**BP增强 添加自定义屏幕 KNA1 通用数据（英文）**

其他参考（中文）：<https://blog.csdn.net/guangcong2009/article/details/80569877>

                英文：<http://www.saptechnical.com/Tutorials/ExitsBADIs/BP/Page6.htm>

Step by step explanation of how to add a new tab to BP screen.

* Add new fields to standard structure incl\_eew\_kna1. incl\_eew\_kna1 is included in KNA1. Fields will be in KNA1 automatically.

[*Click on images to see with original size.*](http://www.abaplog.com/Uploads/pic_kna1_add.png)

* Add same fields to incl\_eew\_kna1\_x. Make sure that data element must be char1 for ex. bapiupdate. [*Click on images to see with original size.*](http://www.abaplog.com/Uploads/pic_kna1_x_add.png)
* Create a Z function group. IN top include define varibles:

[*Click on images to see with original size.*](http://www.abaplog.com/Uploads/fg_1.png)

1. TABLES:
2. kna1.
4. constants:
5. table\_name\_kna1 type fsbp\_table\_name value 'KNA1',
6. false type boole-boole value ' '.
8. DATA:
9. gs\_kna1 TYPE kna1.
10. Create a new screen in function group.
11. IN PBO of sceen call function ‘BUS\_PBO’ and in PAI call BUS\_PAI.
    1. MODULE status\_9001 OUTPUT.
    2. CALL FUNCTION *'BUS\_PBO'.*
    3. ENDMODULE.
    5. user\_command\_9001 INPUT.
    6. CALL FUNCTION *'BUS\_PAI'.*
    7. ENDMODULE.

1. In screen layout press F6.Get KNA1-Z fields to the screen. Activate the screen and function group.

Lets begin to customizing.

* Go to BUPT t-code.

[*Click on images to see with original size.*](http://www.abaplog.com/Uploads/bupt.png)

* Create a new application named ‘ZKNA’ with New Entries button. Bussiness Partner->Control->Applications. Define a description and active is selected.

[*Click on images to see with original size.*](http://www.abaplog.com/Uploads/applications.png)

* Create a new dataset named ‘ZKNA01’ New Entries button. Bussiness Partner->Control->Data Sets.  [*Click on images to see with original size.*](http://www.abaplog.com/Uploads/datasets.png)
* Create a new field group named 601 New Entries button. Bussiness Partner->Control->Screen Layout->Field Groups.  Set a description and save.

[*Click on images to see with original size.*](http://www.abaplog.com/Uploads/field%20groups.png)

* Assign fields to fields group. Table and field name should be added on screen that is created in Z function group. [*Click on images to see with original size.*](http://www.abaplog.com/Uploads/field%20group%20fields.png)
* Create a new view named ZKNA01 New Entries button. Bussiness Partner->Control->Screen Layout->Views. Set a description.Set Application we created before, same as Data Set. Set program name and screen number. Save the view. Double click on View -> Field Groups. Set field group number we created. Now we should set fm names for before output and after entry. [*Click on images to see with original size.*](http://www.abaplog.com/Uploads/views.png)[*Click on images to see with original size.*](http://www.abaplog.com/Uploads/view-fields.png)
* Create a fm for before output named zsd\_kna1\_pbo. Copy ‘CVIC\_BUPA\_PBO\_CVIC03’. Assign fm to Z function group.

1. function ZSD\_KNA1\_PBO.
2. \*"--------------------------------------------------------------------
3. \*"\*"Local Interface:
4. \*"--------------------------------------------------------------------
5. data:
6. lt\_kna1 type table of kna1,
7. lt\_knas type table of knas.
9. \* step 1: request data from xo for dynpro structure
10. cvi\_bdt\_adapter=>data\_pbo(
11. exporting
12. i\_table\_name = table\_name\_kna1
13. importing
14. e\_data\_table = lt\_kna1[]
15. ).
16. if lt\_kna1[] is initial.
17. clear gs\_kna1.
18. else.
19. read table lt\_kna1 into gs\_kna1 index 1.
20. endif.
22. endfunction.

