
<td>C(1)</td>
<td>Use Change Date</td>
</tr>
<tr>
<td>USE_PLAN_DATES</td>
<td>C(1)</td>
<td>Use Planned Dates</td>
</tr>
<tr>
<td>NO_STATUS</td>
<td>C(1)</td>
<td>No status details</td>
</tr>
<tr>
<td>NO_ITEM</td>
<td>C(1)</td>
<td>No SL Item Details</td>
</tr>
<tr>
<td>USE_JEST</td>
<td>C(1)</td>
<td>Expect high backlog</td>
</tr>
<tr>
<td>GET_VALUE</td>
<td>C(1)</td>
<td>Get Transit Value</td>
</tr>
<tr>
<td>DWXDAY</td>
<td>I</td>
<td>Due within x days</td>
</tr>
<tr>
<td>ESCHK</td>
<td>C</td>
<td>Extended status check</td>
</tr>
<tr>
<td>CURFIYEAR</td>
<td>C</td>
<td>Current Fiscal Year</td>
</tr>
<tr>
<td>XFIYEARS</td>
<td>N(2)</td>
<td>with x years in past</td>
</tr>
<tr>
<td>CHANGE_VERSIONS</td>
<td>C</td>
<td>Consider current Change Versions</td>
</tr>
<tr>
<td>CURREN_APPROVERS</td>
<td>C</td>
<td>Show current Approvers</td>
</tr>
<tr>
<td>SMTP_ADDR</td>
<td>ADR6-SMTP_ADDR</td>
<td>E-Mail recipient</td>
</tr>
<tr>
<td>OFFSET_DAYS</td>
<td>C(8)</td>
<td>Creation Date Offset</td>
</tr>
<tr>
<td>REF_DATE</td>
<td>C(5)</td>
<td>Reference Date</td>
</tr>
<tr>
<td>AGE_DATE_FIELD</td>
<td>C(30)</td>
<td>Age date field</td>
</tr>
<tr>
<td>CHECK_VBUND</td>
<td>C(1)</td>
<td>Exclude internal customers / vendors</td>
</tr>
</tbody>
</table>

**Table 3** Available Parameters (content of structure TS_CHDR_STD (Include /SSA/IEF))
6.2 Define Detail List

Via button ‘Detail List’ (Figure 9, marked with number 2) you must define the display list of the collector. The following Figure 21 is showing the definition of the list as it is needed for this example. For this kind of use case it is possible to copy the list automatically from an existing monitor and extend it as needed.

With the button ‘Copy Detaillist from Keyfig’ it is possible to copy the whole detail list from another key figure of the same monitor.

Maintain the list in your system as it is shown in Figure 21.

The meaning of the columns is as follows:

- **Fieldname, DDIC Table and DDIC field:** here you have to enter the fieldname (same name as used in the detail structure TS_ABCD01_DETAIL_01, explained in chapter 9.4.1), the data dictionary table and the data dictionary field. All fields from the detail structure which should be shown in the result list must be entered; the sequence is arbitrary and can be easily changed by marking a line and the use of the buttons ‘Up’ or ‘Down’.

- **Short Text, Medium text, Long Text:** here you can overwrite the name for a column of the detail list with your own content. Without entering any text in these fields, the text is taken from the data element.

- **Length:** Overwrite the length for a field, which is otherwise taken from the data element.

- **Invisible:** Makes a field invisible in the result list, although it is available via button “Change Layout” (CTRL + F8) of the ALV table.

- **Hotspot and user command:** defines a field in an ALV list as a hotspot. This functionality is used to realize the possibility to jump from a result list into a standard transaction (e.g. from the list of sales orders with a document number as hotspot and jump into transaction VA03). When clicking on the field defined as hotspot, the form routine filled in the field ‘User Command’ is executed. There is no naming convention for this form routine, based on the standard monitors you may name it USER_COMMAND_xx, where xx has to be replaced by the technical name of the monitor. In this example xx is replaced by ABCD01, so the full name of the form routine here is
For one detail list multiple hotspots can be maintained, this means if there are columns with different document numbers (e.g. sales order, delivery, and sales invoice) in the detail list it is possible to jump to these directly. The name of the user command must be the same within one key figure, inside the code it is distinguished which display transaction is called.

Scroll the table to the right to see some more customizing fields for the detail list as shown in the following Figure 22.

**Aging:** With this flag it is defined which field is used later in Business Process Analytics for the age analysis. It should be the same field than the one which is used in the where-clause of the select statement related e.g. to the parameter ‘Older than x days’. In this example it is the field FKDAT.

**Parameter ID:** In this field the names of the identifiers which are used for the grouping functionality in Business Process Analytics (those fields for which a ‘BI. Para’ was entered, see Figure 20) have to be entered. There is an F4-help available which shows the identifiers and descriptions of all parameters assigned to the key figure.

**Is Key:** In case of key figures which select data for a certain time range (Postprocessing Indicator “TBI_LT”) it must be ensured that each document is counted correctly. When the document number is not unique, the other key fields (fiscal year, company code) must be marked as key fields. The document number itself should not be marked, but the Semantic ID “BW_DOCNO” must be entered.

**Semantic ID:** For the usage of Advanced Benchmarking in Business Process Analytics it is important to enter a Semantic ID for each field for which a description is available. The F4 help offers Semantic IDs for the most important fields used in ERP.

The ‘Semantic Description’ shows the meaning of the Semantic ID, maintained in the Semantic Repository.
Some Semantic IDs have always to be entered for certain kind of field to enable specific Analytic Functions:

- **BW_DOCNO**: Document Number of the relevant document in the detail list (has to be used when Postprocessing Indicator “TBI_LT” is used to ensure that no document is written more than once to the BW cube)
- **BW_POSNO**: Item Number of the relevant item in the detail list (has to be used when Postprocessing Indicator “TBI_LT” is used to ensure that no item is written more than once to the BW cube)
- **BC_USER**: Must be entered for all user names in the detail list as an authority check to display or hide the fields is based on the Semantic ID
- **BC_NETVAL**: Must be entered for all fields with monetary values in the detail list as an authority check to display or hide the fields is based on the Semantic ID
- **BC_WAERS**: Must be entered for all currency fields in the detail list as an authority check to display or hide the fields is based on the Semantic ID
- **CALDAY**: Must be entered for all fields with a date in the detail list as certain functions in Business Process Analytics are based on the Semantic ID
- **CALMONTH**: Must be entered for all fields with a calendar month in the detail list as certain functions in Business Process Analytics are based on the Semantic ID
- **CALWEEK**: Must be entered for all fields with a calendar week in the detail list as certain functions in Business Process Analytics are based on the Semantic ID
- **CALYEAR**: Must be entered for all fields with a calendar year in the detail list as certain functions in Business Process Analytics are based on the Semantic ID

- **Data Element**: The data element is used to retrieve master data (description of the fields) from the managed systems into the Solution Manager, as less information is stored instead of all DDIC tables and fields. Accordingly, redundant data is avoided. With the corresponding button ‘Load Data Elements’ (see Figure 22) the data elements are determined for those fields where DDIC details are maintained in case an ‘RFC connection’ is also selected.

With button “Check and Define Mapping” (see Figure 23, number 1) a popup opens to maintain the mapping. The mapping allows that for Business Process Analytics the data is written in the BW cube of the SAP Solution Manager. Depending on the Function Pool different functions are available in BP Analytics and for the different functions, different postprocessors are relevant.
6.2.1 Define Mapping

In detail, the mapping is explained in chapter 8. In this chapter, it is only roughly shown based on the exemplary key figure ‘Billing due sales documents’.

The most important postprocessor for Business Process Analytics is BPM_PP_ANACUBE. Figure 25 shows the mapping which has to be maintained for this postprocessor. To do this, mark the line with the postprocessor (see Figure 24) and choose “Map Fields for Postprocessors”.

<table>
<thead>
<tr>
<th>Source Field</th>
<th>Target Field</th>
<th>Aggregation Function</th>
<th>Function Parameter</th>
<th>Unit Field</th>
</tr>
</thead>
<tbody>
<tr>
<td>FKART</td>
<td>OSM_GP01</td>
<td>GROUP_BY</td>
<td></td>
<td></td>
</tr>
<tr>
<td>VSUPF</td>
<td>OSM_GP02</td>
<td>GROUP_BY</td>
<td></td>
<td></td>
</tr>
<tr>
<td>FKTVP</td>
<td>OSM_GP03</td>
<td>GROUP_BY</td>
<td></td>
<td></td>
</tr>
<tr>
<td>VECRO</td>
<td>OSM_GP04</td>
<td>GROUP_BY</td>
<td></td>
<td></td>
</tr>
<tr>
<td>VCLNW</td>
<td>OSM_GP05</td>
<td>GROUP_BY</td>
<td></td>
<td></td>
</tr>
<tr>
<td>VSTEI</td>
<td>OSM_GP06</td>
<td>GROUP_BY</td>
<td></td>
<td></td>
</tr>
<tr>
<td>LILAND</td>
<td>OSM_GP07</td>
<td>GROUP_BY</td>
<td></td>
<td></td>
</tr>
<tr>
<td>NETWR</td>
<td>OSM_SUMV</td>
<td>SUM</td>
<td></td>
<td>WAERK</td>
</tr>
</tbody>
</table>

Please mind: in case of key figures for which the measured value is based on the number of lines in the detail list, which is the most common case, and when there is no monetary value in the detail list, it is not necessary to maintain the mapping for postprocessor BPM_PP_ANACUBE. In case it is maintained anyhow, it must be ensured that this is done correctly.

There is one line per parameter which is defined as a group-by field (compare Figure 20). The source field is the corresponding name of the field in the detail list. The target field is the generic parameter 0SM_GPxx, in which the placeholder xx has to be replaced by the number of the BI generic parameter of the group-by field (compare Figure 20). The aggregation function for those group-by fields is GROUP_BY. As the measured value of the key figure is determined by the summation of the lines of the detail list, the target field is 0SMD_SUM and the aggregation function is ROWCOUNT.

To enable value-based Trend Analysis in Business Process Analytics it is necessary to add another line with the value as source field and the currency as unit. The target field is 0SM_SUMV and as the value is determined by summation, the aggregation function is SUM.

For all columns a value help is available, and it is recommended to use it: depending on the postprocessor the possible entries are shown.
The mapping for postprocessor BPM_PP_DETAILLIST which is used for Advanced Benchmarking and the Detail Analysis does not have to be maintained for common key figures, for which the measured value is based on the number of lines in the detail list. Consider chapter 8.2, Table 10 and Table 12 for details regarding when it is necessary to maintain the mapping.

As the function pool of the key figure (BPM_ANALYTICS_BL_VAL, see Figure 11) also provides value benchmarking, additionally the mapping for postprocessor BPM_PP_DETAILLIST_Val has to be maintained.

<table>
<thead>
<tr>
<th>Source Field</th>
<th>Target Field</th>
<th>Aggregation Function</th>
<th>Function Parameter</th>
<th>Unit Field</th>
</tr>
</thead>
<tbody>
<tr>
<td>NETWR</td>
<td>MEASURE_SUM</td>
<td>SUM</td>
<td></td>
<td>WAERK</td>
</tr>
</tbody>
</table>

**Figure 26** Mapping for postprocessor BPM_PP_DETAILLIST_VAL

The source field is the field which contains the monetary value, the unit field shows the field name of the currency. As the single values are added up for the value benchmarking, the correct target field is MEASURE_SUM and the aggregation function is SUM.

The mapping for postprocessor BPM_PP_ALERTING is only relevant for alert notifications based on the distinct values of the group-by fields. To use this functionality of Business Process Monitoring a certain setup in the Business Process Monitoring Setup session is required which will not be explained here. For the usage of Business Process Analytics and normal Business Process Monitoring it is not necessary to maintain a mapping for this postprocessor.

<table>
<thead>
<tr>
<th>Source Field</th>
<th>Target Field</th>
<th>Aggregation Function</th>
<th>Function Parameter</th>
<th>Unit Field</th>
</tr>
</thead>
<tbody>
<tr>
<td>PKART</td>
<td>INFO1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>VKTYP</td>
<td>INFO2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SKTYP</td>
<td>INFO3</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>VEXOG</td>
<td>INFO4</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>VTNEG</td>
<td>INFO5</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>VSTEL</td>
<td>INFO6</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>LLAND</td>
<td>INFO7</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Figure 27** Example for mapping for postprocessor BPM_PP_ALERTING

With button ‘Create Mapping Proposal’ (see Figure 23, number 2) it is possible to maintain the mapping for the both postprocessors BPM_PP_ANACUBE and BPM_PP_ALERTING automatically, assumed it is a common key figure, for which the measured value is based on the number of lines in the detail list (as we have it here). It is recommended always to use this functionality if possible as the relevant customizing is checked: if there is an error e.g. regarding the group-by parameters, the mapping is maintained only partly or not at all. Entries regarding value-based analysis are not maintained automatically!

