SAP Business Data Toolset

ADD A CUSTOM FIELD IN LESS THAN 30 MINS FOR ANY BDT DRIVEN OBJECT

WITHOUT ANY BDT KNOWLEDGE

APACE
Development Ltd

Prerequisites

To be able to follow this How-To guide you need no knowledge of the Business Data Toolset at all, but you need to know how to create and maintain the fundamental ABAP development objects like function groups, function modules, table appends, data elements and so on.

What is the Business Data Toolset? It's a framework to extend SAP standard master data maintenance dialogues (there are only a few transactional data dialogues driven by BDT). It's used by SAP, SAPs development partners and SAPs customers to enhance SAPs standard transactions with new fields and checks. Not every SAP maintenance dialogue is driven by BDT technology.

Before you start to add a custom field to a standard SAP application, you must check that your SAP application is actually driven by the Business Data Toolset (BDT) and find the transaction (actually an area menu) to configure the BDT. There are several ways to do this, if you don't know the appropriate transaction for your application.

Here is a list of some applications driven by BDT with their configuration transaction. If you know of some area menus not listed here, I would be happy if you drop me a line at bdt@apadev.co.uk to include it in the next version of this guide.

BDT config transaction	Application
BUPT	Business Partner
INSO	FS-CD Insurance Object
CACSBDT	ICM commission contract
CACSBDTI	ICM commission case
CACSBDTB	ICM commission contract bundle
CACSBDTD	ICM commission document
CAMA	FI-CA master agreement
CAVT	Provider contract
CAWM	FS-CD contract account
CBWABDT	Waste Management
CFM_CR	Credit Risk Analyzer
CMS_CUS_LIQ_BDT	Liquidation UI
FMFA_BDT	Funds Management master data
FMCA_PSOB	PS-CD contract objects
FOAR00	Financial Assets Management: Real Estate / Applicant
FOI0	Real Estate: General contract
FOWB00	IS-RE economic feasibility (German)
FOTI00	Tenant Information
GRANT_BDT	Grants Management master data
PPAC	Prepaid account FI-CA
KLFZ	Risk management: Facilities
RCC00	Risk Objects
RECAMENUDEV	RE Extension
RECAMENUBDT	RE-Extension: BDT Customizing Menu
UG01	Financials Master Data

If you can't find your application in question in the list above, you can check this:

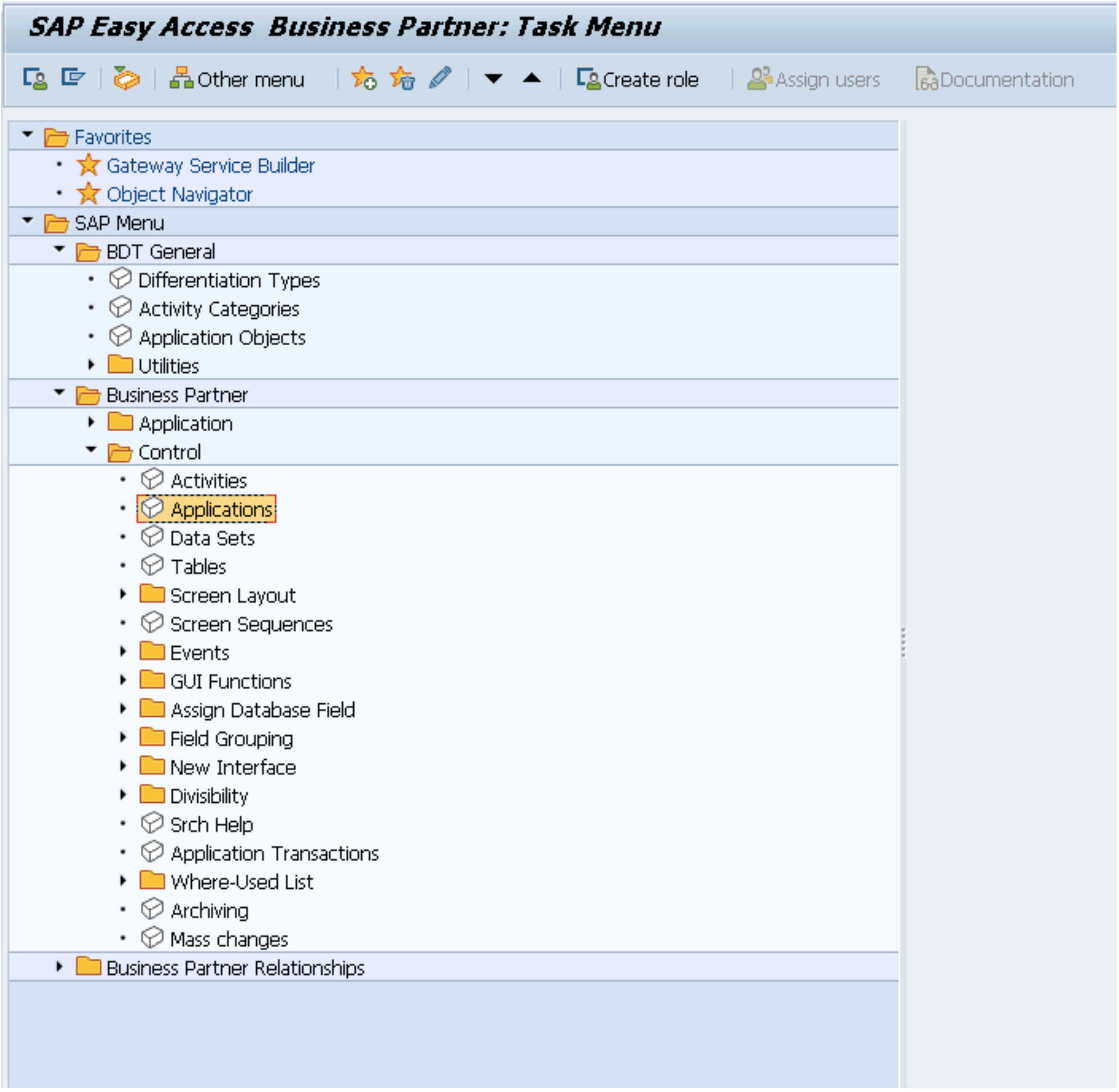
All BDT configuration transactions are implemented as area menus. Call transaction SE93 and enter transaction BUSP (a utility transaction that is part of every BDT area menu) and do a “where-used” list on area menus. This way you can generate a list with all BDT configuration transactions in your system.

If you still cannot identify the BDT configuration this way, you can search for the maintenance transaction of your SAP object within those area menus: E.g. you don’t know the BDT configuration transaction for the FS-CD insurance object, but you know the maintenance transaction for insurance objects is INSOCHANGE. Go to transaction SE93 and enter INSOCHANGE and do a where used analysis within area menus. Check if you can find an area menu that is also in the list created within the search for transaction BUSP. This is your BDT configuration transaction. If you cannot find any area menu that fits the criteria mentioned above, your SAP object is probably not driven by BDT.

If you have the area menu for your SAP application, then let’s begin to add a custom field in less than 30 minutes…

Create a new BDT application and function group (5mins)

Call the aforementioned BDT area menu. Choose Control -> Applications to create a new BDT application. From now on I will illustrate each step by using an example of the SAP Business Partner using BDT area menu BUPT. The structure of your BDT area menu should be the same.

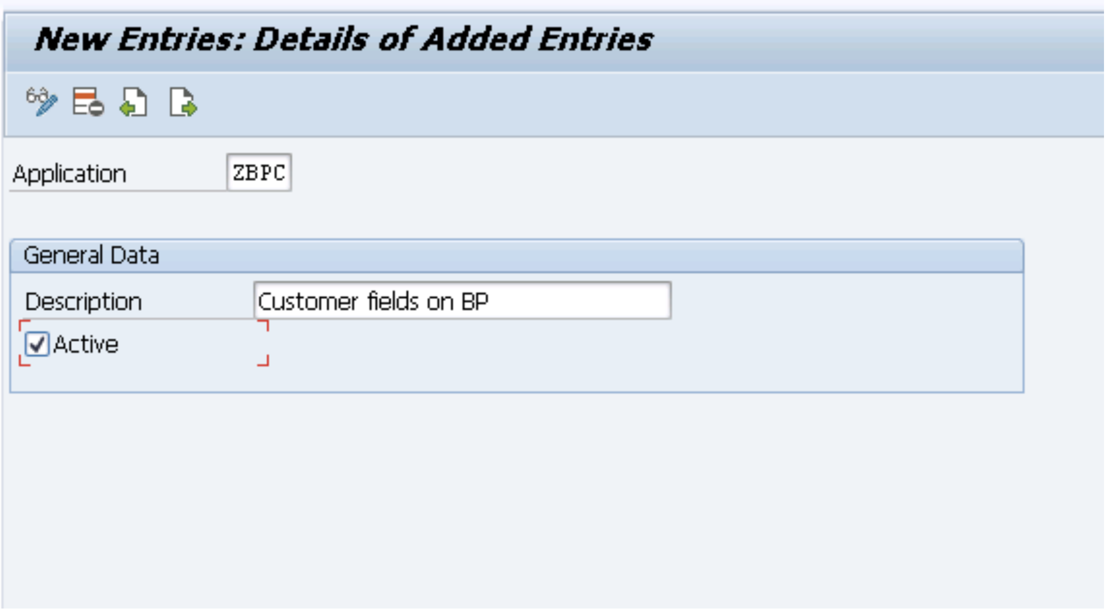


SAP Easy Access Business Partner: Task Menu

- Favorites
  - Gateway Service Builder
  - Object Navigator
- SAP Menu
  - BDT General
    - Differentiation Types
    - Activity Categories
    - Application Objects
    - Utilities
  - Business Partner
    - Application
    - Control
      - Activities
      - Applications
      - Data Sets
      - Tables
      - Screen Layout
      - Screen Sequences
      - Events
      - GUI Functions
      - Assign Database Field
      - Field Grouping
      - New Interface
      - Divisibility
      - Srch Help
      - Application Transactions
      - Where-Used List
      - Archiving
      - Mass changes
  - Business Partner Relationships

Figure 1: Application entry in the BDT area menu for Business Partner

The BDT application will be the key for all your further BDT configurations. Choose a four-character application key that is in your customer name space (starting with Z or Y). You can activate or deactivate an application, which makes it easy to switch applications without changing any coding. Set your new application to “active”. I will refer to your application key from now on as <appl>.



New Entries: Details of Added Entries

Application ZBPC

General Data

Description Customer fields on BP

☑ Active

Figure 2: Creating a new BDT application

Next create a new function group within your customer name space. The BDT naming convention is to include the application and the BDT object in the name. So if your customer name space is Z, your function group should be called Z_<appl>_<obj> with <obj> being the BDT object. In the BDT area menu you will find a section called "BDT General". This holds an entry "Application Objects". Here you'll find the 4-letter code for your object, from now on referred to as <obj>. Note: not complying with the naming convention of the function group has no effect on the functionality; it's just formalism to enhance readability.

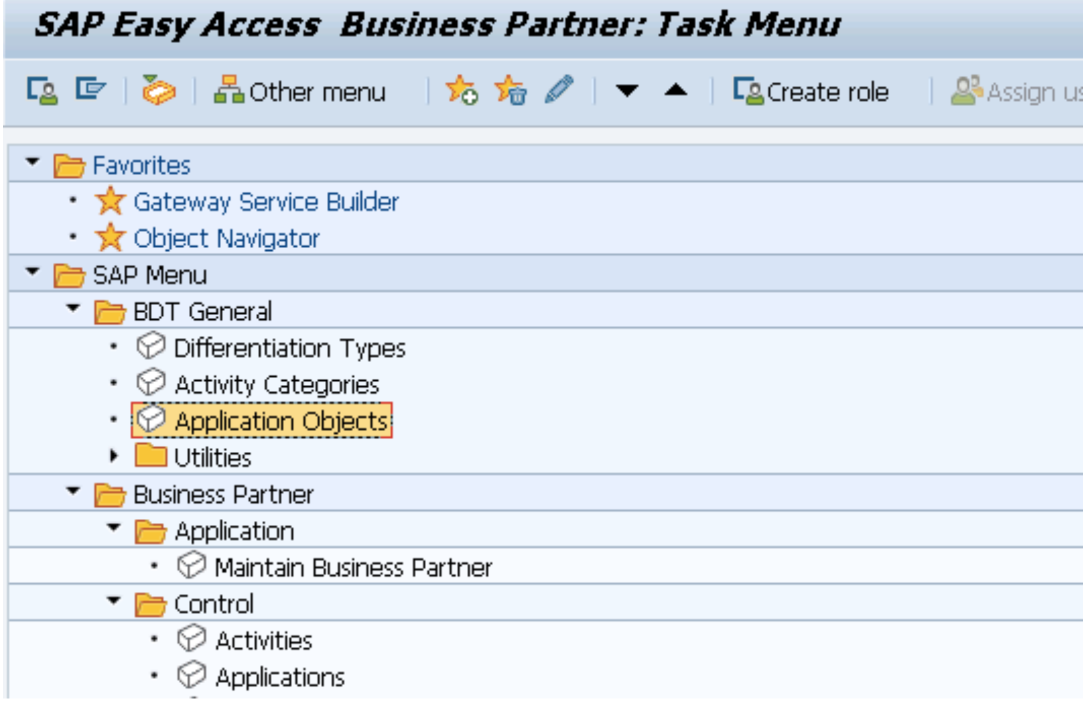
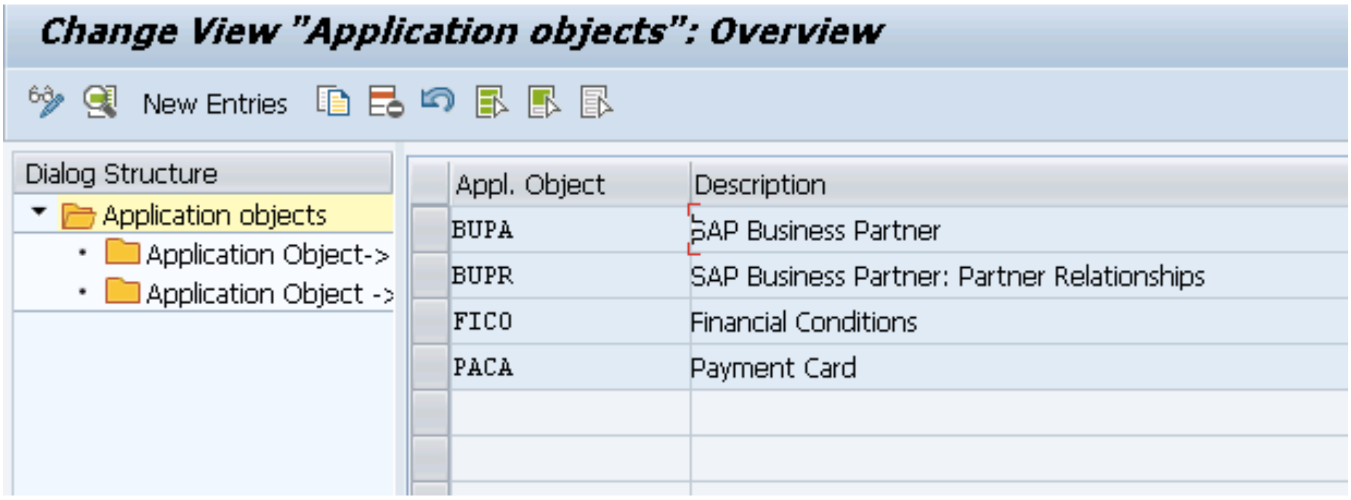