* Create a fm for after entry named zsd\_kna1\_pai. Copy ‘CVIC\_BUPA\_PAI\_CVIC03’. Assign fm to Z function group.change

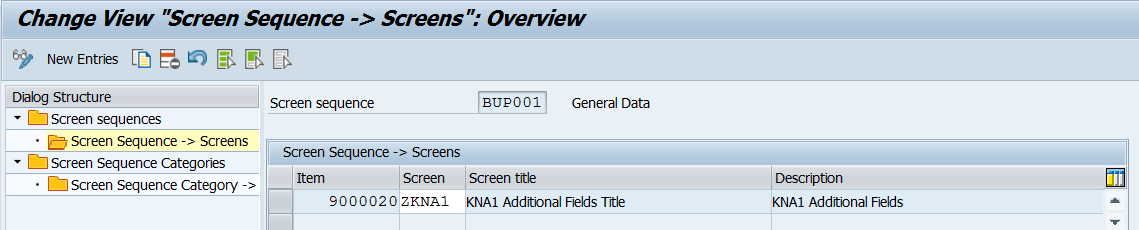
1. function ZSD\_KNA1\_PAI.
2. \*"--------------------------------------------------------------------
3. \*"\*"Local Interface:
4. \*"--------------------------------------------------------------------
5. data:
6. lt\_kna1 type table of kna1.
7. field-symbols:
8. <kna1> like line of lt\_kna1.
10. check cvi\_bdt\_adapter=>is\_direct\_input\_active( ) = false.
11. \* step 1: update xo memory from dypro structure
12. cvi\_bdt\_adapter=>get\_current\_bp\_data(
13. exporting
14. i\_table\_name = table\_name\_kna1
15. importing
16. e\_data\_table = lt\_kna1[]
17. ).
19. if lt\_kna1[] is initial.
20. if gs\_kna1 is not initial.
21. gs\_kna1-kunnr = cvi\_bdt\_adapter=>get\_current\_customer( ).
22. append gs\_kna1 to lt\_kna1.
23. endif.
24. else.
25. read table lt\_kna1 assigning <kna1> index 1.
26. <kna1>-zz\_field1 = gs\_kna1-zz\_field1.
27. endif.
29. cvi\_bdt\_adapter=>data\_pai(
30. i\_table\_name = table\_name\_kna1
31. i\_data\_new = lt\_kna1[]
32. i\_validate = false
33. ).
35. endfunction.