As a prerequisite for the automatically created mapping proposal, for all group-by parameters the 'BI Para' and the Semantic ID (see Figure 20) resp. in the detail list the 'Parameter ID' and the 'Semantic ID' (see Figure 22) have to be maintained correctly. Consider chapter 8.3 for details regarding the mapping for different calculations.

When you press the button ‘Create Mapping Proposal’, a popup with a warning appears, as the operation will remove the existing mapping for the both postprocessors BPM_PP_ALERTING and BPM_PP_ANACUBE.
6.3 Define Description Text on Key Figure Level

Via button ‘Key Figure Description’ (Figure 9, marked with number 4) you can enter a description which is key figure specific. The description text should describe shortly the meaning of the key figure in business terms. In Solution Manager, when creating a new key figure data collection from the local catalog in the Solution Documentation the text will be displayed when clicking on a key figure.

Figure 28  Description text of a key figure during setup

6.4 Define Instruction Text on Key Figure Level

Via button ‘Instructions’ (Figure 9, marked with number 5) you can enter an instruction which is key figure specific.

The instruction text should explain the purpose of the key figure. Insert some explanation on: What does the key figure measure, describe the available selection parameter and you can add important hints e.g. regarding the performance of this collector (like: recommended to run this key figure in background to avoid time out).

In Solution Manager, when creating a new key figure data collection from the local catalog in the Solution Documentation the text will be displayed when clicking on a key figure.
Figure 29  Instruction text of a key figure during setup
7 Checking the customizing with report /SSA/EMR

Since ST-A/PI 01S SP0, report /SSA/EMR (BPAnalytics: Maintenance Reports) is available.

Figure 30 Report /SSA/EMR provides a tool to check the consistency of the customizing

Among others, the report contains a tool to check the customizing for project ECU (as of ST-A/PI 01T SP1 you will need to enter the code ADMIN to see this tool). After choosing the button ‘Check Customizing’, on the next screen you may choose a certain check group using the value help (not mandatory).

Figure 31 Choose project ECU to check the customizing of the customer exit
Figure 32  Result of the customizing check

The following checks are executed by the report:

- Function Pools with "VAL* should contain value functions
- Flag NO EXTENSION should be active for fields DATUM and DSNAM
- Checkboxes: Customize possible Values
- Tech Name 2 of parameters should not be initial
- Several Checks on BI Gen Parameter
- Validity check on Category, function pool and post-processing indicator
- Is the instruction resp. the long text maintained?
- Checkboxes: Do not use the same checkbox for different parameters
- Customizing of same parameter in different key figures not identical
- Is the correct UNIT maintained?
- Is TECHNAME2 unique on key figure level?
- SEMID on Parameter and detail list should be the same
- Check data-type of Parameter
- TECHNAME2 for Parameter should be the same as FIELDNAME in APPMOND
- Value Benchmarking not allowed for some key figures
- Check function pool for lead time and automation rate key figures
- Group by Parameter should be part of detail list
- PARA_ID assigned to the correct field in detail list?
- Same SEMID on parameter and detail list definition
- Detail list with value fields: Either a currency field for every value field or only one currency field at all
- AC key figures shall not have an Aging Flag in the detail list
- There should be only one aging flag per figure
- Aging flag is not allowed for several function pools
- Is there a field with aging flag activated?
- User Command maintained when hot spot field active?
- Check if text in detail list is maintained when TECHNAM2 is initial
- Check if DDICTABLE and DDICFIELD do exist
- A value field must be assigned to SEMID BC_NETVAL
- The parameter ID of a parameter should be entered in APPMOND
• Mapping for post processing indicator TBI_LT correct?
• Group by field mapping for Postprocessor
• Mapping of Info Fields Postprocessor BPM_PP_ALERTING
• Mapping for all key figures with function pool "VAL"
• Key figures of type lead-time need a mapping definition
• When DDIC data element is maintained, a subroutine cannot be used
• Semantic Repository: When a text Table is maintained, a text field must be maintained
• Semantic Repository: When a filter field is maintained, a filter value must be maintained

⚠️ Attention: An error in the result list does not necessarily mean that it is really an error, as there can be many situations for which the rules used by the report are too strict! The result of the report must be evaluated manually step by step.
8 Functions and mapping in Business Process Analytics

8.1 Available Function Pools for Business Process Analytics

There are different function pools which can be assigned to each key figure (compare chapter 6.1). A function pool summarizes a combination of functions in Business Process Analytics. Those functions are for example Benchmarking, Trend Analysis, or Age Analysis. Depending on the selected Function Pool, different functions will be available for a key figure in Business Process Analytics. There is a naming convention for the function pools and currently there are different types available:

<table>
<thead>
<tr>
<th>Function Pool</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>BPM_ANALYTICS_TREND</td>
<td>Only a trend analysis is available in Business Process Analytics</td>
</tr>
<tr>
<td>BPM_ANALYTICS_BENCHM_TREND</td>
<td>Trend analysis and benchmarking are available in Business Process Analytics</td>
</tr>
<tr>
<td>BPM_ANALYTICS_BL_*</td>
<td>Set of function pools available for backlog key figures</td>
</tr>
<tr>
<td>BPM_ANALYTICS_LT_*</td>
<td>Set of function pools available for lead time key figures</td>
</tr>
<tr>
<td>BPM_ANALYTICS_TH_*</td>
<td>Set of function pools available for throughput key figures</td>
</tr>
</tbody>
</table>

Table 4 Naming convention for function pools

The following tables show the function pools which should be used, together with all functions of each pool.
### Table 5  Function pools with very limited functions in Business Process Analytics

<table>
<thead>
<tr>
<th>Function Pool</th>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>BPM_ANALYTICS_BENCHM_TREND</td>
<td>BENCHMARKING</td>
<td>Benchmarking</td>
</tr>
<tr>
<td></td>
<td>TREND_ANA</td>
<td>Trend Analysis</td>
</tr>
<tr>
<td>BPM_ANALYTICS_TREND</td>
<td>TREND_ANA</td>
<td>Trend Analysis</td>
</tr>
</tbody>
</table>

### Table 6  Function pools for backlog key figures. The shortcut _REF_ stands for ‘reference period’. For key figures analyzing a reference period, an age analysis makes no sense is not included accordingly.

<table>
<thead>
<tr>
<th>Function Pool</th>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>BPM_ANALYTICS_BL_REF_STD</td>
<td>DETAIL_ANA</td>
<td>Detail Analysis</td>
</tr>
<tr>
<td></td>
<td>TREND_ANA</td>
<td>Trend Analysis</td>
</tr>
<tr>
<td></td>
<td>BENCHMARKING</td>
<td>Benchmarking</td>
</tr>
<tr>
<td></td>
<td>BENCHMARKING_ADV</td>
<td>Advanced Benchmarking</td>
</tr>
<tr>
<td>BPM_ANALYTICS_BL_REF_VAL</td>
<td>DETAIL_ANA</td>
<td>Detail Analysis</td>
</tr>
<tr>
<td></td>
<td>TREND_ANA</td>
<td>Trend Analysis</td>
</tr>
<tr>
<td></td>
<td>TREND_ANA_VAL</td>
<td>Trend Analysis (value based)</td>
</tr>
<tr>
<td></td>
<td>BENCHMARKING</td>
<td>Benchmarking</td>
</tr>
<tr>
<td></td>
<td>BENCHMARKING_VAL</td>
<td>Benchmarking (value based)</td>
</tr>
<tr>
<td></td>
<td>BENCHMARKING_ADV</td>
<td>Advanced Benchmarking</td>
</tr>
<tr>
<td></td>
<td>BENCHMARKING_ADV_VAL</td>
<td>Advanced Benchmarking (value based)</td>
</tr>
<tr>
<td>BPM_ANALYTICS_BL_STD</td>
<td>DETAIL_ANA</td>
<td>Detail Analysis</td>
</tr>
<tr>
<td></td>
<td>TREND_ANA</td>
<td>Trend Analysis</td>
</tr>
<tr>
<td></td>
<td>BENCHMARKING</td>
<td>Benchmarking</td>
</tr>
<tr>
<td></td>
<td>BENCHMARKING_ADV</td>
<td>Advanced Benchmarking</td>
</tr>
<tr>
<td></td>
<td>AGE_ANA</td>
<td>Age Analysis</td>
</tr>
<tr>
<td>BPM_ANALYTICS_BL_STD_NOAGE</td>
<td>DETAIL_ANA</td>
<td>Detail Analysis</td>
</tr>
<tr>
<td></td>
<td>TREND_ANA</td>
<td>Trend Analysis</td>
</tr>
<tr>
<td></td>
<td>BENCHMARKING</td>
<td>Benchmarking</td>
</tr>
<tr>
<td></td>
<td>BENCHMARKING_ADV</td>
<td>Advanced Benchmarking</td>
</tr>
<tr>
<td>BPM_ANALYTICS_BL_VAL</td>
<td>DETAIL_ANA</td>
<td>Detail Analysis</td>
</tr>
<tr>
<td></td>
<td>TREND_ANA</td>
<td>Trend Analysis</td>
</tr>
<tr>
<td></td>
<td>TREND_ANA_VAL</td>
<td>Trend Analysis (value based)</td>
</tr>
<tr>
<td></td>
<td>BENCHMARKING</td>
<td>Benchmarking</td>
</tr>
<tr>
<td></td>
<td>BENCHMARKING_VAL</td>
<td>Benchmarking (value based)</td>
</tr>
<tr>
<td></td>
<td>BENCHMARKING_ADV</td>
<td>Advanced Benchmarking</td>
</tr>
<tr>
<td></td>
<td>BENCHMARKING_ADV_VAL</td>
<td>Advanced Benchmarking (value based)</td>
</tr>
<tr>
<td></td>
<td>AGE_ANA</td>
<td>Age Analysis</td>
</tr>
<tr>
<td></td>
<td>AGE_ANA_VAL</td>
<td>Age Analysis incl. value based analysis</td>
</tr>
<tr>
<td>BPM_ANALYTICS_BL_VAL_NOAGE</td>
<td>DETAIL_ANA</td>
<td>Detail Analysis</td>
</tr>
<tr>
<td></td>
<td>TREND_ANA</td>
<td>Trend Analysis</td>
</tr>
<tr>
<td></td>
<td>TREND_ANA_VAL</td>
<td>Trend Analysis (value based)</td>
</tr>
<tr>
<td></td>
<td>BENCHMARKING</td>
<td>Benchmarking</td>
</tr>
<tr>
<td></td>
<td>BENCHMARKING_VAL</td>
<td>Benchmarking (value based)</td>
</tr>
<tr>
<td></td>
<td>BENCHMARKING_ADV</td>
<td>Advanced Benchmarking</td>
</tr>
<tr>
<td></td>
<td>BENCHMARKING_ADV_VAL</td>
<td>Advanced Benchmarking (value based)</td>
</tr>
<tr>
<td>Function Pool</td>
<td>Function</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------------</td>
<td>------------</td>
<td>----------------------------------</td>
</tr>
<tr>
<td>BPM_ANALYTICS_LT_STD</td>
<td>TREND_ANA</td>
<td>Trend Analysis</td>
</tr>
<tr>
<td></td>
<td>DETAIL_ANA</td>
<td>Detail Analysis</td>
</tr>
<tr>
<td></td>
<td>BENCHMARKING</td>
<td>Benchmarking</td>
</tr>
<tr>
<td></td>
<td>BENCHMARKING_ADV</td>
<td>Advanced Benchmarking</td>
</tr>
<tr>
<td>BPM_ANALYTICS_LT_VAL</td>
<td>TREND_ANA</td>
<td>Trend Analysis</td>
</tr>
<tr>
<td></td>
<td>TREND_ANA_VAL</td>
<td>Trend Analysis (value based)</td>
</tr>
<tr>
<td></td>
<td>DETAIL_ANA</td>
<td>Detail Analysis</td>
</tr>
<tr>
<td></td>
<td>BENCHMARKING</td>
<td>Benchmarking</td>
</tr>
<tr>
<td></td>
<td>BENCHMARKING_VAL</td>
<td>Benchmarking (value based)</td>
</tr>
<tr>
<td></td>
<td>BENCHMARKING_ADV</td>
<td>Advanced Benchmarking</td>
</tr>
<tr>
<td></td>
<td>BENCHMARKING_ADV_VAL</td>
<td>Advanced Benchmarking (value based)</td>
</tr>
</tbody>
</table>

Table 7  Function pool for lead time key figures.