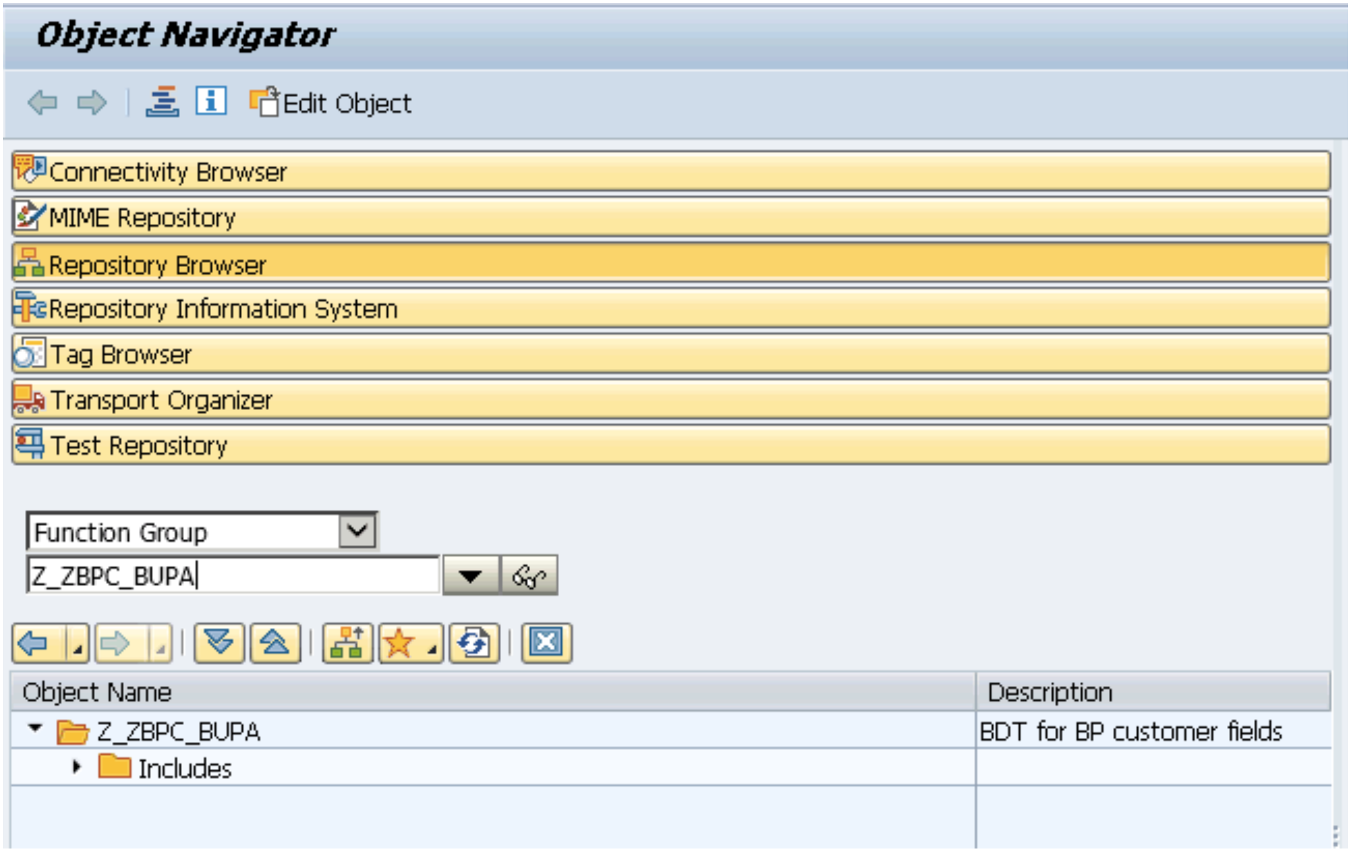
Figure 3: Application Object entry in the BDT area menu



Appl. Object	Description
BUPA	SAP Business Partner
BUPR	SAP Business Partner: Partner Relationships
FICO	Financial Conditions
PACA	Payment Card

Figure 4: Display the Business Partner Application Object

The function group you just created will hold all development objects of your BDT application. Do not spread your BDT development objects over several function groups. Rule: One BDT application = one function group.



Function Group

Z_ZBPC_BUPA

Object Name	Description
Z_ZBPC_BUPA	BDT for BP customer fields
Includes	

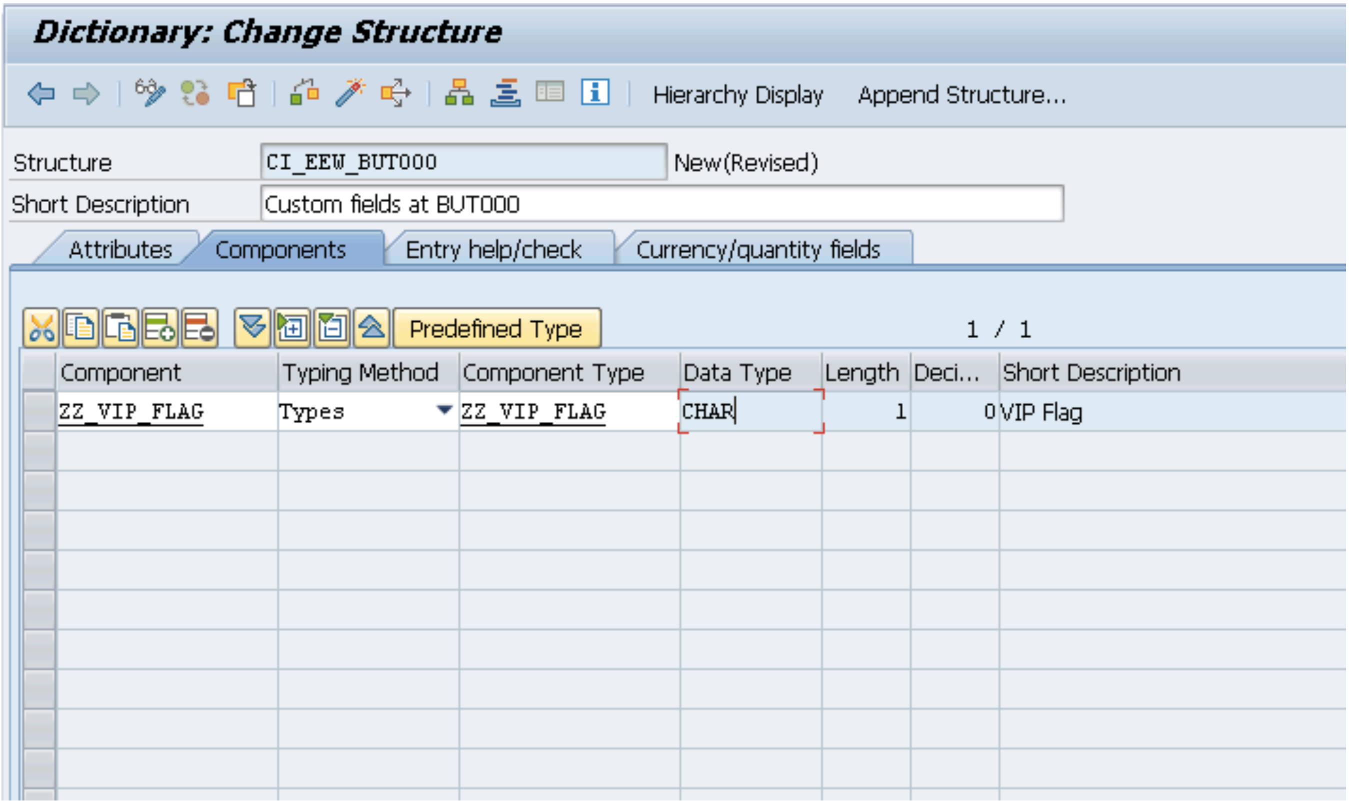
Figure 5: Create a new function group

Create the new field on the database (5mins)

If you're adding a new field to an existing SAP BDT driven object, like e.g. the business partner, you have to add the field to one of the existing SAP database tables, e.g. table BUT000 in case of the business partner (if your field belongs to the general data of the business partner). In this case, when you add fields to an existing BDT object, your BDT application is called the participating application and the SAP BDT application that owns the table is called the table owning application.

There are two ways of adding fields to SAP tables: table appends and customer includes. Table appends can be added to any transparent table via the SE11 -> GOTO -> append structure.

Customer includes are includes in the SAP table structure that are like place holders. They are delivered empty by SAP. By double clicking on the include name in the table display in SE11 you can create the include and populate it with your field. Save and activate the include. You have now a new empty field on the database in this table. The business partner table BUT000 e.g. has the following customer include: CI_EEW_BUT000. I activated it and added the custom field I will add to the BP dialogue: VIP flag[1].



Dictionary: Change Structure

Structure CI_EEW_BUT000 New(Revised)

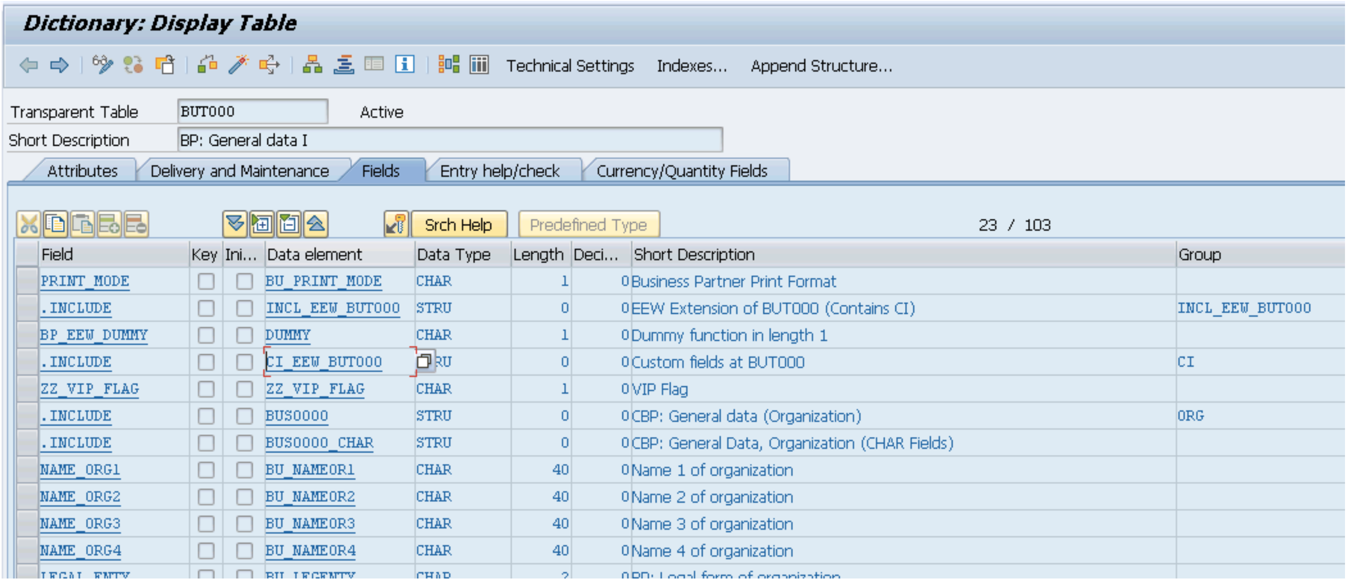
Short Description Custom fields at BUT000

1 / 1

Component	Typing Method	Component Type	Data Type	Length	Deci...	Short Description
ZZ_VIP_FLAG	Types	ZZ_VIP_FLAG	CHAR	1	0	VIP Flag

Figure 6: Adding the new field to a customer include

[1] If you're familiar with the SAP business partner, you'll know that there is already a SAP standard field for VIP functionality – excuse my lack of originality.



Transparent Table: BUT000 Active

Short Description: BP: General data I

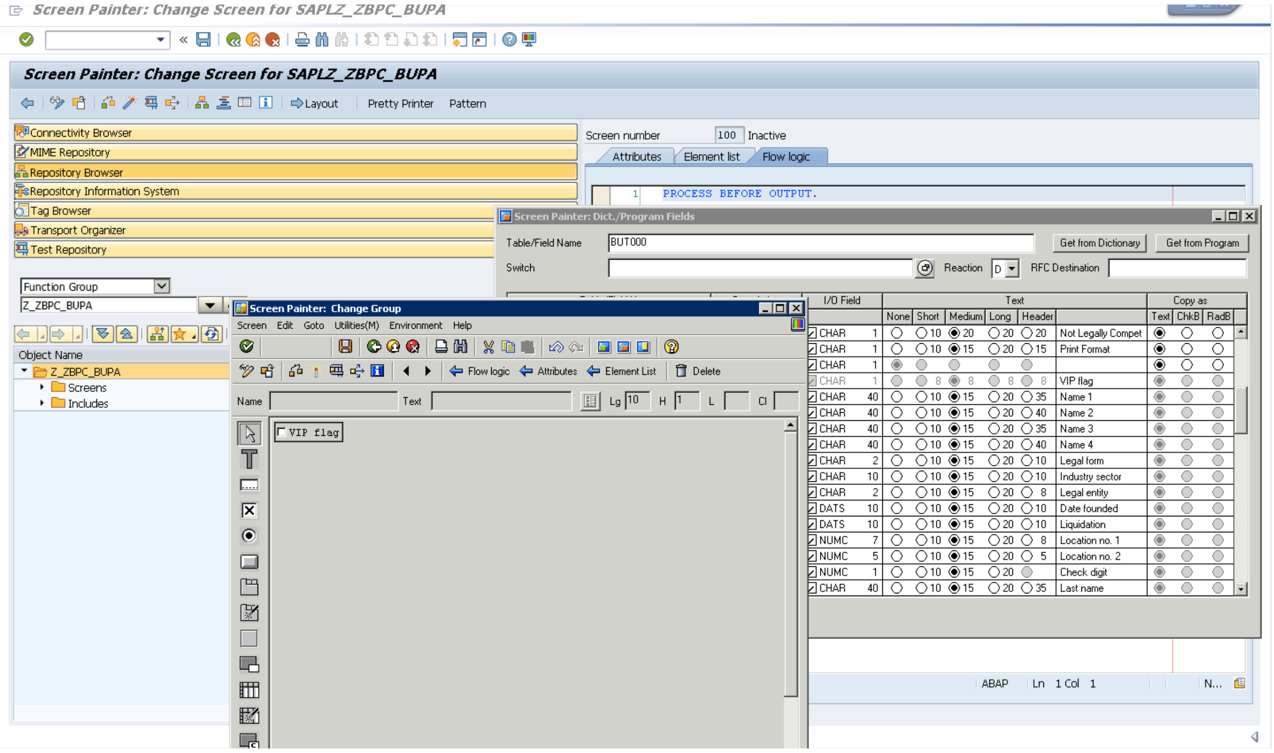
23 / 103

	Field	Key	Ini...	Data element	Data Type	Length	Deci...	Short Description	Group
	PRINT_MODE	□	□	BU_PRINT_MODE	CHAR	1	0	Business Partner Print Format	
	.INCLUDE	□	□	INCL_EEW_BUT000	STRU	0	0	EEW Extension of BUT000 (Contains CI)	INCL_EEW_BUT000
	BP_EEW_DUMMY	□	□	DUMMY	CHAR	1	0	Dummy function in length 1	
	.INCLUDE	□	□	CI_EEW_BUT000	RU	0	0	Custom fields at BUT000	CI
	ZZ_VIP_FLAG	□	□	ZZ_VIP_FLAG	CHAR	1	0	VIP Flag	
	.INCLUDE	□	□	BUS0000	STRU	0	0	CBP: General data (Organization)	ORG
	.INCLUDE	□	□	BUS0000_CHAR	STRU	0	0	CBP: General Data, Organization (CHAR Fields)	
	NAME_ORG1	□	□	BU_NAMEOR1	CHAR	40	0	Name 1 of organization	
	NAME_ORG2	□	□	BU_NAMEOR2	CHAR	40	0	Name 2 of organization	
	NAME_ORG3	□	□	BU_NAMEOR3	CHAR	40	0	Name 3 of organization	
	NAME_ORG4	□	□	BU_NAMEOR4	CHAR	40	0	Name 4 of organization	
	LEGAL_ENTY	□	□	BU_LEGENTY	CHAR	2	0	BP: Legal form of organization	

Figure 7: The new field is now included in the table BUT000 via the customer include

Create a new sub screen to hold your new field (5mins)

Go to your new function group and create a new sub screen. Place your new field with a DDIC reference on that sub screen. Do not place any logic in the PBO and PAI modules of your sub screen. Just call function module BUS_PBO in PBO and BUS_PAI in PAI module.