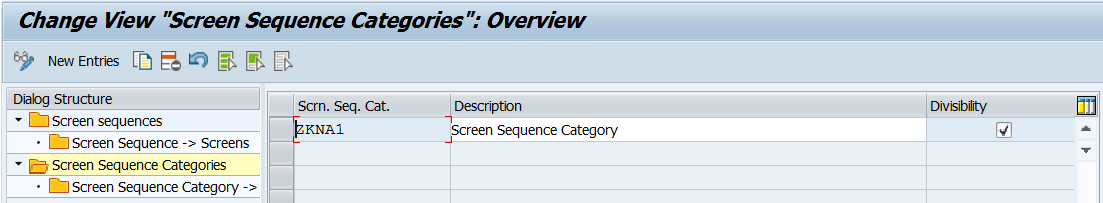
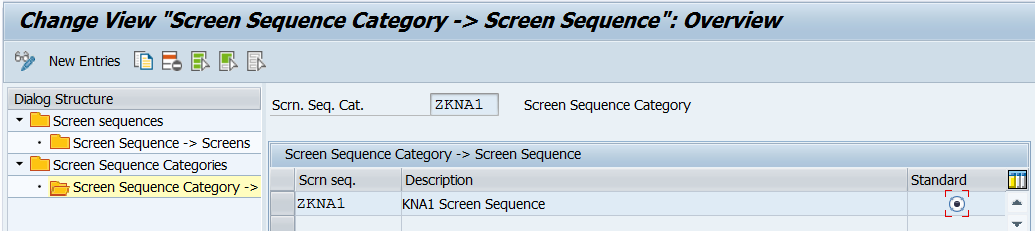
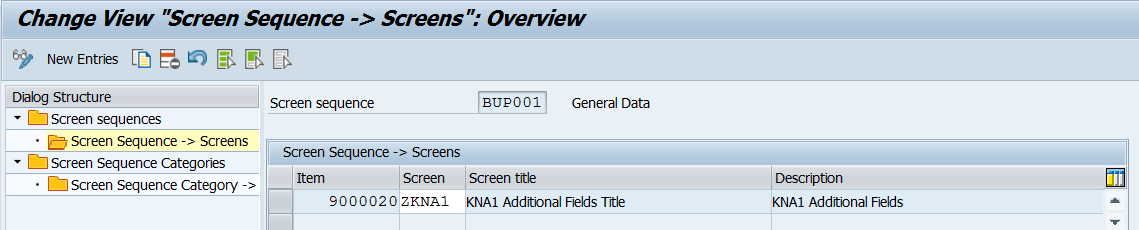
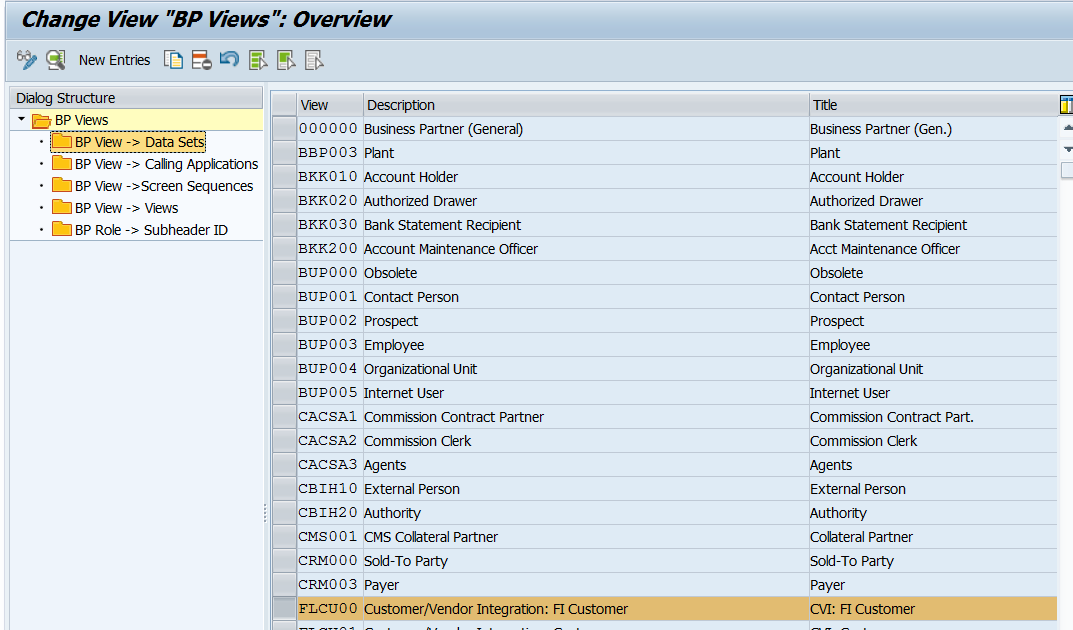
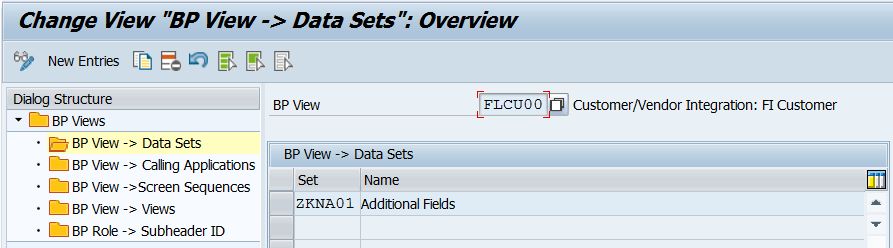
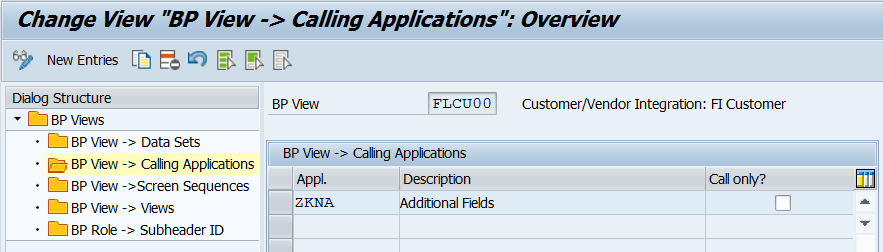
Set these fm’s to view by Bussiness Partner->Control->Screen Layout->Views.[*Click on images to see with original size.*](http://www.abaplog.com/Uploads/view%20details.png)