<table>
<thead>
<tr>
<th>Function Pool</th>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>BPM_ANALYTICS_TH_STD</td>
<td>TREND_ANA</td>
<td>Trend Analysis</td>
</tr>
<tr>
<td></td>
<td>DETAIL_ANA</td>
<td>Detail Analysis</td>
</tr>
<tr>
<td></td>
<td>BENCHMARKING</td>
<td>Benchmarking</td>
</tr>
<tr>
<td></td>
<td>BENCHMARKING_ADV</td>
<td>Advanced Benchmarking</td>
</tr>
<tr>
<td>BPM_ANALYTICS_TH_VAL</td>
<td>TREND_ANA</td>
<td>Trend Analysis</td>
</tr>
<tr>
<td></td>
<td>TREND_ANA_VAL</td>
<td>Trend Analysis (value based)</td>
</tr>
<tr>
<td></td>
<td>DETAIL_ANA</td>
<td>Detail Analysis</td>
</tr>
<tr>
<td></td>
<td>BENCHMARKING</td>
<td>Benchmarking</td>
</tr>
<tr>
<td></td>
<td>BENCHMARKING_VAL</td>
<td>Benchmarking (value based)</td>
</tr>
<tr>
<td></td>
<td>BENCHMARKING_ADV</td>
<td>Advanced Benchmarking</td>
</tr>
<tr>
<td></td>
<td>BENCHMARKING_ADV_VAL</td>
<td>Advanced Benchmarking (value based)</td>
</tr>
</tbody>
</table>

Table 8  Function pools for throughput key figures.

Other available function pools ending with _BPA are only necessary for standard key figures delivered in the ST-A/PI. Function pools containing the abbreviation _QTY_ provide the functionality “Quantity based analysis” which is not available so far, therefore these function pools are not mentioned here.
8.2 Necessity of Mapping for Business Process Analytics

As already explained in chapter 6.2, for different functions in Business Process Analytics it is necessary to map the fields of the detail list to the fields of the data target in the BW cube of the SAP Solution Manager. The necessity to maintain mapping depends on the analytical function and the postprocessing indicator.

For certain postprocessors the mapping always must be maintained, for others only if the measured value is not based on the number of lines in the detail list, but on e.g. a count per line which is cumulated. Examples for the latter case are key figures with grouped lists (e.g. key figure “Sales Documents created (aggregated)”, KPSD000101) as a detail list.

Together with the function pool, the postprocessing indicator determines which postprocessors are executed respectively how the selected data from the detail list will be transformed to fill the data targets in the BW Cube. The postprocessors are used for different analytical functions.

<table>
<thead>
<tr>
<th>Postprocessor</th>
<th>Analytical Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>BPM_PP_ANACUBE</td>
<td>Benchmarking (number resp. value based), Trend Analysis (number resp. value based)</td>
</tr>
<tr>
<td>BPM_PP_DETAILLIST</td>
<td>Advanced Benchmarking, Age Analysis</td>
</tr>
<tr>
<td>BPM_PP_DETAILLIST_VAL</td>
<td>Advanced Benchmarking (value based), Age Analysis</td>
</tr>
<tr>
<td>BPM_PP_ANACUBE_TR</td>
<td>Benchmarking, Trend Analysis ((number resp. value based)</td>
</tr>
<tr>
<td>BPM_PP_ANACUBE_LT</td>
<td>Benchmarking, Trend Analysis</td>
</tr>
</tbody>
</table>

Table 9 Assignment of analytical functions to postprocessors

For all key figures of category ‘BL’ and those of category ‘TH’ analyzing only one day per collector run, the postprocessing indicator normally typically is ‘TBI_STD’. The following table shows for which function pools in combination with postprocessing indicator TBI_STD a mapping is needed. The entry ‘Optional’ in column
‘Mapping required?’ means that it is not necessary to maintain it, but in case the mapping is maintained then it must be done correctly.

<table>
<thead>
<tr>
<th>Postprocessing Indicator: TBI_STD</th>
<th>Postprocessors</th>
<th>Mapping required?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Function pools</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BPM_ANALYTICS_TH_STD</td>
<td>BPM_PP_ANACUBE</td>
<td>Optional for normal key figure</td>
</tr>
<tr>
<td>BPM_ANALYTICS_BL_STD</td>
<td></td>
<td>YES for key figures with grouped detail list</td>
</tr>
<tr>
<td>BPM_ANALYTICS_BL_STD_NOAGE</td>
<td>BPM_PP_DETAILLIST</td>
<td>Optional for normal key figure</td>
</tr>
<tr>
<td>BPM_ANALYTICS_BL_REF_STD</td>
<td></td>
<td>YES for key figures with grouped detail list</td>
</tr>
<tr>
<td>BPM_ANALYTICS_TH_VAL</td>
<td>BPM_PP_ANACUBE</td>
<td>YES (Mapping to fill 0SM_SUMV is required)</td>
</tr>
<tr>
<td>BPM_ANALYTICS_BL_VAL</td>
<td>BPM_PP_DETAILLIST</td>
<td>YES for key figures with grouped detail list</td>
</tr>
<tr>
<td>BPM_ANALYTICS_BL_VAL_NOAGE</td>
<td>BPM_PP_DETAILLIST</td>
<td>Optional for normal key figure</td>
</tr>
<tr>
<td>BPM_ANALYTICS_BL_REF_VAL</td>
<td></td>
<td>YES for key figures with grouped detail list</td>
</tr>
</tbody>
</table>

Table 10  Mapping for key figures with Postprocessing Indicator TBI_STD

Postprocessing Indicator “TBI_TR” ensures that no documents are written more than once to the cube. This is important as “TBI_TR” is used for throughput key figures which analyze a defined time range (e.g. last 10 days) and in case the data collector runs every day it is probable that one document is collected in more than one collector runs.

Throughput key figures with postprocessing Indicator “TBI_TR” must be defined with a detail list on single document level (no grouped list).

<table>
<thead>
<tr>
<th>Postprocessing Indicator: TBI_TR</th>
<th>Postprocessors</th>
<th>Mapping required?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Function pools</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BPM_ANALYTICS_TH_STD</td>
<td>BPM_PP_ANACUBE_TR</td>
<td>YES</td>
</tr>
<tr>
<td>BPM_ANALYTICS_TH_VAL</td>
<td>BPM_PP_ANACUBE_TR</td>
<td>YES</td>
</tr>
</tbody>
</table>

Table 11  Mapping for key figures with Postprocessing Indicator TBI_TR

For key figures of category ‘AC’ the Postprocessing Indicator typically is ‘TBI_LT’.

Postprocessing Indicator “TBI_LT” ensures that no documents are written more than once to the cube. This is important as e.g. Lead Time key figures collect data for a certain reference period.

<table>
<thead>
<tr>
<th>Postprocessing Indicator: TBI_LT</th>
<th>Postprocessors</th>
<th>Mapping required?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Function pools</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BPM_ANALYTICS_LT_STD</td>
<td>BPM_PP_ANACUBE_LT</td>
<td>YES</td>
</tr>
<tr>
<td>BPM_PP_DETAILLIST</td>
<td></td>
<td>Optional</td>
</tr>
<tr>
<td>BPM_ANALYTICS_LT_VAL</td>
<td>BPM_PP_ANACUBE_LT</td>
<td>YES</td>
</tr>
<tr>
<td>BPM_PP_DETAILLIST_VAL</td>
<td></td>
<td>YES</td>
</tr>
</tbody>
</table>

Table 12  Mapping for key figures with Postprocessing Indicator TBI_LT

### 8.3 Maintenance of Mapping for Business Process Analytics

In this chapter, with the help of some standard key figures it will be explained how the mapping has to be maintained. There are some general recommendations which should be considered when the mapping is maintained:
- Use the value help to maintain the mapping, especially for the target field and the aggregation function. Depending on the basic customizing and the postprocessor only allowed entries are shown.
- Do not use target field OSMD_TIHM (Reference Timestamp) for any backlog key figure with category ‘BL’ (Backlog). The target field is filled internally with the date when the Postprocessing was started.

8.3.1 Postprocessor BPM_PP_ANACUBE

For common key figures, for which the measured value is based on the number of lines in the detail list, the mapping of the postprocessor looks as follows. The mapping can be done by using the button “Create Mapping Proposal”. If no value based analysis should be available, it is optional to maintain this mapping.

Figure 35 Correlation of mapping for postprocessor BPM_PP_ANACUBE (example: project EKP, monitor SD0001, key figure 26)
Attention: The mapping which is necessary for the value based trend analysis must be entered manually, it is not done automatically using button “Create Mapping Proposal”.

Figure 36 Correlation of mapping for postprocessor BPM_PP_ANACUBE providing value based trend analysis (example: project EKP, monitor SD0002, key figure 6)
In case of e.g. grouped lists, for which the measured value does not reflect the number of lines of the detail list, it is mandatory to maintain the mapping.

Figure 37 Correlation of mapping for postprocessor BPM_PP_ANACUBE for a key figure with a grouped list (example: project EKP, monitor SD0001, key figure 01)

8.3.2 Postprocessor BPM_PP_DETAILLIST

For common key figures, for which the measured value is based on the number of lines in the detail list, the mapping of this postprocessor is not necessary; therefore, also for the most standard key figures it is not maintained. If there is no mapping but there is a mapping for postprocessor BPM_PP_ANACUBE this is used instead, and the target field OSMD_SUM is replaced by MEASURE_SUM internally. The following figure shows how it would have to be maintained. The same mapping is also generated internally as a fallback in case there is neither a mapping for BPM_PP_DETAILLIST nor BPM_PP_ANACUBE.
Figure 38  Correlation of mapping for postprocessor BPM_PP_DETAILLIST for normal key figures (example: project EKP, monitor SD0001, key figure 26)

In case of e.g. grouped lists for which the measured value does not reflect the number of lines of the detail list, it is mandatory to maintain the mapping for postprocessor BPM_PP_DETAILLIST. Similar to normal key figures, if there is no mapping, the mapping for postprocessor BPM_PP_ANACUBE is used instead, and the target field 0SMD_SUM is replaced by MEASURE_SUM internally.

Figure 39  Correlation of mapping for postprocessor BPM_PP_DETAILLIST for key figures with a grouped list (example: project EKP, monitor SD0001, key figure 01)
8.3.3 Postprocessor BPM_PP_DETAILLIST_VAL

For all key figures, for which the detail list contains monetary values, the mapping for postprocessor BPM_PP_DETAILLIST_VAL has to be maintained. It enables the use of advanced value benchmarking additionally to the advanced benchmarking.

![Diagram](image.png)

Figure 40  Correlation of mapping for postprocessor BPM_PP_DETAILLIST_VAL (example: project EKP, monitor SD0001, key figure 25)
8.3.4 Postprocessor `BPM_PP_ANACUBE_TR`

For all throughput key figures which analyze a time range the mapping for postprocessor `BPM_PP_ANACUBE_TR` must be maintained. The key figure category is TH in all cases.

8.3.4.1.1 Example for a throughput key figure without monetary value

"Number of CO Postings" is a typical throughput key figure key figure analyzing a time range.

![Figure 41 Correlation of mapping for postprocessor BPM_PP_ANACUBE_TR (example: project EKF, monitor FI0023, key figure 01)](image)

<table>
<thead>
<tr>
<th>Source Field</th>
<th>Target Field</th>
<th>Aggregation Function</th>
<th>Function Parameter</th>
<th>Unit Field</th>
</tr>
</thead>
<tbody>
<tr>
<td>WORKS</td>
<td>OSM_GF01</td>
<td>GROUP BY</td>
<td></td>
<td></td>
</tr>
<tr>
<td>VRSID</td>
<td>OSM_GF02</td>
<td>GROUP BY</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CLASS</td>
<td>OSM_GF03</td>
<td>GROUP BY</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Based on the BI Generic Parameters. The number of the target field (e.g. OSM_GF01) corresponds to the number of the BI generic parameter (e.g. 1)

The number of lines of the detail list is mapped to the target field OSMO_SUM

The creation date is mapped to the target field OSMO_TIDM
8.3.4.1.2 Example for a throughput key figure with monetary value

"Sales documents created" is a typical throughput key figure key figure analyzing a time range, which provides also the monetary value information in the detail list.