Screen number 100 Inactive

```
PROCESS BEFORE OUTPUT.
```

Table/Field Name BUT000

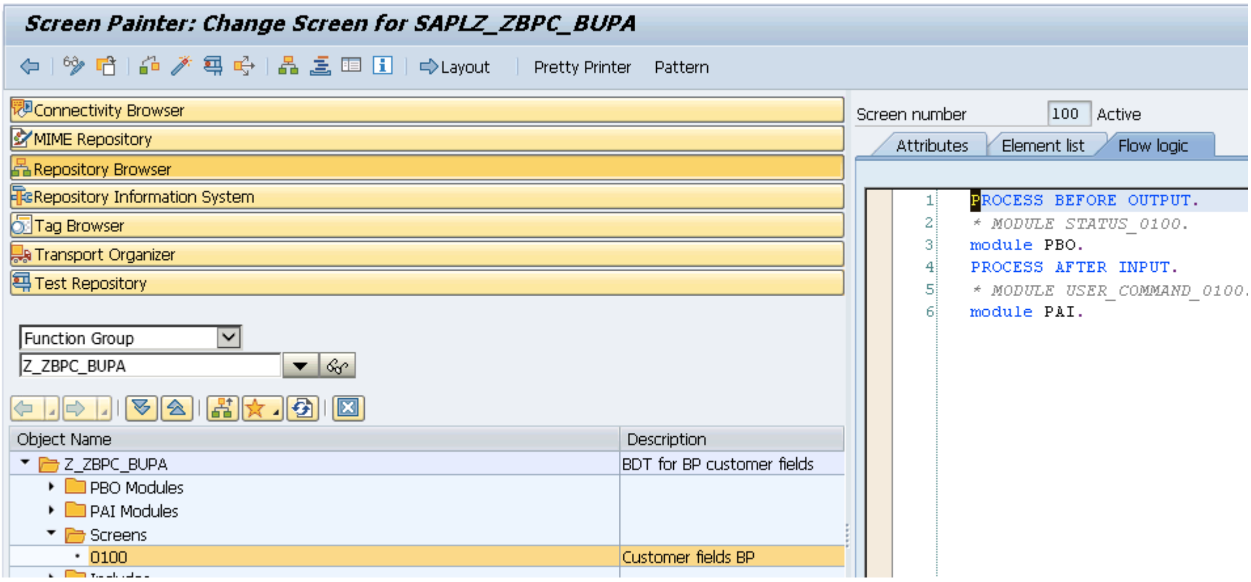
Reaction D

Lg 10 H 1

☐ VIP flag

I/O Field		None	Short	Medium	Long	Header	Text	Copy as Text	ChkB	RadB
CHAR	1	○	○ 10	● 20	○ 20	○ 20	Not Legally Compet	●	○	○
CHAR	1	○	○ 10	● 15	○ 20	○ 15	Print Format	●	○	○
CHAR	1	●	○	○	○	○		●	○	○
CHAR	1	○	○ 8	● 8	○ 8	○ 8	VIP flag	●	○	○
CHAR	40	○	○ 10	● 15	○ 20	○ 35	Name 1	●	○	○
CHAR	40	○	○ 10	● 15	○ 20	○ 40	Name 2	●	○	○
CHAR	40	○	○ 10	● 15	○ 20	○ 35	Name 3	●	○	○
CHAR	40	○	○ 10	● 15	○ 20	○ 40	Name 4	●	○	○
CHAR	2	○	○ 10	● 15	○ 20	○ 10	Legal form	●	○	○
CHAR	10	○	○ 10	● 15	○ 20	○ 10	Industry sector	●	○	○
CHAR	2	○	○ 10	● 15	○ 20	○ 8	Legal entity	●	○	○
DATS	10	○	○ 10	● 15	○ 20	○ 10	Date founded	●	○	○
DATS	10	○	○ 10	● 15	○ 20	○ 10	Liquidation	●	○	○
NUMC	7	○	○ 10	● 15	○ 20	○ 8	Location no. 1	●	○	○
NUMC	5	○	○ 10	● 15	○ 20	○ 5	Location no. 2	●	○	○
NUMC	1	○	○ 10	● 15	○ 20	○	Check digit	●	○	○
CHAR	40	○	○ 10	● 15	○ 20	○ 35	Last name	●	○	○

Figure 8: Adding the new field to your sub screen



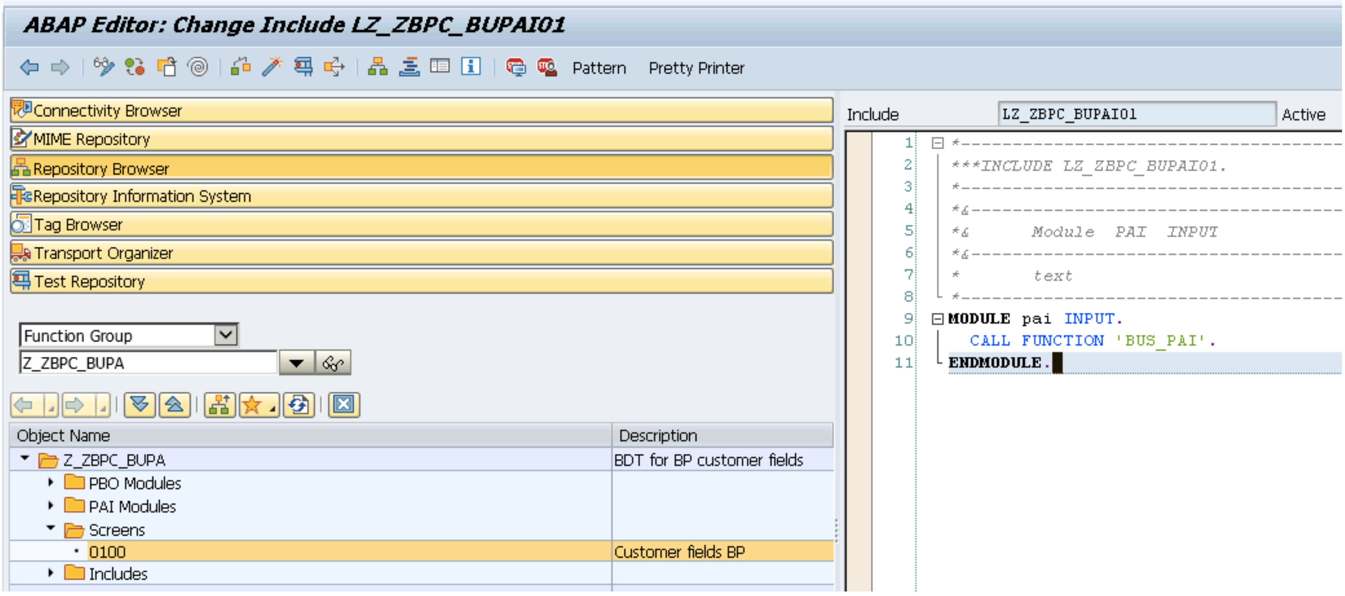
Screen number 100 Active

Z_ZBPC_BUPA

Object Name	Description
Z_ZBPC_BUPA	BDT for BP customer fields
PBO Modules	
PAI Modules	
Screens	
0100	Customer fields BP

```
PROCESS BEFORE OUTPUT.
* MODULE STATUS_0100.
module PBO.
PROCESS AFTER INPUT.
* MODULE USER_COMMAND_0100.
module PAI.
```

Figure 9: Adding PBO and PAI modules to the sub screen



Function Group
Z_ZBPC_BUPA

Object Name	Description
Z_ZBPC_BUPA	BDT for BP customer fields
PBO Modules	
PAI Modules	
Screens	
0100	Customer fields BP
Includes	

Include LZ_ZBPC_BUPAI01 Active

```
*---------------------------------------------------------------------
***INCLUDE LZ_ZBPC_BUPAI01.
*---------------------------------------------------------------------
*&--------------------------------------------------------------------
*&      Module  PAI  INPUT
*&--------------------------------------------------------------------
*       text
*---------------------------------------------------------------------
MODULE pai INPUT.
  CALL FUNCTION 'BUS_PAI'.
ENDMODULE.
```

Figure 10: Calling function module BUS_PAI in PAI module

Set up the screen layout (5mins)

The BDT screen layout defines how and where fields appear on SAP screens. The screen layout elements are hierarchically ordered: A field is assigned to a field group, which is assigned to a view. Views are assigned to sections, which are assigned to screens. Screens are assigned to screen sequences.

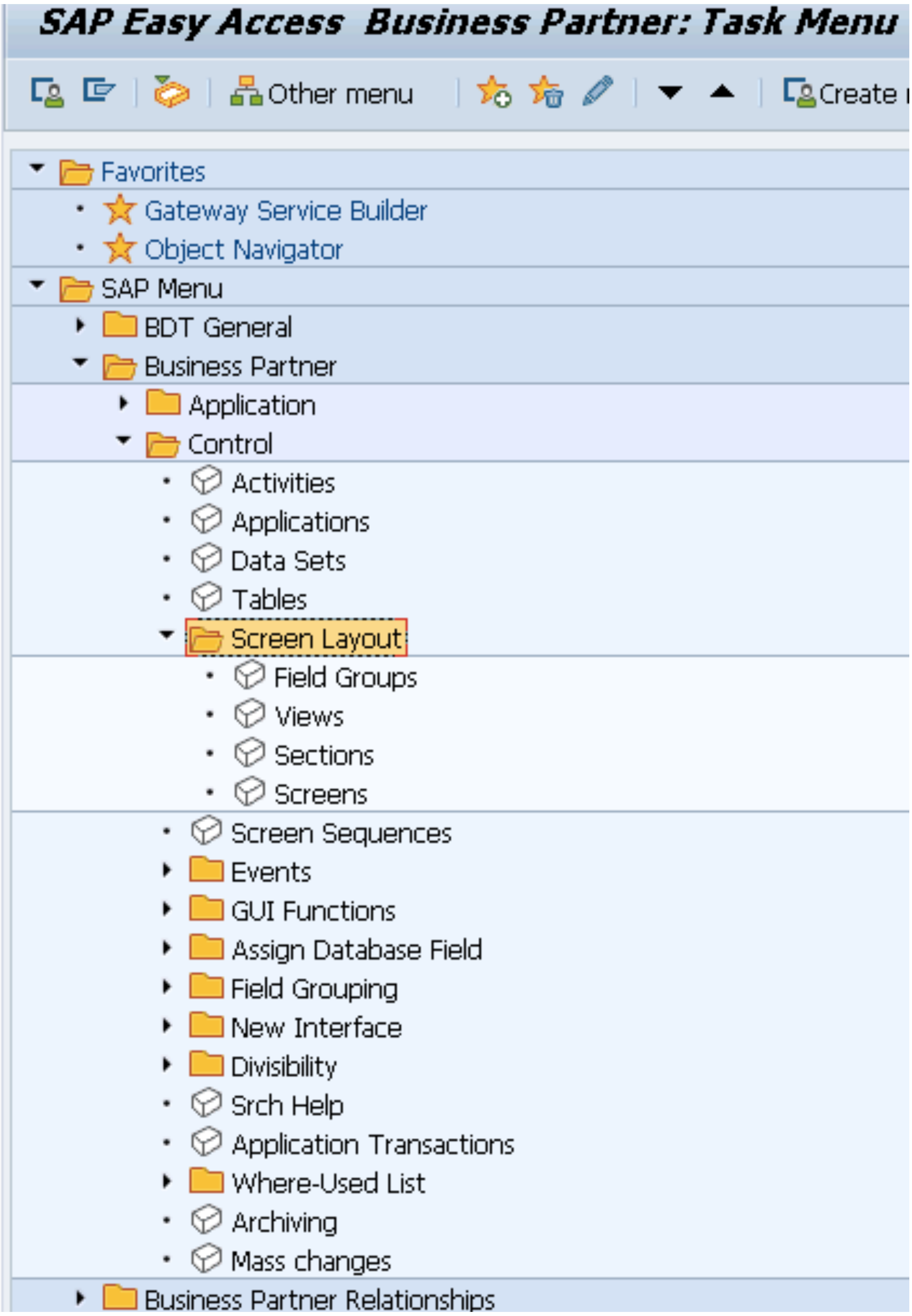
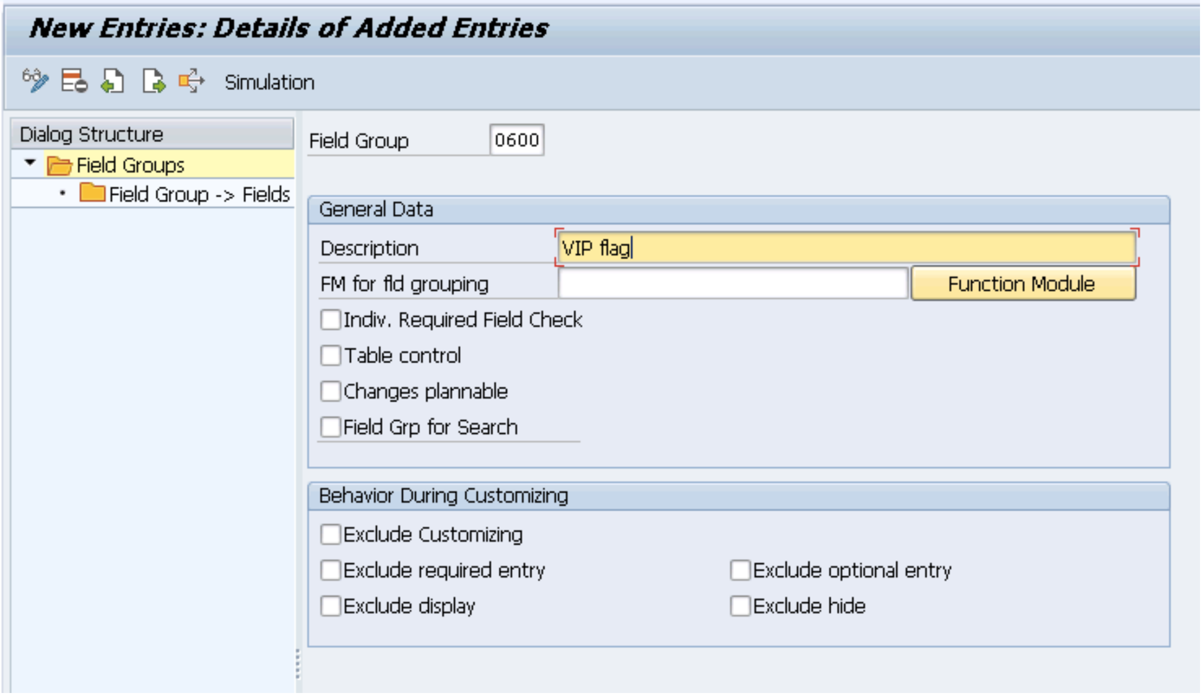


Figure 11: Path to the screen layout settings

Field group

Go to your BDT area menu and create a new field group in Control -> Screen Layout -> Field Groups. The customer name space for field groups is 600-749. Assign your field to your field group.