Figure 42 Correlation of mapping for postprocessor BPM_PP_ANACUBE_TR (example: project EKP, monitor SD0001, key figure 33)
8.3.5 Postprocessor BPM_PP_ANACUBE_LT

For all lead time key figures, the mapping for postprocessor BPM_PP_ANACUBE_LT must be maintained. The key figure category is AC in all cases.

8.3.5.1.1 Example for a typical lead time key figure

"WF-LT: FI postings WF creation > WF approval" is a typical lead time key figure.

Figure 43 Correlation of mapping for postprocessor BPM_PP_ANACUBE_LT (example: project EKF, monitor FI0001, key figure 75)
8.3.5.1.2 Example for an average count key figure

"Automation rate: FI-GL clearing [%]" is an example for a key figure showing a calculated average value as measured value. As the result is a percentage, the unit of the key figure has to be PERCENT.

Figure 44 Correlation of mapping for postprocessor BPM_PP_ANACUBE_LT (example: project EKF, monitor FI0021, key figure 40)
8.4 Maintenance of Mapping for Business Process Monitoring

To enable alert notifications based on the distinct values of the group-by fields it is necessary to maintain a mapping.

<table>
<thead>
<tr>
<th>Postprocessor</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>BPM_PP_ALERTING</td>
<td>Alerting</td>
</tr>
</tbody>
</table>

Table 13  There is one postprocessor used for alert notifications

8.4.1 Postprocessor BPM_PP_ALERTING

When this extended functionality for alert notifications based on the distinct values of the group-by fields is used, the mapping of the postprocessor needs to be maintained. The mapping can be done easily by using the button “Fill InfoField”. For the usage of Business Process Analytics and normal Business Process Monitoring it is not necessary to maintain a mapping for this postprocessor.

Figure 45  Correlation of mapping for postprocessor BPM_PP_ALERTING (example: project EKP, monitor SD0001, key figure 25)
8.5 Maintenance of new Semantic IDs in the Semantic Repository

The assignment of a Semantic ID to a column of a detail list of a key figure allows uploading the language dependent text description into BW. The Semantic Repository within report /SSA/EXM offers Semantic IDs for the most important fields used in ERP and for many used in CRM and SRM. Especially in the case of fields in the customer namespace but also for standard fields it might happen that there is no Semantic ID available. In the following it will be explained how to create Semantic IDs in the customer namespace.

Run the report /SSA/EXM and enter ECU as project ID. Within the report, the Semantic Repository can be entered via the menu.

Figure 46 Enter the Semantic repository via the menu

Inside the Semantic Repository you see a list of all Semantic IDs which are shipped with the ST-A/PI. All these standard Semantic IDs cannot be changed when report /SSA/EXM is executed for project ECU. Semantic IDs created under project ECU are stored with an own key using a separate namespace: The Semantic ID has to start either with Y or Z. To enter a new record, you must scroll down to the end of the list.

According to the naming convention, the name of the Semantic ID should contain the name of the data element to which it refers. This makes it easier to assign the correct Semantic ID to a field.

The column ‘Description’ contains the description which is shown later e.g. in the value help.

Figure 47 Maintenance of Semantic IDs (left side)
In the following, with the help of the five standard Semantic IDs listed in Table 14 it is explained how the maintenance works.

<table>
<thead>
<tr>
<th>Semantic ID</th>
<th>DDIC Data Element</th>
<th>Maintained Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 ERP_GBSTK</td>
<td>GBSTK</td>
<td>DDIC Data Element</td>
</tr>
<tr>
<td>2 ERP_LLAND</td>
<td>LLAND</td>
<td>DDIC Data Element</td>
</tr>
<tr>
<td>3 ERP_PO_ART</td>
<td>BSART</td>
<td>Text table, key field, text field, filter field, filter value</td>
</tr>
<tr>
<td>4 ERP_EKORG</td>
<td>EKORG</td>
<td>Text table, key field, text field</td>
</tr>
<tr>
<td>5 ERP_BUKRS</td>
<td>BUKRS</td>
<td>Report name, subroutine</td>
</tr>
</tbody>
</table>

Table 14  Depending on the DDIC data element different details have to be maintained in the Semantic Repository

### 8.5.1 Fixed values are maintained for DDIC data element

When fixed values are maintained in the ABAP Dictionary for a data element, beside the name of the Semantic ID and the description only the DDIC data element has to be entered. An example for this case is Semantic ID ‘ERP_GBSTK’.

The check if fixed values are available for a DDIC data element can be done by using forward navigation in the ABAP Dictionary (explained with data type ‘GBSTK’):

Navigate to the ABAP Dictionary via SAP Menu -> Tools -> ABAP Workbench -> Development -> Dictionary (or using Transaction SE11) and enter ‘GBSTK’ as data type. Then press display.

On the next screen double-click on the domain ‘STATV’. In the display of the domain, change from tab ‘Definition’ to tab ‘Value range’. Here several fixed values are available. In a case like this, it will be sufficient to enter the data element.
8.5.2 Text table is connected to DDIC data element

Also when a text table is connected to the DDIC data element, beside the name of the Semantic ID and the description only the DDIC data element has to be entered. This is the case e.g. for Semantic ID ‘ERP_LLAND’.

![Figure 50](image)

Also in this case only the DDIC data element had to be entered.

The check if a text table is connected to a DDIC data element can be done using forward navigation in the ABAP Dictionary (explained with data type ‘LLAND’):

Navigate to the ABAP Dictionary via SAP Menu -> Tools -> ABAP Workbench -> Development -> Dictionary (or using Transaction SE11) and enter ‘LLAND’ as data type. Then press display.

On the next screen double-click on the domain ‘LAND1’. In the display of the domain, change from tab ‘Definition’ to tab ‘Value range’. In case a table with all distinct values is available (here: value table ‘T005’), double-click on this table. In the display of the table, go to the text table in case there is one via the menu: Goto -> Text Table. In case there is a text table which contains the name of the domain, the description text, and the language key as it is the case in this example (table ‘T005T’), it is sufficient to enter the data element.

8.5.3 Text table is connected to DDIC data element but a filter is needed

In case a text table is connected to the DDIC data element, but a filter on a field of this table has to be used to get the correct distinct values, the DDIC data element may not be entered. Here beside the name of the Semantic ID and the description additionally the table which contains the description, the key field, the text field, the filter field, and the filter value have to be entered. This is the case e.g. for Semantic ID ‘ERP_PO_ART’.

![Figure 51](image)

Instead of the DDIC data element the table which contains the descriptions and the filter details have to be entered.

The check if a text table is connected to a DDIC data element can be done using forward navigation in the ABAP Dictionary (as explained in the previous chapter 8.5.2):

Attention: It is very important that in this case the column ‘DDIC Data Element’ remains blank, as this entry has the highest priority and further entries as a text table are not considered.
8.5.4 Table with description is available but not connected to DDIC data element

In case there is no text table connected to the DDIC data element, beside the name of the Semantic ID and the description additionally the table which contains the description, the key field and the text field must be entered. This is the case e.g. for Semantic ID ‘ERP_EKORG’. Additionally, it might be necessary to enter a filter field and a filter value as shown in chapter 8.5.3.

Figure 52 Instead of the DDIC data element the table which contains the descriptions has to be entered

The table T024E, which contains all distinct values for ‘EKORG’ does not have a text table which means that the descriptions will be retrieved from somewhere else. Here table T024E contains descriptions and can be used.

Attention: It is very important that in this case the column ‘DDIC Data Element’ remains blank, as this entry has the highest priority and further entries as a text table are not considered.

8.5.5 Sophisticated logic is necessary to get descriptions

Sometimes it might be necessary to implement a subroutine to collect the descriptions. In this case beside the name of the Semantic ID and the description additionally the report name which includes the subroutine and the name of the subroutine must be entered. This is the case e.g. for Semantic ID ‘ERP_BUKRS’.

Figure 53 A subroutine is necessary to collect the descriptions

Attention: For Semantic IDs developed in the customer namespace, Z_BPM_ECU_COLLECTOR must be entered as report name.

The code of this subroutine looks as follows:

```plaintext
*--------------------------------------------------------------------*
* Form GET_MD_ERP_BUKRS                                             *
*--------------------------------------------------------------------*
FORM GET_MD_ERP_BUKRS USING
  PF_SEMID TYPE TS_APPMONP_SEM-SEMID
  PT_VALUE TYPE TT_VALUE
  PT_SPRAS TYPE TT_SPRAS
CHANGING
  PT_MASTER_DATA TYPE TT_MD.
TYPES:
  BEGIN OF TS_SPRAS,
    SPRAS TYPE LANG,
  END OF TS_SPRAS.
DATA:
  LS_MASTER_DATA LIKE LINE OF PT_MASTER_DATA,
  LT_SPRAS TYPE TABLE OF TS_SPRAS,
```
**Customer Exit in Business Process Analytics**

```abap
LS_SPRAS LIKE LINE OF LT_SPRAS,
LT_BURKS TYPE TABLE OF T001,
IDX TYPE SY-INDEX.

FIELD-SYMBOLS:

<LS_BURKS> LIKE LINE OF LT_BURKS.

* Evaluate all used languages in table T001 for company code
SELECT DISTINCT SPRAS
FROM T001
  INTO CORRESPONDING FIELDS OF TABLE LT_SPRAS
WHERE SPRAS IN PT_SPRAS.

* no initial value allowed
DELETE LT_SPRAS WHERE SPRAS IS INITIAL.

* no lower case values allowed
LOOP AT LT_SPRAS INTO LS_SPRAS.
  IDX = SY-TABIX.
  TRANSLATE LS_SPRAS-SPRAS TO UPPER CASE.
  MODIFY LT_SPRAS FROM LS_SPRAS INDEX IDX.
ENDLOOP.

* Ensure that english version for MD is available in Analytics
READ TABLE LT_SPRAS WITH KEY SPRAS = 'E' TRANSPORTING NO FIELDS.
IF SY-SUBRC <> 0.
  LS_SPRAS = 'E'.
  APPEND LS_SPRAS TO LT_SPRAS.
ENDIF.

* Select all company codes from T001
SELECT DISTINCT BUKRS BUTXT
FROM T001
  INTO CORRESPONDING FIELDS OF TABLE LT_BURKS
WHERE BUKRS IN PT_VALUE.

* loop over all possible languages and create entry for md
CHECK PF_SEMID = 'ERP_BUKRS'.
LOOP AT LT_BURKS ASSIGNING <LS_BURKS>.
  LOOP AT LT_SPRAS INTO LS_SPRAS.

    * nested loop, but only small amount of different values (languages)
    CLEAR LS_MASTER_DATA.
    LS_MASTER_DATA-VALUE = <LS_BURKS>-BUKRS.
    LS_MASTER_DATA-TEXT = <LS_BURKS>-BUTXT.
    LS_MASTER_DATA-SPRAS = LS_SPRAS.
    APPEND LS_MASTER_DATA TO PT_MASTER_DATA.
  ENDLOOP.
ENDLOOP.

* ensure that there are no double entries
SORT PT_MASTER_DATA BY VALUE SPRAS TEXT.
DELETE ADJACENT DUPLICATES FROM PT_MASTER_DATA.
ENDFORM.
```

The purpose of the subroutine is to fill the result table PT_MASTER_DATA which consists of three columns: the value (VALUE), the language (SPRAS), and the description (TEXT).
9 Development of customer exit report

A text file with the coding of a customer exit for this example can be found as an attachment to note 1556760. It can be copied and should provide a working customer exit in your system (together with the customizing which was explained in chapter 6). The coding can be implemented into an ERP system with a release as of 470.

In this document the coding is also available in the next chapter 9.1 (but there might be problems during copying this text into a report with the format settings).

Be aware that you have to reload the monitor definitions in the SAP Solution Manager before you are able to make a setup for this customer exit.

In the chapters 9.4 (and following) the different parts of the example report are described, especially the customer specific part of the coding where adoptions have to be done.

9.1 Example coding for Z_BPM_ECU_COLLECTOR

```
*---------------------------------------------------------------------*
* Report  Z_BPM_ECU_COLLECTOR
*---------------------------------------------------------------------*

FUNCTION-POOL  Z_BPM_ECU_COLLECTOR
NO STANDARD PAGE  HEADING MESSAGE-ID S1.

* global constants for ECU
CONSTANTS:
  GC_RELID_TC LIKE /SSF/PTAB-RELID  VALUE 'TC',
  GC_PROJECTID LIKE /SSF/PTAB-PID  VALUE 'ECU',
  GC_OBJKEY_A LIKE /SSF/PTAB-OBJKEY  VALUE 'A',
  GC_OBJKEY_S LIKE /SSF/PTAB-OBJKEY  VALUE 'S'.

* Includes for basic definitions and basic functions
INCLUDE /SSF/TYPEINC.
INCLUDE /SSA/IET.
INCLUDE /SSA/IEB.
INCLUDE /SSA/IEF.
INCLUDE /SSA/IES.
INCLUDE /SSA/IEA.
INCLUDE /SSA/IEO.

****************

*TYPE definition of result detail structure
*TYPES (TS_xxxxxx_DETAIL; TT_xxxxxx_DETAIL)
TYPES: BEGIN OF TS_ABCD01_DETAIL_01,
  VBELN LIKE VKDFS-VBELN,
  VBTYP LIKE VKDFS-VBTYP,
  FKTYP LIKE VBRK-FKTYP,
  FKART LIKE VBRK-FKART,
  VKORG LIKE VBRK-VKORG,
  VTWEG LIKE VBRK-VTWEG,
  SPART LIKE VBRK-SPART,
  VSTEL LIKE VKDFS-VSTEL,
  LLAND LIKE VKDFS-LLAND,
  KUNNR LIKE VKDFS-KUNNR,
  FAKSK LIKE VKDFS-FAKSK,
  NETWR LIKE VKDFS-NETWR,
  WAERK LIKE VKDFS-WAERK,
  FKDAT LIKE VBRK-FKDAT,
END OF TS_ABCD01_DETAIL_01,
TT_ABCD01_DETAIL_01 TYPE STANDARD TABLE OF
  TS_ABCD01_DETAIL_01 WITH DEFAULT KEY.
```
TYPES:
BEGIN OF TS_ABCD01_KEYFIG,
  RESULT_FOR_COLLECTOR TYPE I,
  KF_01 TYPE I,
END OF TS_ABCD01_KEYFIG,
BEGIN OF TS_ABCD01_RESULTS,
  KEYFIG TYPE TS_ABCD01_KEYFIG,
  KF_01 TYPE TT_ABCD01_DETAIL_01,
END OF TS_ABCD01_RESULTS.

DATA: GS_ABCD01_RESULTS TYPE TS_ABCD01_RESULTS.

*************************** TOP END ********************************
*************************** Classes ********************************
*---------------------------------------------------------------------
*       CLASS ABCD01_MONOBJ DEFINITION
*---------------------------------------------------------------------
*       SD Invoices (AR) (Customer Exit)
*---------------------------------------------------------------------
CLASS ABCD01_MONOBJ DEFINITION INHERITING FROM MONITORING_OBJECT.
  PUBLIC SECTION.
*  PROTECTED SECTION.
*    METHODS:
*     COLLECT_DATA REDEFINITION,
*     " Redefinition is not necessary when there is no data collection
*     " on monitoring object level
*     GET_KF_CLASSNAME REDEFINITION.
*     " Redefinition is not necessary when the key figure class is
*     " maintained in the attributes
ENDCLASS .

*---------------------------------------------------------------------
*       CLASS ABCD01_MONOBJ IMPLEMENTATION
*---------------------------------------------------------------------
*       SD Invoices (AR) (Customer Exit)
*---------------------------------------------------------------------
CLASS ABCD01_MONOBJ IMPLEMENTATION.
  METHOD COLLECT_DATA.
   "This method is needed if the collection will be done on object level
  ENDMETHOD.                    "collect_data

  METHOD GET_KF_CLASSNAME .
   "This method is needed if the collection will be done on keyfig level
   CASE PF_KEYFIG.
     WHEN '01'.
     PF_CLASSNAME = 'ABCD01_KF_01'.
   ENDCASE.
  ENDMETHOD.                           "get_kf_classname
ENDCLASS.                           "MONOBJ_ABCD01 IMPLEMENTATION

*---------------------------------------------------------------------
*       CLASS KEYFIG_01 DEFINITION
*---------------------------------------------------------------------
*       Billing due sales documents + field LLAND
*---------------------------------------------------------------------
CLASS ABCD01_KF_01 DEFINITION INHERITING FROM KEYFIGURE_TBI.
  PUBLIC SECTION.
  DATA:
    * Table for results (as named in method GETDETAILS_TABLE_NAME)
      MT_DETAILS TYPE TT_ABCD01_DETAIL_01,
    * Selection criteria
      * The name of the parameter is identical to "Tech Name 2" in customizing
      FKART TYPE RANGE OF VBVK-FKART, "Billing Type"
      VBTYP TYPE RANGE OF VKDFS-VBTYP, "SD Document Cat.
      FKTYP TYPE RANGE OF VKDFS-FKTYP, "Billing Cat."
VKORG  TYPE  RANGE  OF  VKDFS-VKORG,  "Sales Org.
VTWEG  TYPE  RANGE  OF  VKDFS-VTWEG,  "Dist. Channel
SPART  TYPE  RANGE  OF  VKDFS-SPART,  "Division
VSTEL  TYPE  RANGE  OF  VBSK-VSTEL,  "Shipping Point
--------- start additional fields modification of standard ---------------
LLAND  TYPE  RANGE  OF  VKDFS-LLAND,  "Destination country
--------- end additional fields modification of standard ---------------
ODSNC  TYPE  TS_CHDR_STD-ODSNC,  "Overdue since
ODLESS  TYPE  TS_CHDR_STD-ODLESS,  "Overdue less
CHECK_VBUND  TYPE  TS_CHDR_STD-CHECK_VBUND,  "Excl. internal cust.

PROTECTED SECTION.

METHODS:
  SELECT_DETAILS               REDEFINITION,
  *      GET_RATING_TYPE         REDEFINITION,
  *      " Redefinition of GET_RATING_TYPE only necessary when the rating
  *      " type differs from the standard logic
  GET_DETAILS_TABLE_NAME       REDEFINITION,
  GET_MVALUE_FROM_DETAILS     REDEFINITION.

ENDCLASS.  "KEYFIG_01 DEFINITION

*---------------------------------------------------------------------
*       CLASS KEYFIG_01 IMPLEMENTATION
*---------------------------------------------------------------------
*       Billing due sales documents + field LLAND
*---------------------------------------------------------------------

CLASS  ABCD01_KF_01 IMPLEMENTATION.

METHOD  SELECT_DETAILS.

** TYPES ************************************************************

TYPES:
  BEGIN OF TS_KUNNR_VBUND,
    KUNNR  TYPE  KNA1-KUNNR,
  END OF TS_KUNNR_VBUND,
  TT_KUNNR_VBUND  TYPE  TABLE  OF  TS_KUNNR_VBUND.

** DATA ************************************************************

DATA:
* Local table for details: LT_ABCD01_Detail
  LS_ABCD01_Detail_01  TYPE  TS_ABCD01_Detail_01,
  LT_ABCD01_Detail_01  TYPE  TT_ABCD01_Detail_01,
  LT_KUNNR_VBUND  TYPE  TT_KUNNR_VBUND,
  LF_ODLESS  TYPE  SY-DATUM,
  LF_ODSNC  TYPE  SY-DATUM.

** FIELD-SYMBOLS ****************************************************

FIELD-SYMBOLS:
  <DETAILS>  TYPE  TS_ABCD01_Detail_01.

* Calculate dates for selection
  LF_ODSNC  =  SY-DATUM  -  ODSNC.
  IF NOT ODLESS IS INITIAL.
    LF_ODLESS  =  SY-DATUM  -  ODLESS.
  ELSE.
    LF_ODLESS  =  '19000101'.
ENDIF.

** Start selection
  SELECT  VKDFS-VBELN  VKDFS-VBTYP  FKYP  FKART  VKORG  VTWEG  SPART
            VSTEL  LLAND  KUNNR  FAKSK  NETWR  WAERK  FKDAT
  FROM  VKDFS  INNER  JOIN  VBUK
  ON  VKDFS-VBELN  =  VBUK-VBELN
  INTO  CORRESPONDING  FIELDS  OF  TABLE  LT_ABCD01_Detail_01
  WHERE  VKDFS-VBTYP  IN  VBTYP
  AND  FKART  IN  FKART
  AND  FKYP  IN  FKYP
  AND  FKART  IN  FKART
  AND  FKYP  IN  FKYP

  **
AND VKORG IN VKORG
AND VTWEG IN VTWEG
AND SPART IN SPART
AND VSTEL IN VSTEL
AND FKDAT LE LF_ODSNC
AND FKDAT GE LF_ODLLESS
AND VBUK-GBSTK IN ('A', 'B') .

* Exclude items of internal customers if flag CHECK_VBUND is set
  IF NOT CHECK_VBUND IS INITIAL.
  IF NOT LT_ABCD01_DETAIL_01[] IS INITIAL.
* Get internal customers
SELECT KUNNR FROM KNA1
  INTO CORRESPONDING FIELDS OF TABLE LT_KUNNR_VBUND
  FOR ALL ENTRIES IN LT_ABCD01_DETAIL_01
  WHERE KUNNR EQ LT_ABCD01_DETAIL_01-KUNNR
  AND VBUND NE '';

SORT LT_KUNNR_VBUND.
DELETE ADJACENT DUPLICATES FROM LT_KUNNR_VBUND.

* Delete the documents with internal customers
LOOP AT LT_ABCD01_DETAIL_01 ASSIGNING <DETAILS> .
  SY-SUBRC = 0 .
  READ TABLE LT_KUNNR_VBUND WITH KEY KUNNR = <DETAILS>-KUNNR
  TRANSPORTING NO FIELDS BINARY SEARCH.
  IF SY-SUBRC = 0 .
    DELETE LT_ABCD01_DETAIL_01.
  ENDIF.
  ENDFILE.
ENDLOOP.
ENDIF.
ENDIF.

SORT LT_ABCD01_DETAIL_01 ASCENDING BY VBELN.

MT_DETAILS[] = LT_ABCD01_DETAIL_01[].
ENDMETHOD.  "SELECT_DETAILS

* METHOD GET_RATING_TYPE.
  **
  ** Rating Type
  **
  **K2 Key figure
  **K0 Key figure (no rating)
  **K2 Key figure (2-step rating upwards)   GC_2_STEP_GT
  **K3 Key figure (2-step rating downwards) GC_2_STEP_LT
  **K4 Key figure (4-step rating)           GC_4_STEP
  **ST Status
  **LI Key figure list   Key figure category BL/AC   GC_2_STEP_GT
  **             Key figure category TH     GC_4_STEP
  **L0 Key figure list (no rating)
  **L2 Key figure list (2-step rating upwards)   GC_2_STEP_GT
  **L3 Key figure list (2-step rating downwards) GC_2_STEP_LT
  **L4 Key figure list (4-step rating)           GC_4_STEP
  **LS Status list
  *
  * PF_RTYPE = GC_2_STEP_GT.
*  ENDMETHOD.  "GET_RATING_TYPE

METHOD GET_DETAILS_TABLE_NAME.
  PF_DETAILSTABNAME = 'MT_DETAILS'.
ENDMETHOD.  "GET_DETAILS_TABLE_NAME

METHOD GET_MVALUE_FROM_DETAILS.
  DESCRIBE TABLE MT_DETAILS LINES PF_MV.
ENDMETHOD.  "GET_MVALUE_FROM_DETAILS
ENDCLASS.  "KEYFIG_01 IMPLEMENTATION
FORM DC_CUABCD01

DATA COLLECTOR FOR CUABCD01

FORM DC_CUABCD01

TABLES
PT_INPUTTAB TYPE TT_INPUTTAB
PT_CUSTTABC TYPE TT_APPMONC
PT_CUSTTABL TYPE TT_APPMONL
PT_CUSTTABM TYPE TT_APPMONM
PT_CUSTTABO TYPE TT_APPMONO
PT_CUSTTABA TYPE TT_APPMONA
PT_CUSTTABS TYPE TT_APPMONS
PT_CUSTTABP TYPE TT_APPMONP
PT_OUTPUTTAB TYPE TT_OUTPUTTAB

USING
PF_SOLUTION TYPE TF_SOLUTION
PF_PARAMETER TYPE TF_PARAMETER

CHANGING
PF_SUBRC LIKE SY-SUBRC.