New Entries: Details of Added Entries

Simulation

Dialog Structure
- Field Groups
  - Field Group -> Fields

Field Group: 0600

General Data

Description: VIP flag

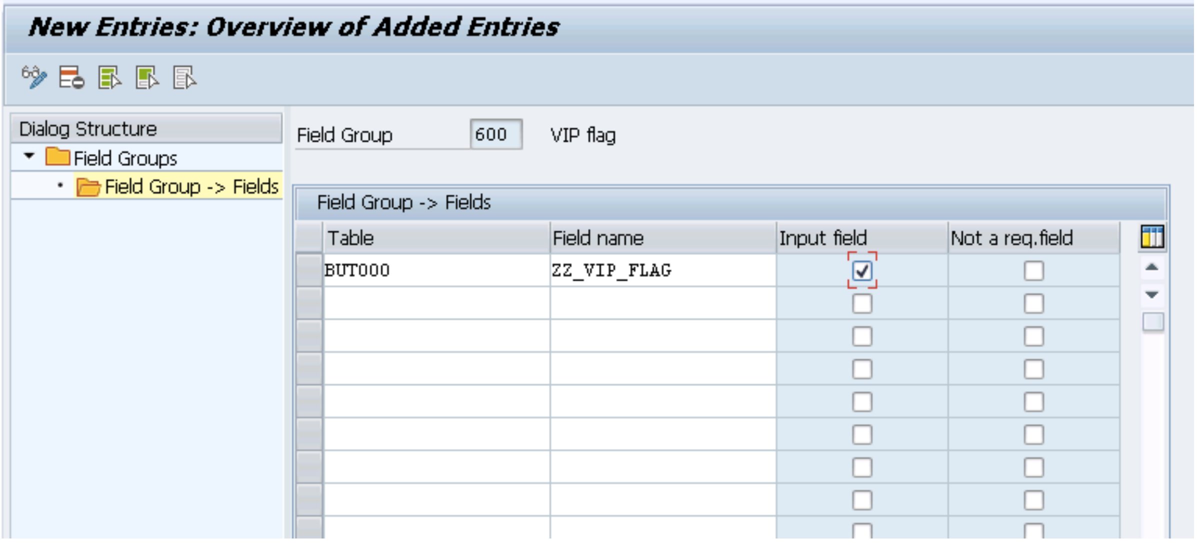
FM for fld grouping:

□ Indiv. Required Field Check
□ Table control
□ Changes plannable
□ Field Grp for Search

Behavior During Customizing

□ Exclude Customizing
□ Exclude required entry
□ Exclude optional entry
□ Exclude display
□ Exclude hide

Figure 12: Creating a new field group



New Entries: Overview of Added Entries

Dialog Structure
- Field Groups
  - Field Group -> Fields

Field Group: 600 VIP flag

Field Group -> Fields

Table	Field name	Input field	Not a req.field
BUT000	ZZ_VIP_FLAG	☑	□
		□	□
		□	□
		□	□
		□	□
		□	□
		□	□
		□	□

Figure 13: Assigning the field to the field group

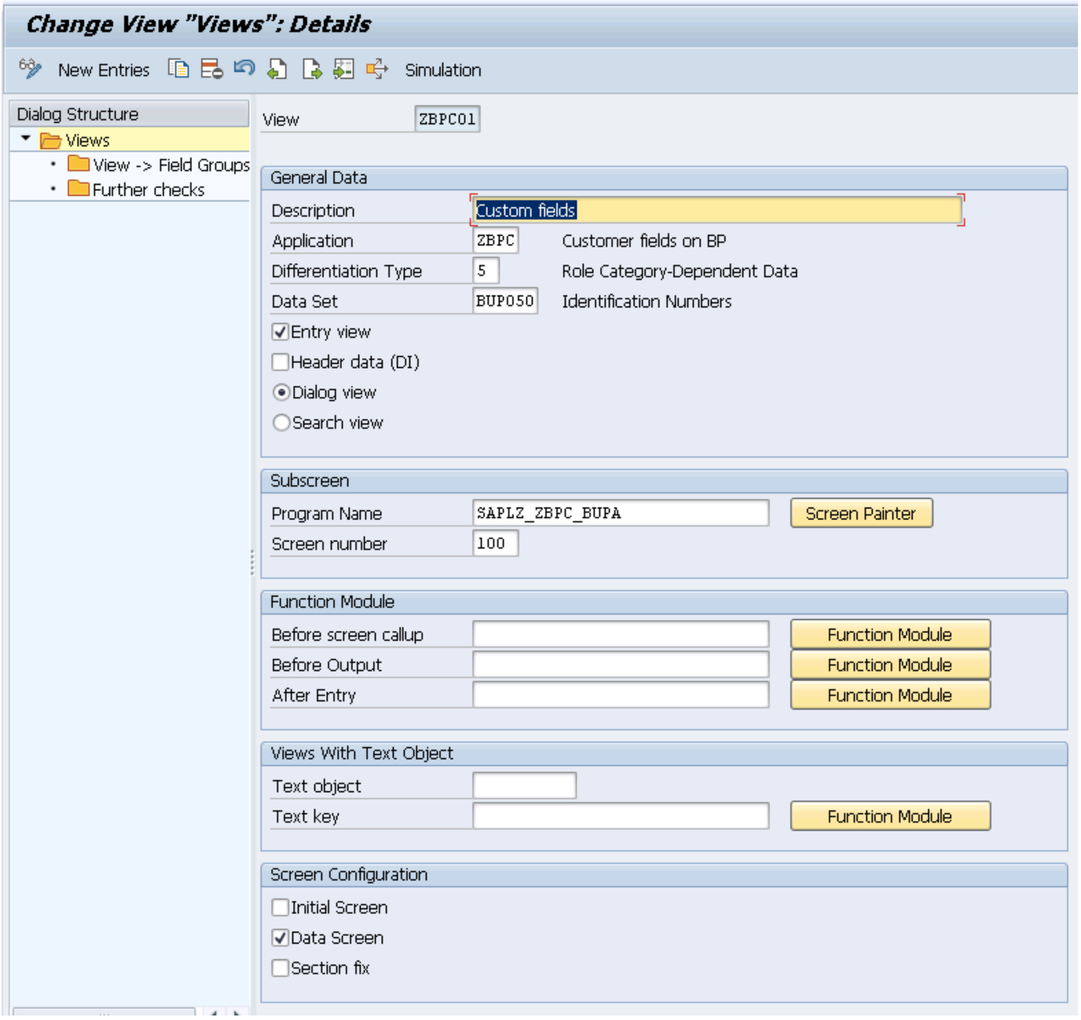
Be aware that there will be warning messages telling you "field does not exist on the assigned subscreen". You can ignore those messages by hitting the return key.

View

In the next step we create a new view. The naming convention is <appl>n with n being a number.

In the general data section add a description and your BDT application. If the application object supports divisibility, like the Business Partner does, you will need to enter a differentiation type. Choose the differentiation type your object uses. Within the Business Partner it's 5 for Role Category Dependent Data. If your object doesn't support divisibility then choose 0 for General Data. You can furthermore add the view to a data set. Data sets are designed to bundle attributes together. In case of divisibility you assign each object part (role in case of the Business Partner) a set of data sets. For my example I choose to add the new field to the Identification data set.

In the frame sub screen you have to enter your sub screens number and the program name of your function group (which is usually SAPL<functiongroupname>).



Change View "Views": Details

View: ZBPC01

General Data

Description: Custom fields
Application: ZBPC Customer fields on BP
Differentiation Type: 5 Role Category-Dependent Data
Data Set: BUP050 Identification Numbers
☑ Entry view
☐ Header data (DI)
● Dialog view
○ Search view

Subscreen

Program Name: SAPLZ_ZBPC_BUPA
Screen number: 100

Function Module

Before screen callup:
Before Output:
After Entry:

Views With Text Object

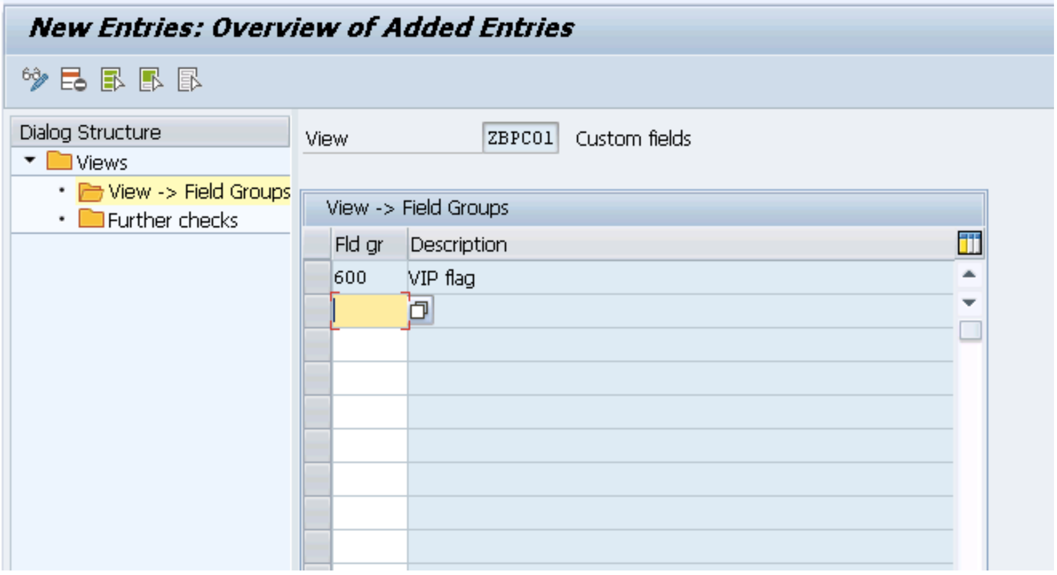
Text object:
Text key:

Screen Configuration

☐ Initial Screen
☑ Data Screen
☐ Section fix

Figure 14: Creating a new view

Assign your field group to that view.



New Entries: Overview of Added Entries

View: ZBPC01 Custom fields

View -> Field Groups

Fld gr	Description
600	VIP flag

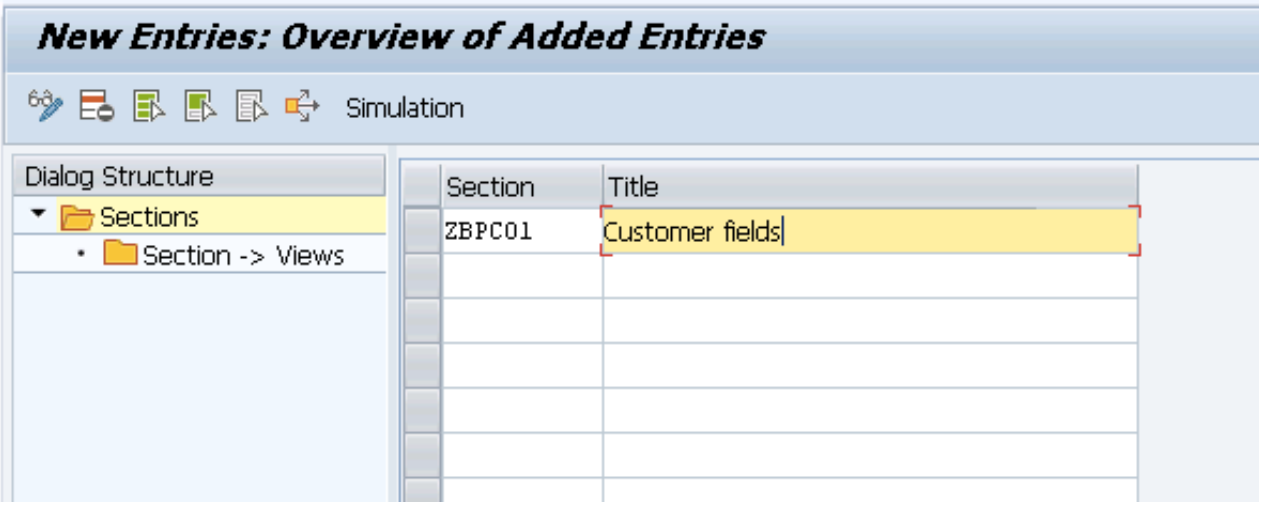
Figure 15: Assign the field group to the view

Depending on where you want your new field to appear, you assign the view to a section of SAP or create your own section.

If you created a new section you must assign it to a screen. Choose an existing SAP screen for simple display of your field or create your own screen if you want it to appear e.g. on a new tab on a SAP tab strip control.

Section

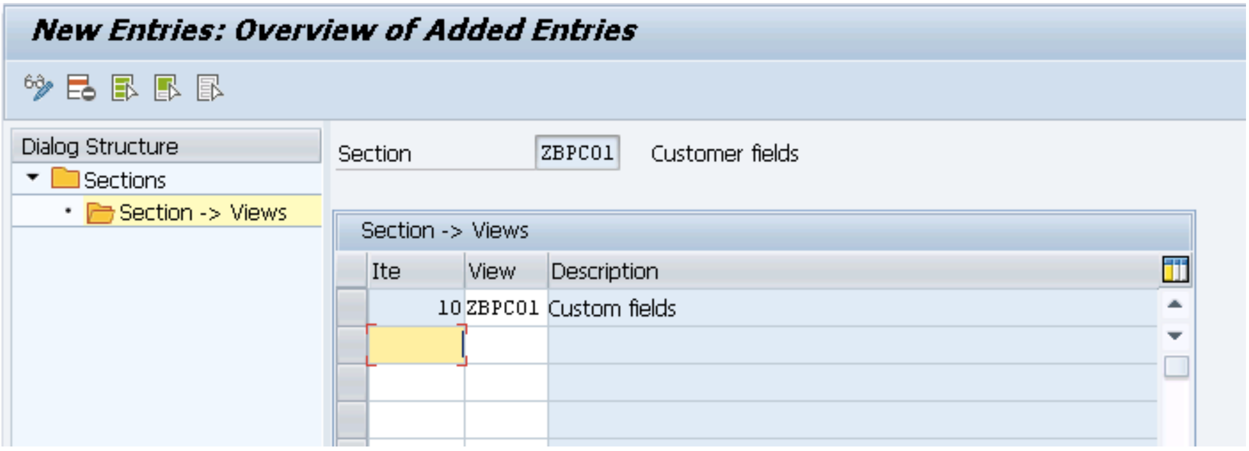
In my example I chose to create a new section that I'm going to add to an existing SAP screen.



New Entries: Overview of Added Entries

Section	Title
ZBPC01	Customer fields

Figure 16: Creating a new section



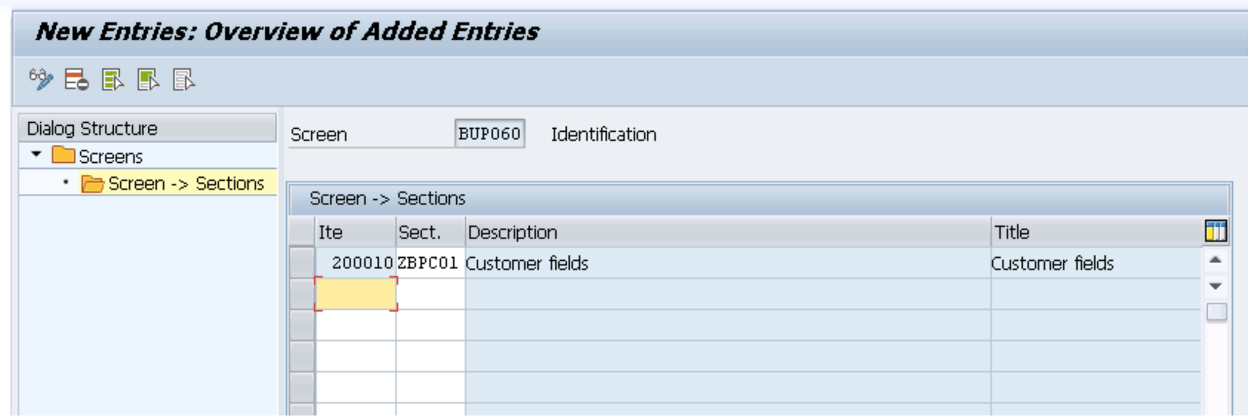
New Entries: Overview of Added Entries

Section ZBPC01 Customer fields

Section -> Views

Ite	View	Description
10	ZBPC01	Custom fields

Figure 17: Assigning the view to the new section



New Entries: Overview of Added Entries

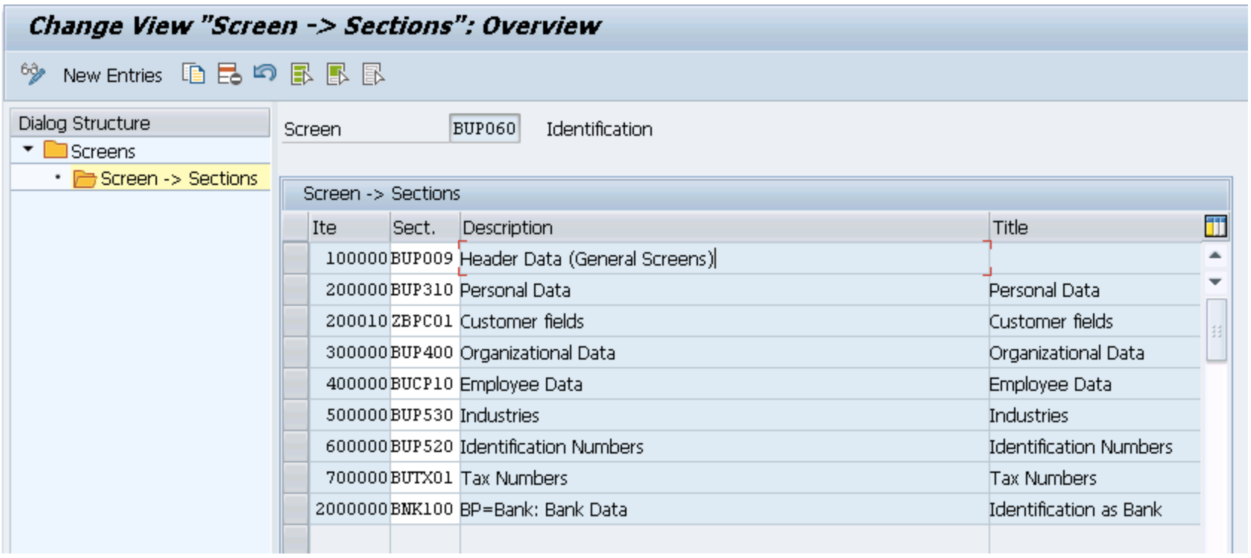
Screen BUP060 Identification

Screen -> Sections

Ite	Sect.	Description	Title
200010	ZBPC01	Customer fields	Customer fields

Figure 18: Assigning the new section to an existing SAP screen

I'm assigning my new section to the SAP screen BUP060 that represents the Identification tab on the Business Partner dialogue.