DATA:
LO_CONFIG_DATA TYPE REF TO CONFIGURATION_DATA_MONID,
LO_COLLECTOR TYPE REF TO ABCD01_MONOBJ,
LF_PROJECT TYPE /SSF/PTAB-PID.

READ TABLE PT_INPUTTAB INDEX 1.
LF_PROJECT = PF_PARAMETER.

CREATE OBJECT LO_CONFIG_DATA
EXPORTING
PF_MONID = PT_INPUTTAB-MONID
PF_SOLUTION = PF_SOLUTION
PF_PROJECT = LF_PROJECT.

CREATE OBJECT LO_COLLECTOR
EXPORTING
PO_CONFDAT = LO_CONFIG_DATA.

CALL METHOD LO_COLLECTOR->GET_ALERTS(
CHANGING
PT_OUTPUTTAB = PT_OUTPUTTAB[]
PT_INPUTTAB = PT_INPUTTAB[]).

ENDFORM.

FORM DI_CUABCD01 TABLES
PS_OUTPUTTAB TYPE TT_APPMONI

CHANGING
PT_ERRMSG TYPE TF_ERRMSG
LF_SY_SUBRC LIKE SY-SUBRC.

DATA:
LO_CONFIG_DATA TYPE REF TO CONFIGURATION_DATA_MONID,
LO_COLLECTOR TYPE REF TO MONITORING_OBJECT.

DATA:
LS_APPMONI TYPE TS_APPMONI,
LT_APPMONI TYPE TT_APPMONI,
LF_SOLUTION TYPE TS_APPMONA-SOLUTION,
LF_MONID TYPE TS_APPMONA-MONID,
LF_OBJECT(6) TYPE C,
**Customer Exit in Business Process Analytics**

```plaintext
LF_KEYFIG(2) TYPE C.

* Coding for detail list, copied from FORM UDI_SHOWDETAILLIST
CHECK NOT PS_OUTPUTTAB[] IS INITIAL.

READ TABLE PS_OUTPUTTAB[] INTO LS_APPMONI INDEX 1.
GS_APPMONI = LS_APPMONI.
LF_SOLUTION = LS_APPMONI-SOLUTION.
LF_MONID = LS_APPMONI-MONID.

MOVE LS_APPMONI-ALERTTYPE(6) TO LF_OBJECT.
MOVE LS_APPMONI-ALERTTYPE+8(2) TO LF_KEYFIG.
CONCATENATE 'KF_' LF_KEYFIG INTO TEXT.

CREATE OBJECT LO_CONFIG_DATA
  EXPORTING
    PF_PROJECT = GC_PROJECTID
    PF_MONID = LF_MONID
    PF_SOLUTION = LF_SOLUTION.

CREATE OBJECT LO_COLLECTOR
  TYPE ABCD01_MONOBJ
  EXPORTING
    PO_CONFDAT = LO_CONFIG_DATA.

GO_COLLECTOR = LO_COLLECTOR.

CALL METHOD LO_COLLECTOR->GET_AND_SHOW_DETAILS
  CHANGING
    PT_APPMONI = PS_OUTPUTTAB[].

ENDFORM.       "DI_CUABCD01"

*--------------------------------------------------------------------
*       ALV USER_COMMAND_ABCD01
*--------------------------------------------------------------------
*  -->  i_ucomm
*  <->  c_selfield
*--------------------------------------------------------------------

FORM USER_COMMAND_ABCD01
  USING
    I_UCOMM LIKE SY-UCOMM
  CHANGING
    C_SELFIELD TYPE SLIS_SELFIELD.

DATA:
  LF_DETWEB TYPE C,
  LF_KEYFIG TYPE TS_APPMONM-KEYFIG,
  LF_CNTR TYPE TS_APPMONA-CNTR,
  LR_DETAILS TYPE REF TO DATA,
  LR_DETAILS_ALV TYPE REF TO DATA,
  LF_VBELN(10),
  LF_VBELN_HELP TYPE VKDFS-VBELN,
  LF_VBTYP(1),
  LF_VBTYP_HELP TYPE VKDFS-VBTYP.

FIELD-SYMBOLS:
  <FT> TYPE ANY,
  <FT_DETAILS> TYPE STANDARD TABLE,
  <FS_DETAILS> TYPE ANY,
  <FF> TYPE ANY.

GET PARAMETER ID 'DET_WEB' FIELD LF_DETWEB.
SET PARAMETER ID 'DET_WEB' FIELD SPACE.
```

October 2018
** Triggered from clicking on the hotspot in a detail report line **

```plaintext
CASE I_UCOMM.
  WHEN 'REFRE'.
    C_SELFIELD-REFRESH = 'X'.
    PERFORM 00F_RELOAD_DATA.
    WHEN 'CALLTA' OR 'sICl'.
      IF LF_DETWEB = 'X'.
        READ TABLE <ALV_TABLE> INDEX 1 ASSIGNING <FS_DETAILS>.
      ELSE.
        IF NOT GS_APPMONI-ALERTTYPE+8(2) IS INITIAL.
          LF_KEYFIG = GS_APPMONI-ALERTTYPE+8(2).
        ELSE.
          LF_KEYFIG = TEXT+3(2).
        ENDIF.
        LF_CNTR = GS_APPMONI-CNTR.
      ENDIF.
      CALL METHOD GO_COLLECTOR->GET_DETAILSTAB_ALV_OF_KEYFIG
        EXPORTING
          PF_KEYFIG = LF_KEYFIG
          PF_CNTR = LF_CNTR
        IMPORTING
          PR_DETAILS_ALV = LR_DETAILS_ALV.
        ASSIGN LR_DETAILS_ALV-* TO <FT_DETAILS>.
        READ TABLE <FT_DETAILS> ASSIGNING <FS_DETAILS> INDEX C_SELFIELD-TA_INDEX.
        IF SY-SUBRC <> 0.
          MESSAGE I333(S1) WITH 'Internal error!'.
        EXIT.
      ENDIF.
    ENDIF.

  WHEN C_SELFIELD-FIELDNAME.
    " If there is only one hotspot, the CASE command is not necessary."
    ASSIGN COMPONENT 'VBELN' OF STRUCTURE <FS_DETAILS> TO <FF>.
    LF_VBELN_HELP = <FF>.
    MOVE LF_VBELN_HELP TO LF_VBELN.
    ASSIGN COMPONENT 'VBTYP' OF STRUCTURE <FS_DETAILS> TO <FF>.
    LF_VBTYP_HELP = <FF>.
    MOVE LF_VBTYP_HELP TO LF_VBTYP.
    CASE LF_VBTYP.
      WHEN 'A'.
        SET PARAMETER ID 'AFN' FIELD LF_VBELN.
        CALL FUNCTION 'SAPGUI_PROGRESS_INDICATOR'
          EXPORTING
            TEXT = 'Start Transaction VA13'.
          PERFORM 00F_AUTH_TA USING 'VA13'.
        CALL TRANSACTION 'VA13' AND SKIP FIRST SCREEN.
      WHEN 'B'.
        SET PARAMETER ID 'AGN' FIELD LF_VBELN.
        CALL FUNCTION 'SAPGUI_PROGRESS_INDICATOR'
          EXPORTING
            TEXT = 'Start Transaction VA23'.
          PERFORM 00F_AUTH_TA USING 'VA23'.
        CALL TRANSACTION 'VA23' AND SKIP FIRST SCREEN.
      WHEN 'C' OR 'I' OR 'K' OR 'L'.
        SET PARAMETER ID 'AUN' FIELD LF_VBELN.
        CALL FUNCTION 'SAPGUI_PROGRESS_INDICATOR'
          EXPORTING
            TEXT = 'Start Transaction VA03'.
          PERFORM 00F_AUTH_TA USING 'VA03'.
    ENDIF.
```

October 2018
CALL TRANSACTION 'VA03' AND SKIP FIRST SCREEN.

WHEN 'E' OR 'F'.
    SET PARAMETER ID 'LBN' FIELD LF_VBELN.
    CALL FUNCTION 'SAPGUI_PROGRESS_INDICATOR'
        EXPORTING
        TEXT = 'Start Transaction VA33'.
    PERFORM 00F_AUTH_TA USING 'VA33'.
    CALL TRANSACTION 'VA33' AND SKIP FIRST SCREEN.

WHEN 'G'.
    SET PARAMETER ID 'KTN' FIELD LF_VBELN.
    CALL FUNCTION 'SAPGUI_PROGRESS_INDICATOR'
        EXPORTING
        TEXT = 'Start Transaction VA43'.
    PERFORM 00F_AUTH_TA USING 'VA43'.
    CALL TRANSACTION 'VA43' AND SKIP FIRST SCREEN.

WHEN 'J' OR 'T' OR '7'.
    SET PARAMETER ID 'VL' FIELD LF_VBELN.
    CALL FUNCTION 'SAPGUI_PROGRESS_INDICATOR'
        EXPORTING
        TEXT = 'Start Transaction VL03N'.
    PERFORM 00F_AUTH_TA USING 'VL03N'.
    CALL TRANSACTION 'VL03N' AND SKIP FIRST SCREEN.

WHEN OTHERS.
    SET PARAMETER ID 'VF' FIELD LF_VBELN.
    CALL FUNCTION 'SAPGUI_PROGRESS_INDICATOR'
        EXPORTING
        TEXT = 'Start Transaction VF03'.
    PERFORM 00F_AUTH_TA USING 'VF03'.
    CALL TRANSACTION 'VF03' AND SKIP FIRST SCREEN.
ENDCASE.

*        WHEN 'VGBEL'. " Just an example, this field does not exist
*         in the detail list of this key figure
*         In case a second hotspot should be implemented for e.g.
*         field 'VGBEL', the logic has to be implemented here.

ENDCASE.

CLEAR: LF_VBELN_HELP, LF_VBELN.
CLEAR: LF_VBTYP_HELP, LF_VBTYP.

WHEN OTHERS.
ENDCASE.
ENDFORM.

* USER_COMMAND_ABCD01

*---------------------------------------------------------------------*
* Form VH_CUABCD01
*---------------------------------------------------------------------*
* Value help for ABCD01
*---------------------------------------------------------------------*

* FORM TO GET VALUE HELP FOR PARAMETER VALUES IN MONITORING SETUP
FORM VH_CUABCD01

TABLES PT_PARAMETER TYPE TT_PARAMETER
      PT_VH_CONTENT TYPE TT_CUSTVH
      PT_CUSTTABS TYPE TT_APPMONS

USING PF_PATTERN TYPE TF_PATTERN
      PF_MONOBJ TYPE TS_APPMONO-MONOBJ
      PF_KEYFIG TYPE TS_APPMONO-KEYFIG
      PS_CUSTTABS TYPE TS_APPMONS.
DATA:
LF_SPECIAL(10) TYPE C, " indicates special F4=Help/exception
LF_CASE_STRING(20) TYPE C, " monobj+keyfig+paraid
LF_TABIX TYPE SY-TABIX,
LF_MONOBJ TYPE TS_APPMONO-MONOBJ, "Monitoring Object
LF_KEYFIG TYPE TS_APPMONO-KEYFIG, "Key Figure
LF_PARAID TYPE TS_PARAMETER-PARA_ID, "Parameter ID
LS_PARAMETER TYPE TS_PARAMETER, "Line of parameters table

* APPMONP
  LT_APPMONP TYPE TABLE OF TS_APPMONP, "Internal table for APPMONP
  LS_APPMONP TYPE TS_APPMONP, "Work Area for lt_appmonp
* PTAB_READ
  LT_LOG TYPE /SSF/LOG OCCURS 0, "Error Log from PTAB read
  LS_RETURN TYPE DDSRETVAL, "Work Area for return_tab
* Return Codes
  LF_SUBRC TYPE SY-SUBRC,
* Output
  LS_CUSTVH TYPE TS_CUSTVH. "Line for output table

CONSTANTS: LC_PROJID_ECU TYPE /SSF/PTAB-PID VALUE 'ECU'.

* Fill internal table for APPMONP directly from cluster table PTAB
PERFORM PTAB_READ_TABLE(/SSF/ULIB)
  TABLES _LT_APPMONP
  USING 'TC'
  LC_PROJID_ECU 'P'.

* Handover values from interface
  LF_MONOBJ = PF_MONOBJ.
  LF_KEYFIG = PF_KEYFIG.

* For the new typ F4-help the parameter table has one entry only
READ TABLE PT_PARAMETER INTO LS_PARAMETER INDEX 1.
  LF_PARAID = LS_PARAMETER-PARA_ID.