Change View "Screen -> Sections": Overview

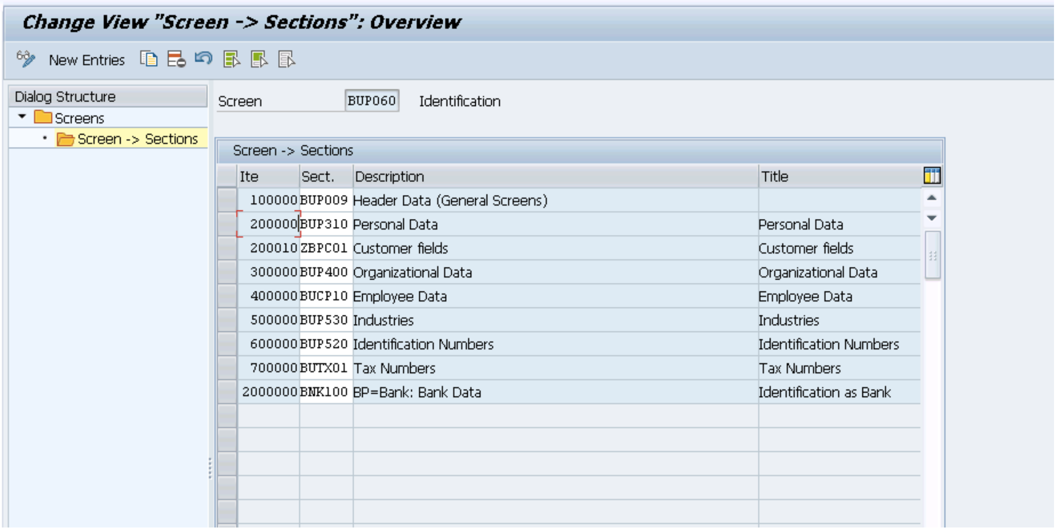
Screen BUP060 Identification

Screen -> Sections

Ite	Sect.	Description	Title
100000	BUP009	Header Data (General Screens)	
200000	BUP310	Personal Data	Personal Data
200010	ZBPC01	Customer fields	Customer fields
300000	BUP400	Organizational Data	Organizational Data
400000	BUCP10	Employee Data	Employee Data
500000	BUP530	Industries	Industries
600000	BUP520	Identification Numbers	Identification Numbers
700000	BUTX01	Tax Numbers	Tax Numbers
2000000	BNK100	BP=Bank: Bank Data	Identification as Bank

Figure 19: The new section appears between the SAP sections

Note the naming convention for the item numbers: the last two digits are the customer name space. If you want your section to appear after SAPs section 200000 you can name it 2000xy with xy[2] being a number of your choice. By choosing 200010 we will have the new section appear between the Personal Data and Organizational Data.



Change View "Screen -> Sections": Overview

Screen BUP060 Identification

Screen -> Sections

Ite	Sect.	Description	Title
100000	BUP009	Header Data (General Screens)	
200000	BUP310	Personal Data	Personal Data
200010	ZBPC01	Customer fields	Customer fields
300000	BUP400	Organizational Data	Organizational Data
400000	BUCP10	Employee Data	Employee Data
500000	BUP530	Industries	Industries
600000	BUP520	Identification Numbers	Identification Numbers
700000	BUTX01	Tax Numbers	Tax Numbers
2000000	BNK100	BP=Bank: Bank Data	Identification as Bank

Figure 20: The new section appears on the SAP screen

If your BDT object supports divisibility, you'll have to assign your new BDT application to one of the objects parts. Within the Business Partner those object parts are roles and are also called BP views. In my BP example I chose to assign the BDT application to the General Role (000000):

[2] X is actually supposed to be for the customers central development (like repository name space Z) and Y is supposed to be for customers branch development (like Y repository name space)

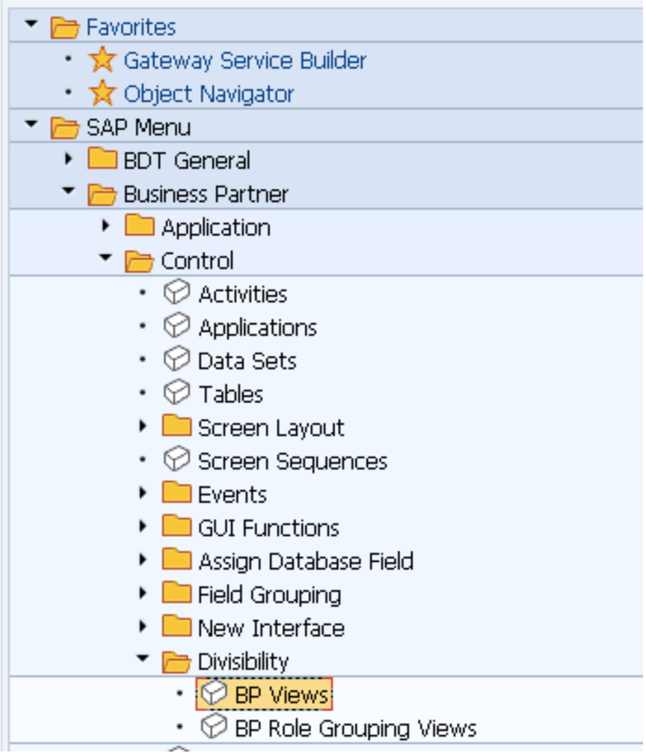
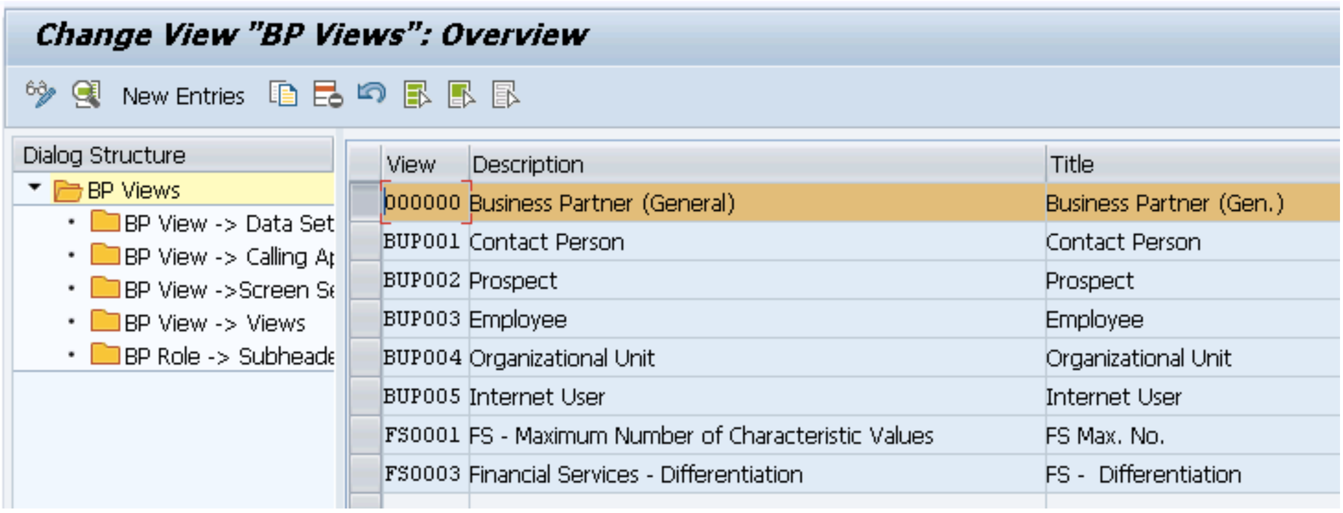
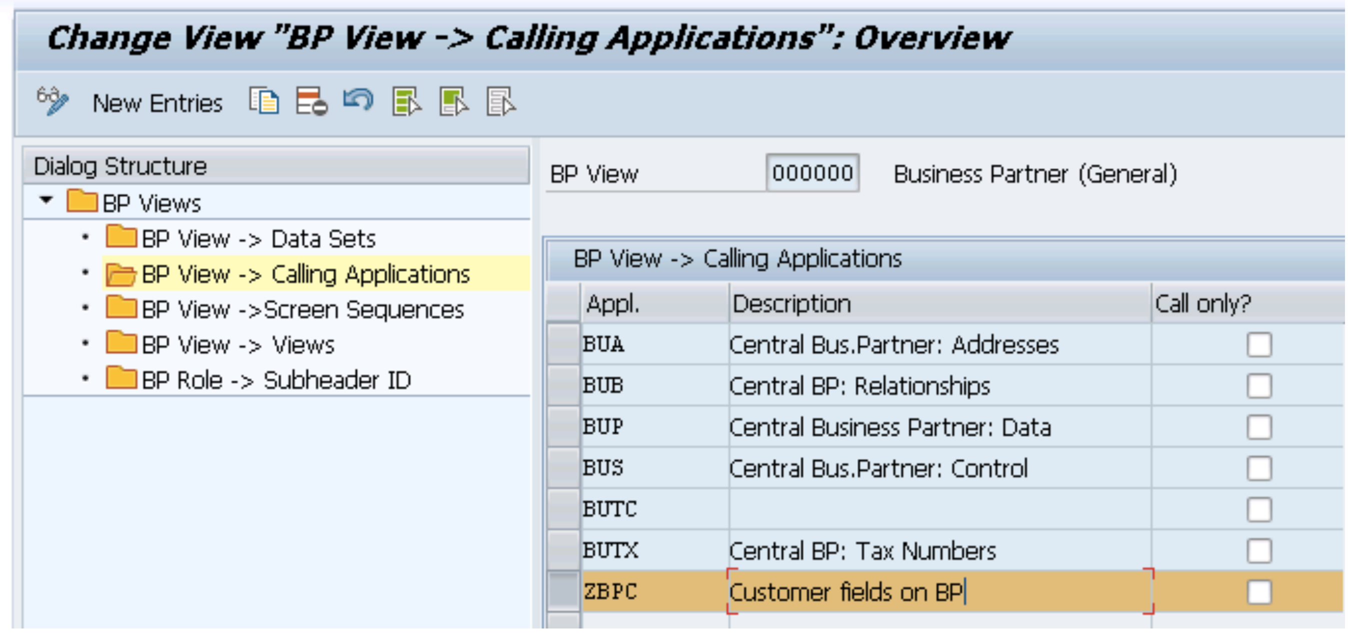


Figure 21: BDT Entry for setting the divisibility choices


Change View "BP Views": Overview

View	Description	Title
000000	Business Partner (General)	Business Partner (Gen.)
BUP001	Contact Person	Contact Person
BUP002	Prospect	Prospect
BUP003	Employee	Employee
BUP004	Organizational Unit	Organizational Unit
BUP005	Internet User	Internet User
FS0001	FS - Maximum Number of Characteristic Values	FS Max. No.
FS0003	Financial Services - Differentiation	FS - Differentiation

Figure 22: Choosing role 000000 (general)



Change View "BP View -> Calling Applications": Overview

Dialog Structure
- BP Views
  - BP View -> Data Sets
  - BP View -> Calling Applications
  - BP View ->Screen Sequences
  - BP View -> Views
  - BP Role -> Subheader ID

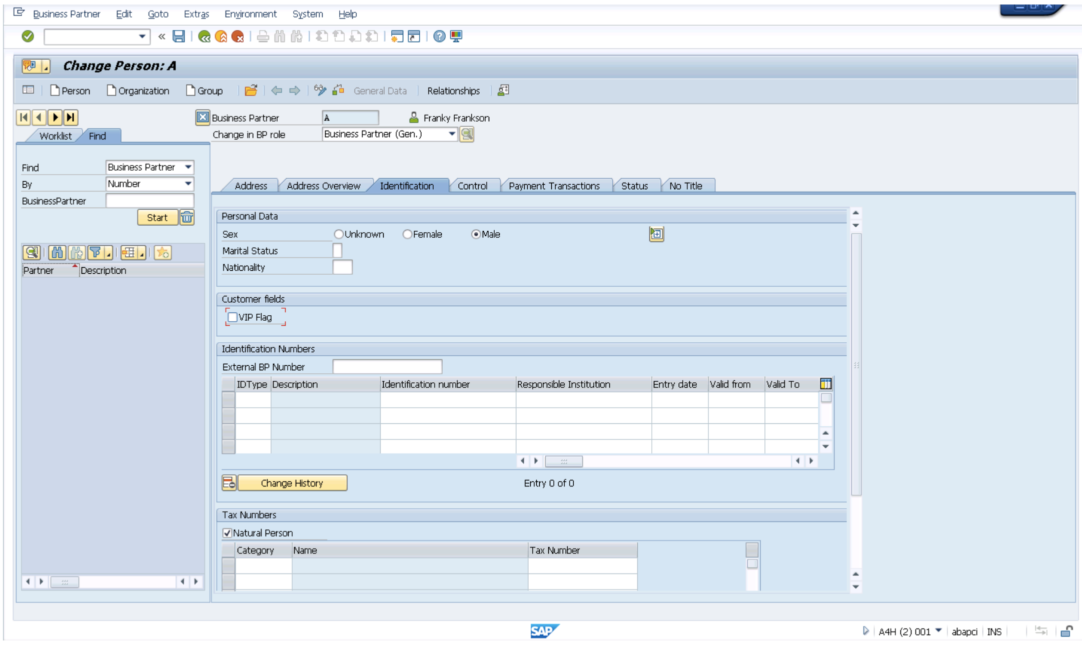
BP View 000000 Business Partner (General)

BP View -> Calling Applications

Appl.	Description	Call only?
BUA	Central Bus.Partner: Addresses	□
BUB	Central BP: Relationships	□
BUP	Central Business Partner: Data	□
BUS	Central Bus.Partner: Control	□
BUTC		□
BUTX	Central BP: Tax Numbers	□
ZBPC	Customer fields on BP	□

Figure 23: Assigning our new BDT application to be called within role 000000

Now it’s time to check that everything is set up correctly. Start your maintenance transaction, e.g. BP for Business Partner, and you should see the new field:



Change Person: A

Business Partner: A — Franky Frankson
Change in BP role: Business Partner (Gen.)

Find: Business Partner
By: Number
BusinessPartner:

Partner	Description

Personal Data
Sex: ○ Unknown ○ Female ● Male
Marital Status:
Nationality:

Customer fields
□ VIP Flag

Identification Numbers
External BP Number:

IDType	Description	Identification number	Responsible Institution	Entry date	Valid from	Valid To

Entry 0 of 0

Tax Numbers
☑ Natural Person

Category	Name	Tax Number

A4H (2) 001 | abapci | INS

Figure 24: The new field shows up on the screen

If you cannot see the new field, check all previous steps again. If you still can't see it, try to generate the BDT sub screen containers:

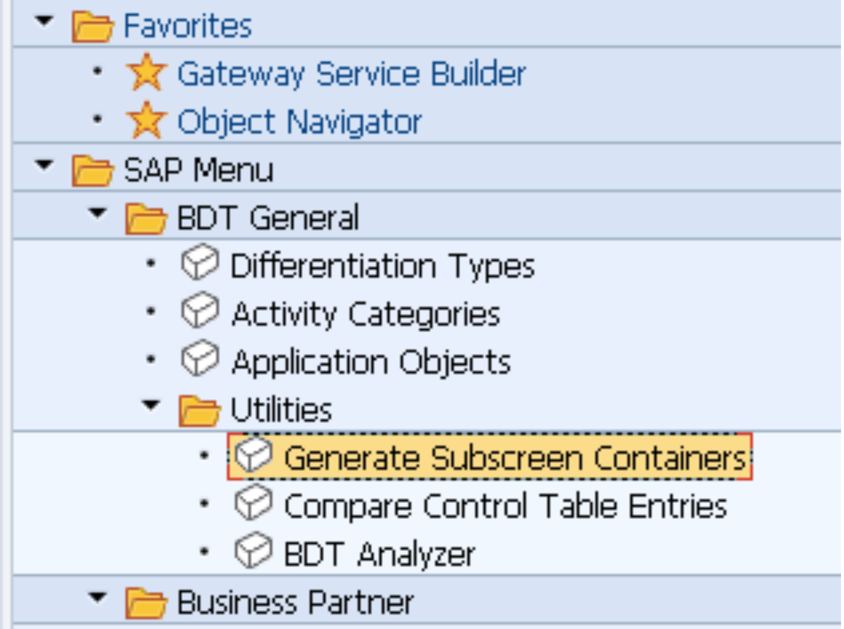
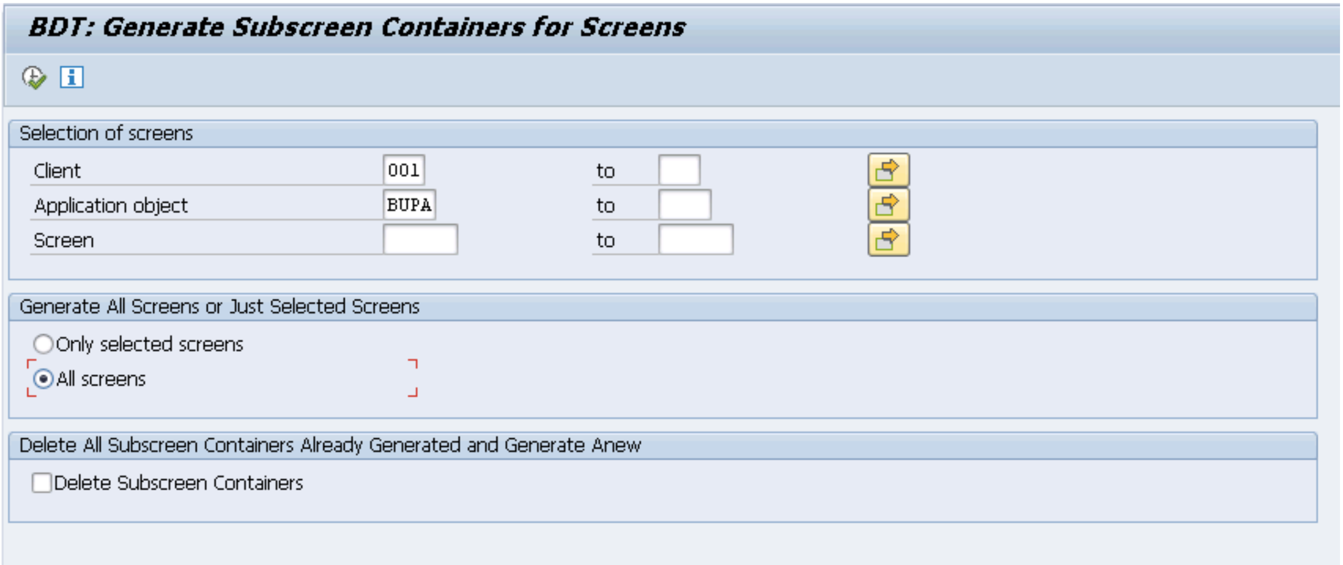


Figure 25: Path to find the Generate Subscreen Container transaction


BDT: Generate Subscreen Containers for Screens

Selection of screens

Client	001	to	
Application object	BUPA	to	
Screen		to	

Generate All Screens or Just Selected Screens

○ Only selected screens
● All screens

Delete All Subscreen Containers Already Generated and Generate Anew

□ Delete Subscreen Containers

Figure 26: Parameters to generate the sub screen containers

Select your application object and the scope to "all screens". After that you should see the field.

When the field is displayed there won't be any functionality for saving the values you entered or displaying the data from the database. Therefore we have to implement the BDT logic for the new field in the next step.

Set up the BDT logic for your new application (10mins)

The table owning application keeps the old and updated values of the application in its function group memory during run time. To add your field value you must get the content of that memory and later at saving time you must pass back your new field value to the table owning application.

The owning application supplies a set of function modules for this purpose. You can find them in your BDT area menu under Control -> Tables. You'll need the Read and the Collect function modules for the table you added your field to:

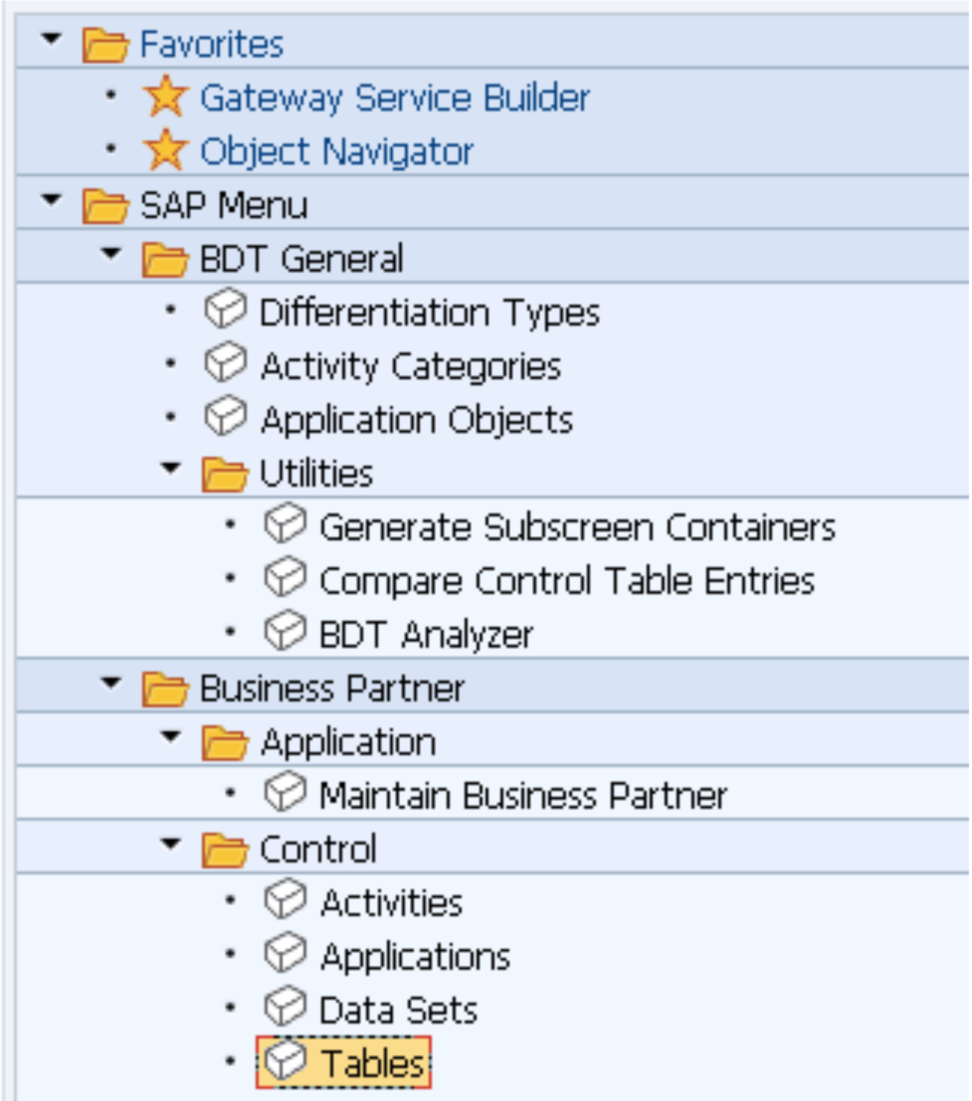
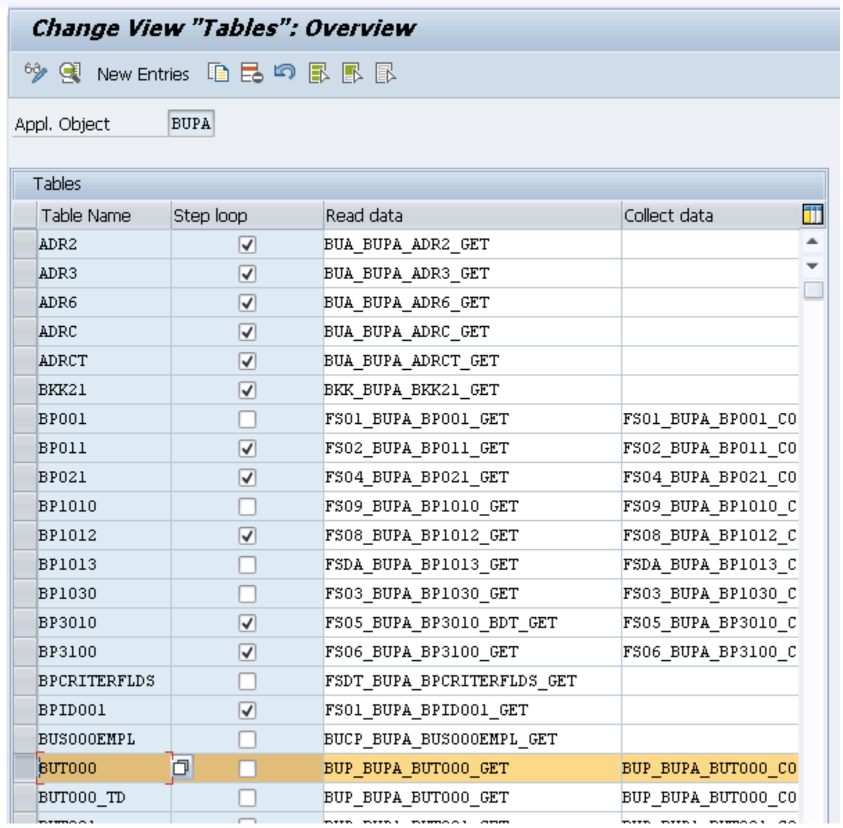


Figure 27: Path in the BDT area menu to the Tables entry



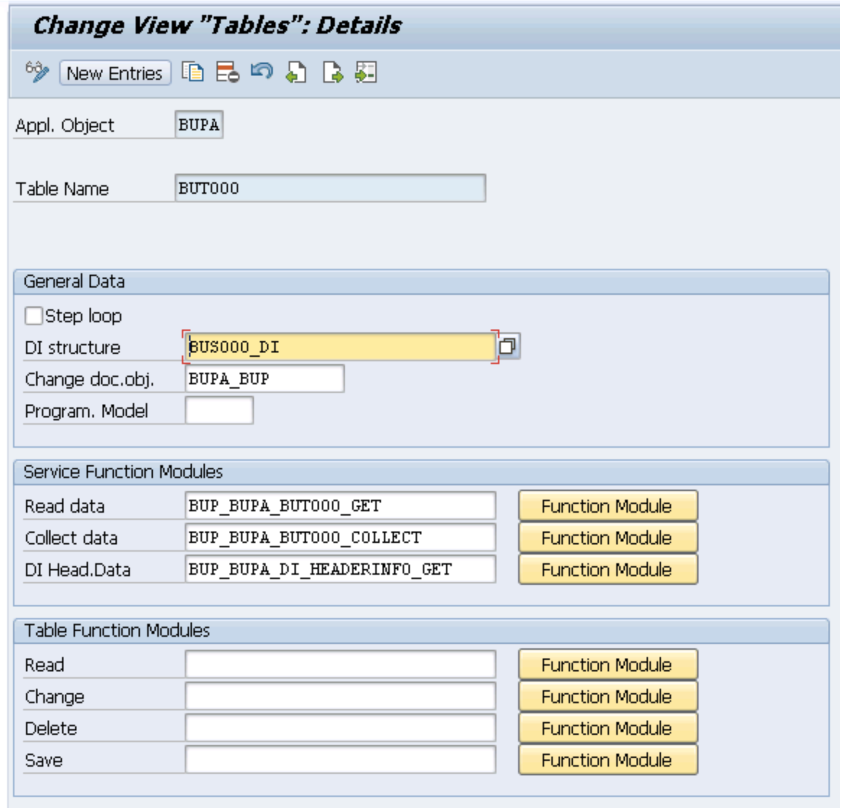
Change View "Tables": Overview

Appl. Object BUPA

Tables

Table Name	Step loop	Read data	Collect data
ADR2	☑	BUA_BUPA_ADR2_GET	
ADR3	☑	BUA_BUPA_ADR3_GET	
ADR6	☑	BUA_BUPA_ADR6_GET	
ADRC	☑	BUA_BUPA_ADRC_GET	
ADRCT	☑	BUA_BUPA_ADRCT_GET	
BKK21	☑	BKK_BUPA_BKK21_GET	
BP001	□	FS01_BUPA_BP001_GET	FS01_BUPA_BP001_CO
BP011	☑	FS02_BUPA_BP011_GET	FS02_BUPA_BP011_CO
BP021	☑	FS04_BUPA_BP021_GET	FS04_BUPA_BP021_CO
BP1010	□	FS09_BUPA_BP1010_GET	FS09_BUPA_BP1010_C
BP1012	☑	FS08_BUPA_BP1012_GET	FS08_BUPA_BP1012_C
BP1013	□	FSDA_BUPA_BP1013_GET	FSDA_BUPA_BP1013_C
BP1030	□	FS03_BUPA_BP1030_GET	FS03_BUPA_BP1030_C
BP3010	☑	FS05_BUPA_BP3010_BDT_GET	FS05_BUPA_BP3010_C
BP3100	☑	FS06_BUPA_BP3100_GET	FS06_BUPA_BP3100_C
BPCRITERFLDS	□	FSDT_BUPA_BPCRITERFLDS_GET	
BPID001	☑	FS01_BUPA_BPID001_GET	
BUS000EMPL	□	BUCP_BUPA_BUS000EMPL_GET	
BUT000	□	BUP_BUPA_BUT000_GET	BUP_BUPA_BUT000_CO
BUT000_TD	□	BUP_BUPA_BUT000_GET	BUP_BUPA_BUT000_CO

Figure 28: Choose the table you're adding the field to



Change View "Tables": Details

Appl. Object BUPA

Table Name BUT000

General Data

□ Step loop

DI structure	BUS000_DI
Change doc.obj.	BUPA_BUP
Program. Model	

Service Function Modules

Read data	BUP_BUPA_BUT000_GET	Function Module
Collect data	BUP_BUPA_BUT000_COLLECT	Function Module
DI Head.Data	BUP_BUPA_DI_HEADERINFO_GET	Function Module

Table Function Modules

Read		Function Module
Change		Function Module
Delete		Function Module
Save		Function Module

Figure 29: Service function modules of the Business Partner for table BUT000

In your BDT area menu there is a BDT event section: Control -> Events -> BDT Events.