MOVE-CORRESPONDING LS_PARAMETER TO LS_CUSTVH.

* Read APPMONP for determining the F4=Help
  READ TABLE LT_APPMONP INTO LS_APPMONP WITH KEY MONOBJ = LF_MONOBJ
  KEYFIG = LF_KEYFIG PARA_ID = LF_PARAID.
  LF_SUBRC = SY-SUBRC.

** Check for special F4=Help requirements / exceptions
* CONCATENATE LF_MONOBJ LF_KEYFIG LF_PARAID INTO LF_CASE_STRING.
* CASE LF_CASE_STRING.
  * WHEN 'CUABCD0101BUKRS'.
  * LF_SPECIAL = 'BUKRS'.
  * WHEN OTHERS.
  * ENDCASE.

** Start Processing F4=Helps
* CASE LF_SPECIAL. "Identify Type of F4=Help and process
  * WHEN 'BUKRS'.
  *
  * DATA: BEGIN OF LT_F4_TAB OCCURS 0,
  * BUKRS TYPE T001-BUKRS,
  * BUTXT TYPE T001-BUTXT,
  * END OF LT_F4_TAB,
  * LS_F4_TAB LIKE LINE OF LT_F4_TAB.
  *
  * SELECT BUKRS BUTXT FROM T001 INTO
  * CORRESPONDING FIELDS OF TABLE LT_F4_TAB.
  *
  * SORT LT_F4_TAB.
9.2 Creating GUI Status

You must define a GUI status for the report Z_BPM_ECU_COLLECTOR. The easiest way to do this is to copy this status from an existing report in transaction SE80.

Call transaction SE80 and choose the display functionality for programs. Enter in the selection field for programs the program name /SSA/EKP.
Use the right mouse button on the entry ALV and choose from the context menu the functionality ‘Copy…’. Enter as target the program name Z_BPM_ECU_COLLECTOR. In the same way you have to copy the LIST1 status.

9.3 Testing during Development

During the development of this customer exit it will be helpful to have the possibility of testing and debugging the new coding. This cannot be done by starting the report Z_BPM_ECU_COLLECTOR from transaction SE38 because this report is only the framework for the other form routines and these form routines are called from the SAP Solution Manager.

Since ST-A/PI 01S SP2 it is possible to test the key figure directly using transaction ST13 -> TBI_REPORTS.
9.4 Changes in example report in Detail

In the following chapters the different parts of the customer exit will be shortly explained. Especially those parts of the coding which are monitor specific will be marked in green color and explained.

9.4.1 DATA declarations

```
*---------------------------------------------------------------------*
*& Report  Z_BPM_ECU_COLLECTOR                                     *
*---------------------------------------------------------------------*
```
The coding starts with a data declaration.

The following structures have to be defined:

- **TS_ABCD01_DETAIL_01** (and TT_ABCD01_DETAIL_01)

Based on this structure the result table for the select statement is defined.

It is recommended to define specific structures for each key figure to ensure that there are no empty columns in the detail list as this might cause performance problems in Business Process Analytics.
- **TS_ABCD01_KEYFIG**
  The naming for this structure has to follow the naming convention TS_<name of monitor>_KEYFIG. It always contains a field RESULT_FOR_COLLECTOR.
  For every key figure of a monitor there must be a corresponding field in this structure (like KF_01 for key figure 01 in this example).

- **TS_ABCD01_RESULTS**
  The naming for this structure must follow the naming convention TS_<name of monitor>_RESULTS. It is a deep structure and contains the structure KEYFIG and the table KF_01 (which is the result table for key figure 01).
  For every key figure of a monitor there must be a corresponding field (like KF_01 for key figure 01 in this example).

- **GS_ABCD01_RESULTS**
  This global structure is needed to show the detail list in the ALV and has to be defined for every monitor.
9.4.2 Class Definition for Monitor

For every monitor a class has to be defined. As explained in the comments in the source code, the redefinition of method COLLECT_DATA REDEFINITION is only necessary when there is a data collection on monitoring object level.

As of ST-A/PI 01Q SP2 the redefinition of method GET_KF_CLASSNAME is not necessary when the key figure class is maintained in the attributes (see Figure 11).

9.4.3 Class Implementation for Monitor

During this implementation, for every key figure of the monitor the name of the key figure specific class is assigned within the code. As of ST-A/PI 01Q SP2 the redefinition of method GET_KF_CLASSNAME is not necessary when the key figure class is maintained in the attributes (see Figure 11). The commented code of method GET_KF_CLASSNAME shows how the code would need to look.
In the definition of the key figure class the parameters which are used for the selection are defined. These are the same fields which are already named in the setup report /SSA/EXM.

💡 The name of the parameter has to be identical to "Tech Name 2" in the customizing report.

### 9.4.5 Class Implementation for Key figure
** Data ***************************************************************
DATA:
* Local table for details: LT_ABCD01_DETAIL
  LS_ABCD01_DETAIL_01 TYPE TS_ABCD01_DETAIL_01,
  LT_ABCD01_DETAIL_01 TYPE TT_ABCD01_DETAIL_01,
  LT_KUNNR_VBUND TYPE TT_KUNNR_VBUND,
  LF_ODLESS TYPE SY DATUM,
  LF_ODSNC TYPE SY DATUM.

** Field-Symbols *******************************************************
FIELD-SYMBOLS:
<DETAILS> TYPE TS_ABCD01_DETAIL_01.
* Calculate dates for selection
  LF_ODSNC = SY-DATUM - ODSNC.
  IF NOT ODLESS IS INITIAL.
    LF_ODLESS = SY-DATUM - ODLESS.
  ELSE.
    LF_ODLESS = '19000101'.
  ENDIF.
* Start selection
  SELECT VKDFS ~ VBELN VKDFS VBTYP FKART VKORG VTWEG SPART
       VSTEL LLAND KUNNR FAKSK NETWR WAERK FKDAT
  FROM VKDFS INNER JOIN VBUK
  ON VKDFS VBELN = VBUK VBELN
  INTO CORRESPONDING FIELDS OF TABLE LT_ABCD01_DETAIL_01
  WHERE VKDFS VBTYP IN VBTYP
       AND FKART IN FKART
       AND FKYP IN FKYP
       AND VKORG IN VKORG
       AND VTWEG IN VTWEG
       AND LLAND IN LLAND
       AND SPART IN SPART
       AND VSTEL IN VSTEL
       AND FKDAT LE LF_ODSNC
       AND FKDAT GE LF_ODLESS
       AND VBUK GBSTK IN 'A', 'B'.
  IF NOT CHECK_VBUND IS INITIAL.
    IF NOT LT_ABCD01_DETAIL_01[]. IS INITIAL.
      * Get internal customers
        SELECT KUNNR FROM KNA1
        INTO CORRESPONDING FIELDS OF TABLE LT_KUNNR_VBUND
        FOR ALL ENTRIES IN LT_ABCD01_DETAIL_01
        WHERE KUNNR EQ LT_ABCD01_DETAIL_01-KUNNR
        AND VBUND NE ''.
    ENDIF.
  ENDIF.
  DELETE ADJACENT DUPLICATES FROM LT_KUNNR_VBUND.
  ENDLOOP.
  IF NOT LT_ABCD01_DETAIL_01[]. IS INITIAL.
  ENDIF.
  END-SELECT-STATEMENT.
The implementation of the key figure specific class consists of 4 methods:

- Method SELECT_DETAILS contains the data selection. The selected data is written into the result table defined in method GET_DETAILS_TABLE_NAME, here it is MT_DETAILS.

- Method GET_RATING_TYPE: in this method the rating type for the key figure has to be declared. As of ST-A/PI 01Q SP2 the redefinition of GET_RATING_TYPE is only necessary when the rating type differs from the standard logic. This logic for each key figure type and the assigned rating types is documented in the commented code of GET_RATING_TYPE. The commented code also shows how the code would need to look before ST-A/PI 01Q SP2.

- Method GET_DETAILS_TABLE_NAME: here the name of the table which finally contains the results is defined.

- Method GET_MVALUE_FROM_DETAILS: here the logic is implemented how the measured value for Business Process Monitoring is determined. It is also relevant for the usage of the key figure within transaction ST13. In most cases the number of entries in the result list MT_DETAILS is counted. The measured value has to be written into the field PV_MV. For Business Process Analytics the method is not relevant as the whole functionality is based on the detail list resp. the mapping.
9.4.6 Form DC_CUABCDxx (Data Collector)

For each monitor in your customer exit you have to create one form routine DC_CUABCDXX where CU stands for the customer exit project ECU and ABCDXX is the name of the monitor.

It is mandatory that this form routine is created directly in the report Z_BPM_ECU_COLLECTOR, not e.g. stored in an include. The reason is that there is a coding scan which checks if the form routine exists.
9.4.7 Form DI_CUABCDxx (Detail information)

For each monitor you have to create a form routine DI_CUABCDXX where CU stands for the customer exit project ECU and ABCDXX is the name of your monitor. This form routine is necessary for the display of the result list.
It is mandatory that this form routine is created directly in the report Z_BPM_ECU_COLLECTOR, not e.g. stored in an include. The reason is that there is a coding scan which checks if the form routine exists.

9.4.8 Form xxD_USER_COMMAND (Interaction on Detail list)

```
*--------------------------------------------------------------------
*       ALV 01D_user command
*--------------------------------------------------------------------
*  -->  i_ucomm
*  <->  c_selfield
*--------------------------------------------------------------------

FORM 01D_USER_COMMAND
USING I_UCOMM LIKE SY-UCOMM
CHANGING C_SELFIELD TYPE SLIS_SELFIELD.

DATA:
  LF_DETWEB TYPE C,
  LF_KEYFIG TYPE TS_APPMONM-KEYFIG,
  LF_CNTR TYPE TS_APPMONA-CNTR,
  LR_DETAILS TYPE REF TO DATA,
  LRDETAILS_ALV TYPE REF TO DATA,
  LF_VBELN_HELP TYPE VKDFS-VBELN,
  LF_VBTYP_HELP TYPE VKDFS-VBTYP.

FIELD-SYMBOLS:
  <FT> TYPE ANY,
  <FT_DETAILS> TYPE STANDARD TABLE,
  <FS_DETAILS> TYPE ANY,
  <FF> TYPE ANY.

GET PARAMETER ID 'DET_WEB' FIELD LF_DETWEB.
SET PARAMETER ID 'DET_WEB' FIELD SPACE.

** Triggered from clicking on the hotspot in a detail report line
CASE I_UCOMM.
  WHEN 'REFRE'.
    C_SELFIELD-REFRESH = 'X'.
    PERFORM 00F_RELOAD_DATA.
  WHEN 'CALLITA' OR '&IC1'.
    IF LF_DETWEB = 'X'.
      READ TABLE <ALV_TABLE> INDEX 1 ASSIGNING <FS_DETAILS>.
    ELSE.
      IF NOT GS_APPMONI-ALERTTYPE+8(2) IS INITIAL.
        LF_KEYFIG = GS_APPMONI-ALERTTYPE+8(2).
      ELSE.
        LF_KEYFIG = TEXT+3(2).
      ENDIF.
      LF_CNTR = GS_APPMONI-CNTR.
    CALL METHOD GO_COLLECTOR->GET_DETAILSTAB_ALV_OF_KEYFIG
      EXPORTING
        PF_KEYFIG = LF_KEYFIG
        PF_CNTR = LF_CNTR
      IMPORTING
        PR_DETAILS_ALV = LR_DETAILS_ALV.
  ELSE.
```
ASSIGN LR_DETAILS_ALV-* TO FT_DETAILS.
READ TABLE FT_DETAILS ASSIGNING FS_DETAILS INDEX C_SELFIELD-TABINDEX.

IF SY-SUBRC <> 0.
  MESSAGE I333(S1) WITH 'Internal error!'.
  EXIT.
ENDIF.
ENDIF.