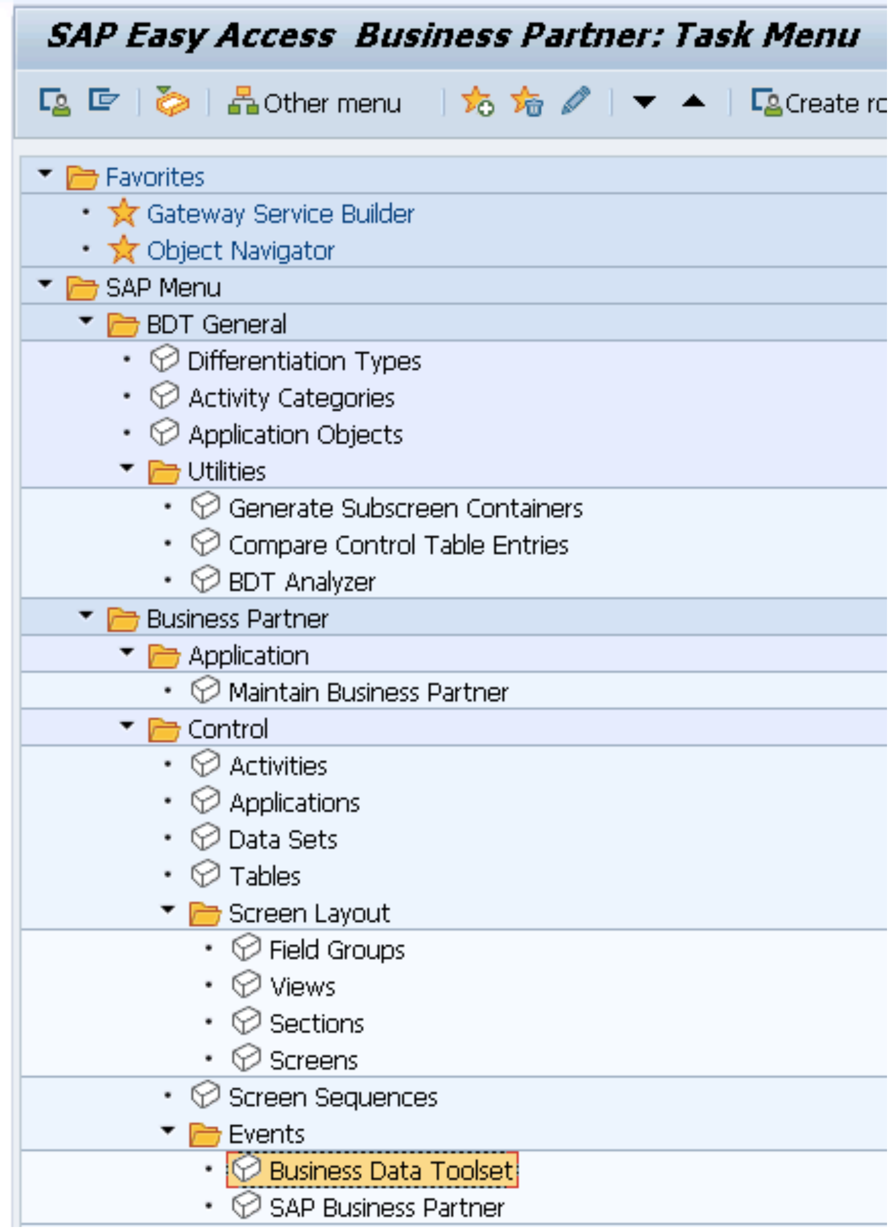


Figure 30: BDT events in the area menu

You must implement the events ISDST, XCHNG, DLVE1 and DSAVB.
See flowcharts below that show the sequence of all BDT events in dialogue mode[3].

[3] The sequence of the BDT events in Direct Input differs slightly from this, but is not part of this How-To

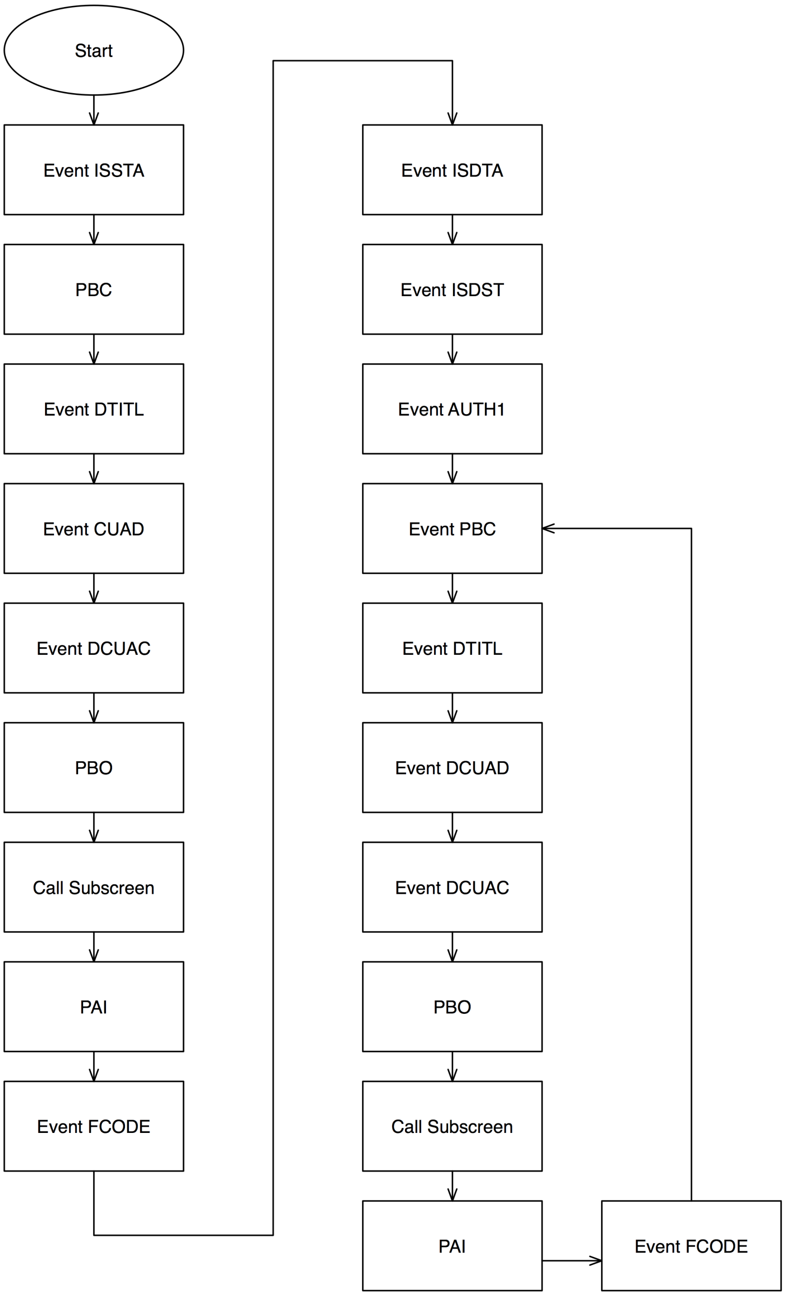


Figure 31: BDT event flow during data entry

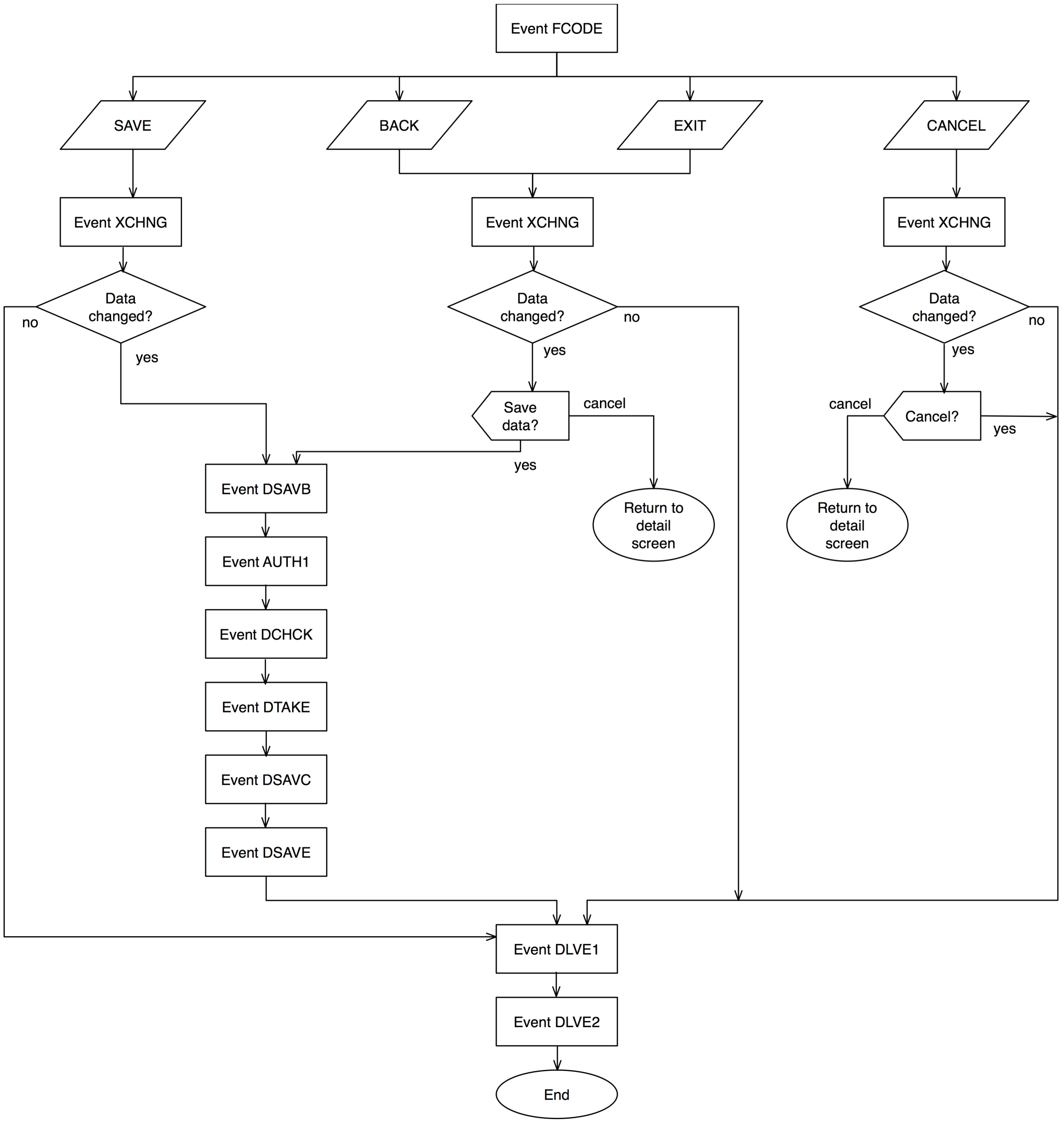
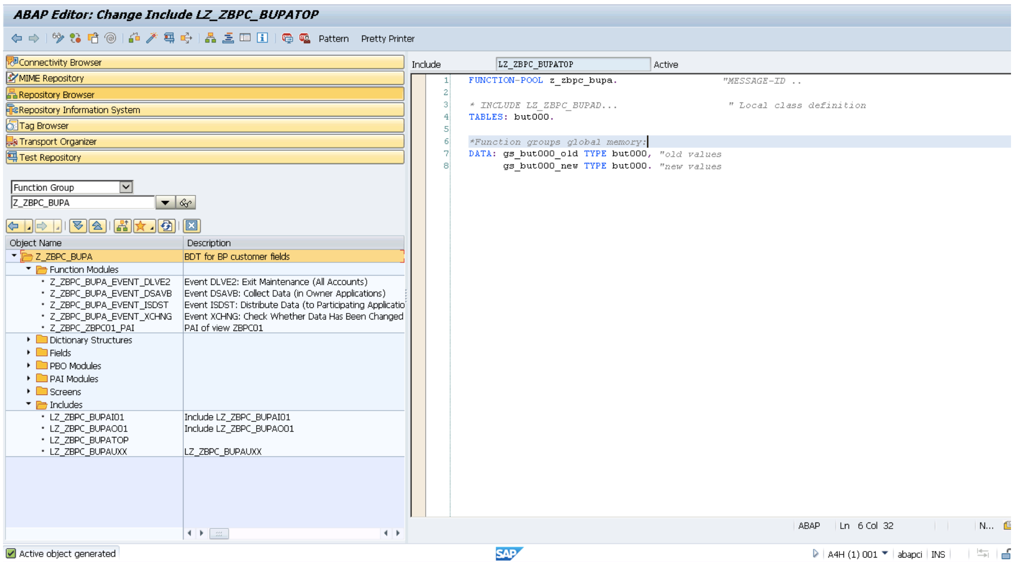


Figure 32: BDT at saving/leaving the dialogue

Global Memory

Create the global memory for the table you're enhancing to keep old and new values of your object in your function group's top include:



ABAP Editor: Change Include LZ_ZBPC_BUPATOP

Function Group: Z_ZBPC_BUPA

Object Name	Description
Z_ZBPC_BUPA	BDT for BP customer fields
Function Modules	
Z_ZBPC_BUPA_EVENT_DLVE2	Event DLVE2: Exit Maintenance (All Accounts)
Z_ZBPC_BUPA_EVENT_DSAVB	Event DSAVB: Collect Data (in Owner Applications)
Z_ZBPC_BUPA_EVENT_ISDST	Event ISDST: Distribute Data (to Participating Applicatio
Z_ZBPC_BUPA_EVENT_XCHNG	Event XCHNG: Check Whether Data Has Been Changed
Z_ZBPC_ZBPC01_PAI	PAI of view ZBPC01
Dictionary Structures	
Fields	
PBO Modules	
PAI Modules	
Screens	
Includes	
LZ_ZBPC_BUPAI01	Include LZ_ZBPC_BUPAI01
LZ_ZBPC_BUPAO01	Include LZ_ZBPC_BUPAO01
LZ_ZBPC_BUPATOP	
LZ_ZBPC_BUPAUXX	LZ_ZBPC_BUPAUXX

Include LZ_ZBPC_BUPATOP Active

```
FUNCTION-POOL z_zbpc_bupa.                       "MESSAGE-ID ..

* INCLUDE LZ_ZBPC_BUPAD...                        " Local class definition
TABLES: but000.

*Function groups global memory:
DATA: gs_but000_old TYPE but000, "old values
      gs_but000_new TYPE but000. "new values
```

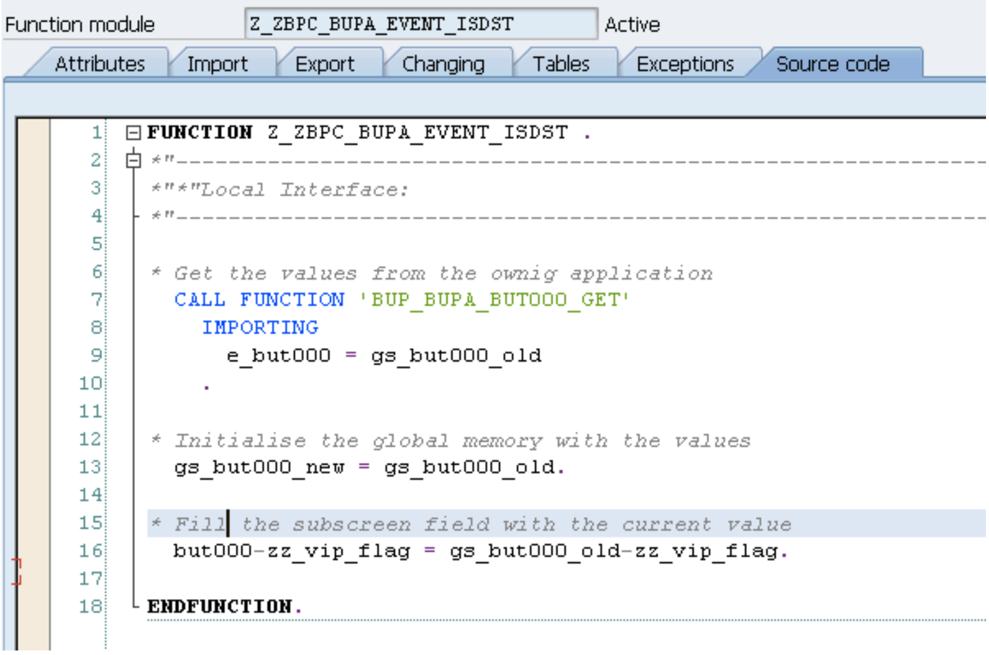
Active object generated

Figure 33: Creating the global memory for the old and new object values

Event ISDST (Getting the data from the owning application)

Create a function module for the event ISDST. Naming convention: Z_<appl>_<obj>_EVENT_ISDST. In event ISDST you have to get the current memory of the owning application. You use the Read function module you determined before. Store the data in your own function group's global memory. This function module has no parameters.