CASE C_SELFIELD: FIELDNAME
  WHEN 'VBELN'
    ASSIGN COMPONENT 'VBELN' OF STRUCTURE FS_DETAILS TO <FF>
    LF_VBELN_HELP = <FF>
    MOVE LF_VBELN_HELP TO LF_VBELN
  END_CASE.
  WHEN 'VBTYP'
    ASSIGN COMPONENT 'VBTYP' OF STRUCTURE FS_DETAILS TO <FF>
    LF_VBTYP_HELP = <FF>
    MOVE LF_VBTYP_HELP TO LF_VBTYP
  END_CASE.
  WHEN CASE LF_VBTYP:
    WHEN A
      SET PARAMETER ID 'AFN' FIELD LF_VBELN
      CALL FUNCTION 'SAPGUI_PROGRESS_INDICATOR' EXPORTING
      TEXT = 'Start Transaction VA13'
      PERFORM 00F_AUTH_TA USING 'VA13'.
      CALL TRANSACTION 'VA13' AND SKIP FIRST SCREEN.
    WHEN B
      SET PARAMETER ID 'AGN' FIELD LF_VBELN
      CALL FUNCTION 'SAPGUI_PROGRESS_INDICATOR' EXPORTING
      TEXT = 'Start Transaction VA23'
      PERFORM 00F_AUTH_TA USING 'VA23'.
      CALL TRANSACTION 'VA23' AND SKIP FIRST SCREEN.
  END_CASE.
  WHEN CASE LF_VBTYP:
    WHEN C
      SET PARAMETER ID 'AUN' FIELD LF_VBELN
      CALL FUNCTION 'SAPGUI_PROGRESS_INDICATOR' EXPORTING
      TEXT = 'Start Transaction VA03'
      PERFORM 00F_AUTH_TA USING 'VA03'.
      CALL TRANSACTION 'VA03' AND SKIP FIRST SCREEN.
    WHEN E
      SET PARAMETER ID 'LBN' FIELD LF_VBELN
      CALL FUNCTION 'SAPGUI_PROGRESS_INDICATOR' EXPORTING
      TEXT = 'Start Transaction VA33'
      PERFORM 00F_AUTH_TA USING 'VA33'.
      CALL TRANSACTION 'VA33' AND SKIP FIRST SCREEN.
    WHEN G
      SET PARAMETER ID 'KTN' FIELD LF_VBELN
      CALL FUNCTION 'SAPGUI_PROGRESS_INDICATOR' EXPORTING
      TEXT = 'Start Transaction VA43'
      PERFORM 00F_AUTH_TA USING 'VA43'.
      CALL TRANSACTION 'VA43' AND SKIP FIRST SCREEN.
    WHEN J
      SET PARAMETER ID 'VL' FIELD LF_VBELN
      CALL FUNCTION 'SAPGUI_PROGRESS_INDICATOR' EXPORTING
      TEXT = 'Start Transaction VL03N'
      PERFORM 00F_AUTH_TA USING 'VL03N'.
      CALL TRANSACTION 'VL03N' AND SKIP FIRST SCREEN.
  END_CASE.
END_CASE.
The form routine `xxD_USER_COMMAND` has to be created for each monitor.

It provides the functionality of calling a standard transaction or report when a hotspot field is clicked in a detail result list. In the above shown example depending on a certain field (the document category `VBTYP`) different standard transactions are called when in the result list the field `VBELN` is clicked.

To make this functionality work the following steps are necessary:

- Define the field(s) you want to use as a hotspot during the customizing of the result list (see chapter 6.2, Figure 21 and Figure 22). In case you define more than one hotspots in a detail list, you have to call the same user command form routine for all of them. To distinguish you have to check field `C_SELFIELD-FIELDNAME` as shown in the example.

- Enter the name of this form routine into the column ‘USER COMMAND’ during the customizing of the detail list (see chapter 6.2, Figure 21 and Figure 22)

- Develop the code for a form routine with the name defined in the previous step.

It is mandatory that this form routine is created directly in the report `Z_BPM_ECU_COLLECTOR`, not e.g. stored in an include. The reason is that there is a coding scan which checks if the form routine exists.

### 9.4.9 Form `VH_CUABCDxx` (Value help for parameters)

By default, whenever both ‘Tech. Name 1’ and ‘Tech. Name 2’ are maintained for a parameter in the customizing, as of SAP Solution Manager 7.1 SP 5 the value help of the corresponding DDIC object is called.

In case the use of a special value help is necessary it is mandatory that ‘Tech. Name 1’ is blank. Only in this case the form `VH_CUABCDxx` containing the value helps for the monitor is called.

The example below shows the thought case that the company code `VBAK-BUKRS` is needed as an additional parameter, and additionally the imagined situation that there is no value help available for this field.
The relevant part of the form VH_CUABCD01 is commented as it is only an example.

As the special value should only be called for the parameter BUKRS of the key figure 01 of monitor ABCD01 in project ECU, the string LF_CASE_STRING is concatenated containing this information. In case the value of this string is CUABCD0101BUKRS, the special F4-help ‘BUKRS’ is executed.

```
* &---------------------------------------------------------------------*
* &      Form VH_CUABCD01                                           *
* &---------------------------------------------------------------------*
*       Value help for ABCD01                                        *
*----------------------------------------------------------------------*
* FORM TO GET VALUE HELP FOR PARAMETER VALUES IN MONITORING SETUP       *
*---------------------------------------------------------------------*
FORM VH_CUABCD01(DBC01)

TABLES
  PT_PARAMETER  TYPE  TT_PARAMETER
  PT_VH_CONTENT TYPE  TT_CUSTVH
  PT_CUSTTABS   TYPE  TT_APPMONS
USING
  PF_PATTERN    TYPE  TF_PATTERN
  PF_MONOBJ    TYPE  TS_APPMONO-MONOBJ
  PF_KEYFIG    TYPE  TS_APPMONO-KEYFIG
  PS_CUSTTABS  TYPE  TS_APPMONS.

DATA:
LF_SPECIAL(10) TYPE  C, " indicates special F4-Help/exception
LF_CASE_STRING(20) TYPE  C, " monobj+keyfig+paraid
LF_TABIX       TYPE  SY-TABIX,
LF_MONOBJ      TYPE  TS_APPMONO-MONOBJ, "Monitoring Object
LF_KEYFIG      TYPE  TS_APPMONO-KEYFIG, "Key Figure
LF_PARAID      TYPE  TS_PARAMETER-PARA_ID, "Parameter ID
LS_PARAMETER   TYPE  TS_PARAMETER, "Line of parameters table
* APPMNP
LT_APPMONP     TYPE  TABLE OF TS_APPMONP, "Internal table for APPMNP
LS_APPMONP     TYPE  TS_APPMONP, "Work Area for lt_appmonp
* PTAB_READ
LT_LOG         TYPE  /SSF/LOG OCCURS 0, "Error Log from PTAB read
* Generic F4-HELPs
LS_RETURN      TYPE  DDSHERTVAL, "Work Area for return_tab
* Return Codes
LF_SUBRC       TYPE  SY-SUBRC,
* Output
LS_CUSTVH     TYPE  TS_CUSTVH. "Line for output table

CONSTANTS:
  LC_PROJID_ECU TYPE  /SSF/PTAB-PID VALUE 'ECU'.
```
* Fill internal table for APPMONP directly from cluster table PTAB

```plaintext
PERFORM PTAB_READ_TABLE(/SSF/ULIB)
    TABLES LT_APPMONP
            LT_LOG
    USING 'PC'
        LC_PROJID_ECU
        'P'.
```

* Handover values from interface

```plaintext
LF_MONOBJ  =  PF_MONOBJ.
LF_KEYFIG  =  PF_KEYFIG.
```

* For the new typ F4-help the parameter table has one entry only

```plaintext
READ TABLE PT_PARAMETER INTO LS_PARAMETER INDEX 1.
LF_PARAID  =  LS_PARAMETER-PARA_ID.
```

```plaintext
MOVE-CORRESPONDING LS_PARAMETER TO LS_CUSTVH.
```

* Read APPMONP for determining the F4-Help

```plaintext
READ TABLE LT_APPMONP INTO LS_APPMONP WITH KEY MONOBJ  =  LF_MONOBJ
            KEYFIG  =  LF_KEYFIG PARA_ID  =  LF_PARAID.
LF_SUBRC  =  SY-SUBRC.
```

** Check for special F4-Help requirements / exceptions

* CONCATENATE LF_MONOBJ LF_KEYFIG LF_PARAID INTO LF_CASE_STRING.

```plaintext
CASE LF_CASE_STRING.
    WHEN 'CUABCD0101BUKRS'.
        LF_SPECIAL  =  'BUKRS'.
    WHEN OTHERS.
        ... END CASE.
```

** Start Processing F4-Helps

```plaintext
CASE LF_SPECIAL. "Identify Type of F4-Help and process

```

```plaintext
DATA:  BEGIN OF LT_F4_TAB OCCURS 0,
            BUKRS TYPE T001-BUKRS,
            BUTXT TYPE T001-BUTXT,
        END OF LT_F4_TAB,
            LS_F4_TAB LIKE LINE OF LT_F4_TAB.
```

```plaintext
SELECT BUKRS BUTXT FROM T001 INTO CORRESPONDING FIELDS OF TABLE LT_F4_TAB.
SORT LT_F4_TAB.
DELETE ADJACENT DUPLICATES FROM LT_F4_TAB.
LOOP AT LT_F4_TAB INTO LS_F4_TAB.
    MOVE-CORRESPONDING LS_APPMONP TO LS_CUSTVH.
    LS_CUSTVH-VALUE   =  LS_F4_TAB-BUKRS.
    LS_CUSTVH-DESCR   =  LS_F4_TAB-BUTXT.
    APPEND LS_CUSTVH TO PT_VH_CONTENT.
ENDLOOP.
```

** In case a conversion routeine is assigned to the corresponding database field, the conversion needs to be reversed.

```plaintext
LOOP AT PT_VH_CONTENT INTO LS_CUSTVH.
    LF_TABIX  =  SY-TABIX.
    LS_RETURN-FIELDVAL  =  LS_CUSTVH-VALUE. "Input for next step
    PERFORM 00F_GET_FIELD_EXIT
```

** End of processing APPMONP / F4-Helps

* WHEN OTHERS. " Generic F4-Help

* END CASE. "End of processing APPMONP / F4-Helps

```plaintext
* In case a conversion routeine is assigned to the corresponding database field, the conversion needs to be reversed.

LOOP AT PT_VH_CONTENT INTO LS_CUSTVH.
    LF_TABIX  =  SY-TABIX.
    LS_RETURN-FIELDVAL  =  LS_CUSTVH-VALUE. "Input for next step
    PERFORM 00F_GET_FIELD_EXIT
```

** End of processing APPMONP / F4-Helps

* WHEN OTHERS. "Generic F4-Help

* END CASE. "End of processing APPMONP / F4-Helps
USING
LS_APPMONP-TECHNAME1
LS_APPMONP-TECHNAME2
CHANGING
LS_RETURN-FIELDVAL
LS_CUSTVH-VALUE.

IF LS_CUSTVH-VALUE IS INITIAL.
LS_CUSTVH-VALUE = LS_RETURN-FIELDVAL.
ENDIF.
MODIFY PT_VH_CONTENT FROM LS_CUSTVH INDEX LF_TABIX.
ENDLOOP.
ENDFORM.

For each monitor a form routine VH_CUABCDxx has to be implemented to provide a F4-Help for the fields used as selection criteria.

It is mandatory that this form routine is created directly in the report Z_BPM_ECU_COLLECTOR, not e.g. stored in an include. The reason is that there is a coding scan which checks if the form routine exists.
10 Further Information

10.1 Setup Business Process Analytics

The setup procedure can be found in the SAP Solution Manager WIKI - Business Process Operations -> Business Process Improvement -> Technical Information -> SAP Solution Manager 7.2.

**Note:** It is not possible to setup customer exit key figures from the Cloud Catalog.

To make the monitors available in the SAP Solution Manager, the monitor definitions need to be transferred from the managed system to a central repository in the SAP Solution Manager System.

Therefore, in the Solution Documentation you need to choose the Analytics Library and start to create a new Analytics Object. You can load or update monitors by selecting the system/client and using the 'Load' button.

![Reload Monitors from the managed system with SAP Solution Manager 7.2](image)

**Figure 57** Reload Monitors from the managed system with SAP Solution Manager 7.2

10.2 Development procedure in a 3-System Landscape

Create the Report (Z_BPM_ECU_COLLECTOR) in the development system

1. Add the customizing to the same request in report /SSA/EXM while you are in project ECU via the menu and afterwards transport everything to the test system:

![Define Monitors for Project](image)

2. After testing the developed Customer Exit, transport the ABAP report Z_BPM_ECU_COLLECTOR and the customizing into your Productive System.
10.3 Related documents

Many detailed documents with information related to Business Process Improvement can be found in the SAP Solution Manager WIKI - Business Process Operations -> Business Process Improvement.