This is what ISDST looks like for my BP example:



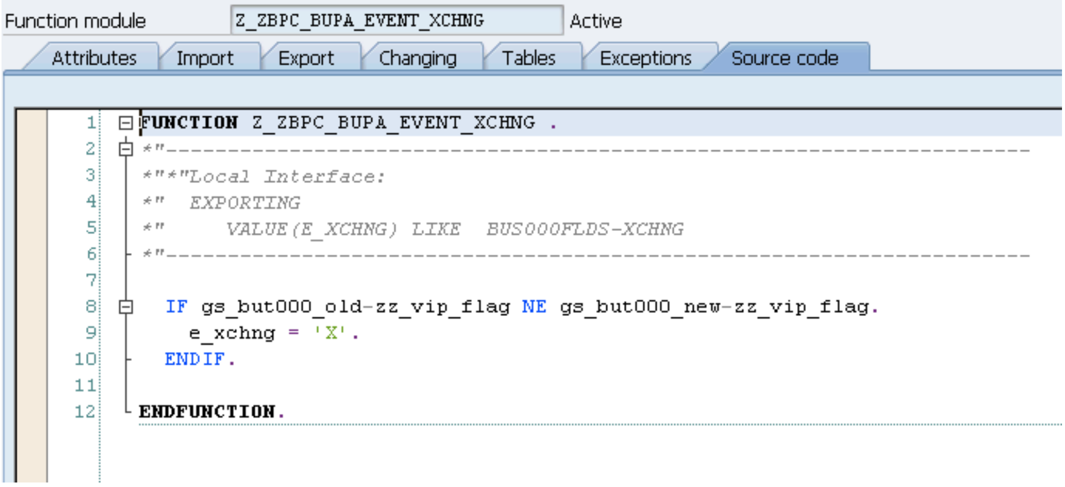
Function module Z_ZBPC_BUPA_EVENT_ISDST Active

```
FUNCTION Z_ZBPC_BUPA_EVENT_ISDST .
*"----------------------------------------------------------------------
*"*"Local Interface:
*"----------------------------------------------------------------------

* Get the values from the ownig application
  CALL FUNCTION 'BUP_BUPA_BUT000_GET'
    IMPORTING
      e_but000 = gs_but000_old
    .

* Initialise the global memory with the values
  gs_but000_new = gs_but000_old.

* Fill the subscreen field with the current value
  but000-zz_vip_flag = gs_but000_old-zz_vip_flag.

ENDFUNCTION.
```

Figure 34: Function module for event ISDST of the Business Partner

Event XCHNG (checking if your field value has changed)

In event XCHNG you have to check if the user changed the value of your custom field:



Function module Z_ZBPC_BUPA_EVENT_XCHNG Active

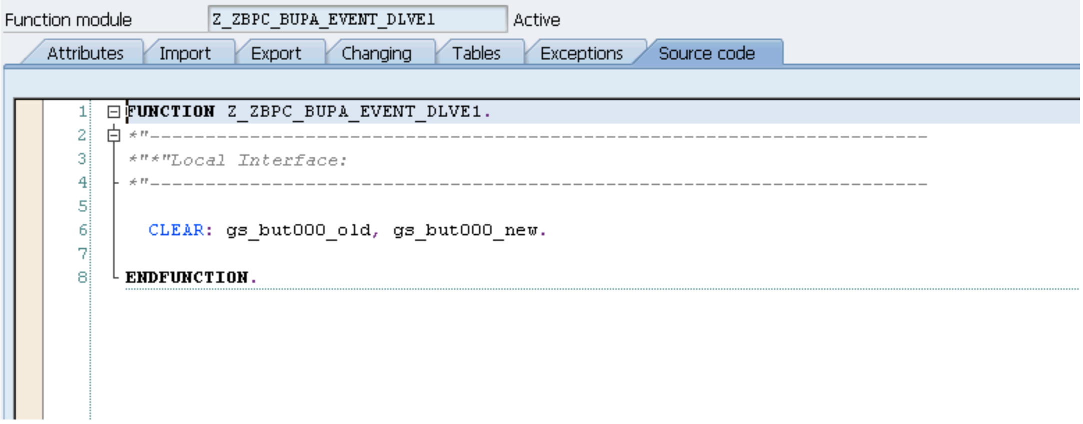
```
FUNCTION Z_ZBPC_BUPA_EVENT_XCHNG .
*"----------------------------------------------------------------------
*"*"Local Interface:
*"  EXPORTING
*"     VALUE(E_XCHNG) LIKE  BUS000FLDS-XCHNG
*"----------------------------------------------------------------------

  IF gs_but000_old-zz_vip_flag NE gs_but000_new-zz_vip_flag.
    e_xchng = 'X'.
  ENDIF.

ENDFUNCTION.
```

Figure 35: Function module for BDT event XCHNG

The function module for event XCHNG has one exporting parameter. Find an SAP function module in the event configuration or set it up as shown in my example above.

Event DLVE1 (reset your function groups memory)

In event DLVE1 you have to reset your function groups global memory. This function module has no parameters.



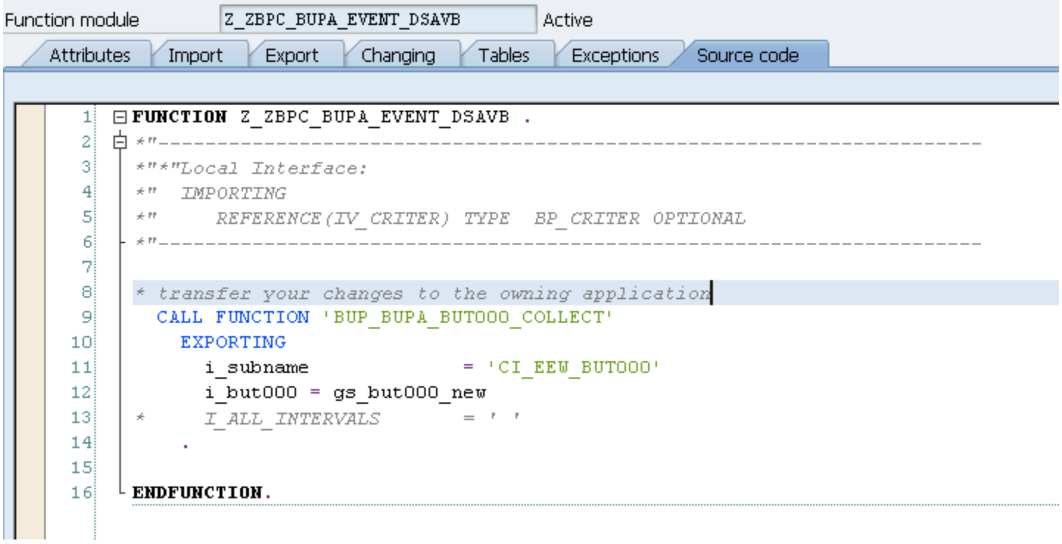
Function module Z_ZBPC_BUPA_EVENT_DLVE1 Active

```
FUNCTION Z_ZBPC_BUPA_EVENT_DLVE1.
*"----------------------------------------------------------------------
*"*"Local Interface:
*"----------------------------------------------------------------------

  CLEAR: gs_but000_old, gs_but000_new.

ENDFUNCTION.
```

Figure 36: Function module for BDT event DLVE1

Event DSAVB (Transferring your field value to the owning application)

In event DSAVB you have to give the owning application your custom field values. Calling the Collect function module we determined before does this. This is again another function module without parameters.

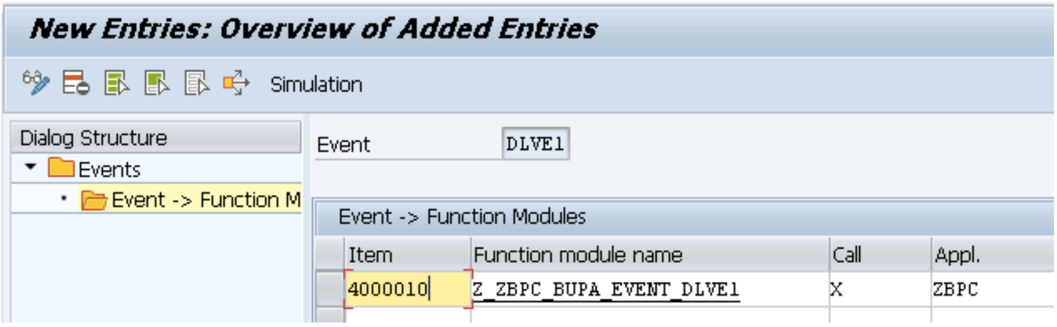


Function module Z_ZBPC_BUPA_EVENT_DSAVB Active

```
FUNCTION Z_ZBPC_BUPA_EVENT_DSAVB .
*"----------------------------------------------------------------------
*"*"Local Interface:
*"  IMPORTING
*"     REFERENCE(IV_CRITER) TYPE  BP_CRITER OPTIONAL
*"----------------------------------------------------------------------

* transfer your changes to the owning application
  CALL FUNCTION 'BUP_BUPA_BUT000_COLLECT'
    EXPORTING
      i_subname             = 'CI_EEW_BUT000'
      i_but000 = gs_but000_new
*     I_ALL_INTERVALS       = ' '
    .

ENDFUNCTION.
```

Figure 37: Function module for BDT event DSAVB

After creating those event function modules, you must put them in the event configuration shown above (Figure 30: BDT events in the area menu).


New Entries: Overview of Added Entries
Simulation
Dialog Structure
Events
Event -> Function M
Event -> Function Modules

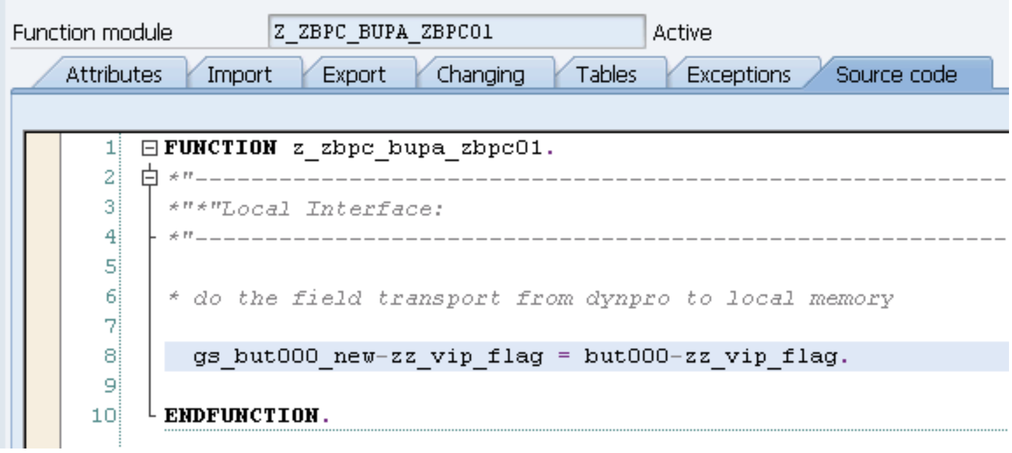
Event DLVE1

Item	Function module name	Call	Appl.
4000010	Z_ZBPC_BUPA_EVENT_DLVE1	X	ZBPC

Figure 38: Configuring the function module for event DLVE1

PAI function module for the view

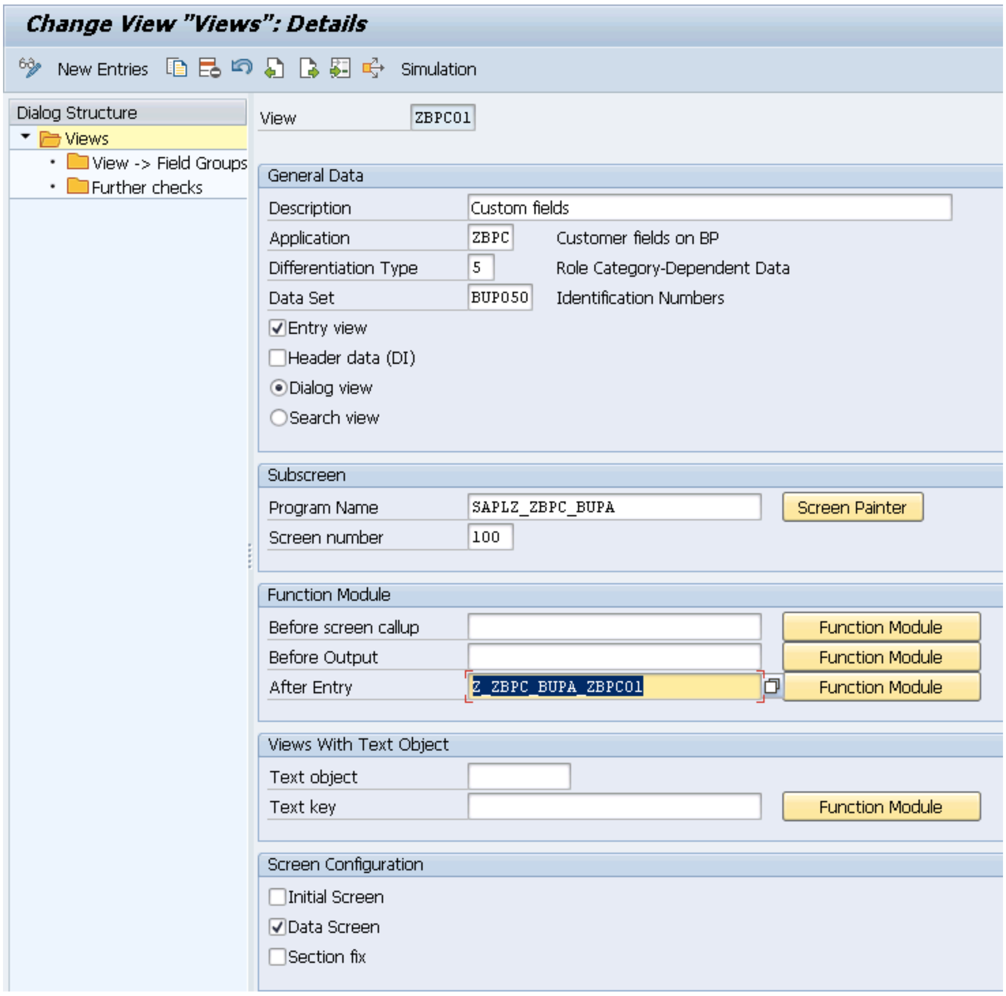
There is one last thing to implement: It’s the Process After Input function module for the view we created. Do not put this functionality in the PAI module of the sub screen. Create a new function module without parameters and use this naming convention: Z_<appl>_<obj>_<view> with <view> being the view name.



Function module Z_ZBPC_BUPA_ZBPC01 Active

```
FUNCTION z_zbpc_bupa_zbpc01.
*"----------------------------------------------------------------
*"*"Local Interface:
*"----------------------------------------------------------------

* do the field transport from dynpro to local memory

  gs_but000_new-zz_vip_flag = but000-zz_vip_flag.

ENDFUNCTION.
```

Figure 39: Implementing the PAI function module for the view



View: ZBPC01

General Data

Description: Custom fields

Application: ZBPC Customer fields on BP

Differentiation Type: 5 Role Category-Dependent Data

Data Set: BUP050 Identification Numbers

☑ Entry view

□ Header data (DI)

● Dialog view

○ Search view

Subscreen

Program Name: SAPLZ_ZBPC_BUPA

Screen number: 100

Function Module

Before screen callup:

Before Output:

After Entry: Z_ZBPC_BUPA_ZBPC01

Views With Text Object

Text object:

Text key:

Screen Configuration

□ Initial Screen

☑ Data Screen

□ Section fix

Figure 40: Adding the PAI function module to the view

This PAI function module is needed to transfer the changed field value to our function group memory.

That's it! You just added a customer field to your application without knowing BDT in detail. Congratulations!

I'm always open to feedback, so please let me know if anything in this short course was unclear or needs further details.

You just learned **how** to add a custom field to the BDT, but I omitted most of the time **why** you had to choose certain settings.

Would you be interested to know why those settings were made?

What the other settings, not mentioned, are good for?

What if you need to do more complex BDT tasks, like adding a new check to an SAP field, adding a whole new customer table to an application?

I'm in the progress of creating an eLearning course that covers every aspect of the Business Data Toolset.

If you have got any questions or issues regarding BDT, drop me a line at bdt@apadev.co.uk and I'm happy to include the information/explanations/solutions in the upcoming course.

Don't hesitate to reach out.

Kind Regards,

Frank Vieregge