* Create a new section named ZKNA1 New Entries button. Bussiness Partner->Control->Screen Layout->Sections. Set a description and title.

[*Click on images to see with original size.*](http://www.abaplog.com/Uploads/sections.png)

* Select the section we created and double click on Section -> Views. Set item number and view name we created. Save changes. [*Click on images to see with original size.*](http://www.abaplog.com/Uploads/section%20views.png)
* Create a new screen named ZKNA1 New Entries button. Bussiness Partner->Control->Screen Layout->Screens. Set a description and screen title. Save changes. [*Click on images to see with original size.*](http://www.abaplog.com/Uploads/screens.png)
* Select the screen we created and double click on Screen -> Sections. Set first item section as BUP009 and second number will be section we created. Save changes. [*Click on images to see with original size.*](http://www.abaplog.com/Uploads/screen%20sections.png)
* Create a new screen sequence named ZKNA1 New Entries button. Bussiness Partner->Control->Screen Sequenes. Set a description and save changes. [*Click on images to see with original size.*](http://www.abaplog.com/Uploads/screen%20seq.png)
* Select the screen seq. we created and double click on Screen Sequence -> Screens. Set item number as screen name that we created. Save changes.

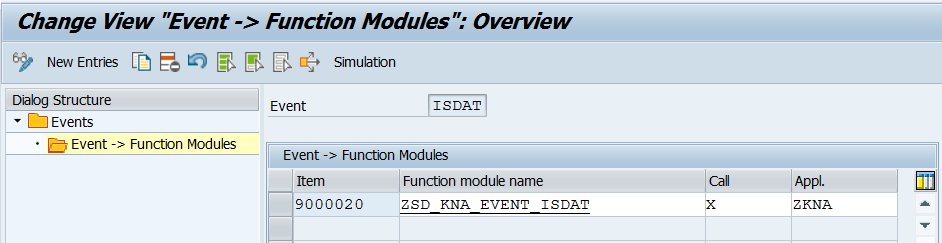
[[](http://www.abaplog.com/Uploads/screen%20seq%20screens.png)*Click on images to see with original size.*](http://www.abaplog.com/Uploads/screen%20seq%20screens.png)

* Double click on Screen Sequence Categories. Create a new category named ZKNA1 New Entries button. Divisibility is selected. [[](http://www.abaplog.com/Uploads/screen%20seq%20cat.png)*Click on images to see with original size.*](http://www.abaplog.com/Uploads/screen%20seq%20cat.png)
* Select the screen seq. cat. we created and double click on Screen Sequence Category -> Screens Sequence. Set screen seq that we created. Standard is selected. Save changes. [[](http://www.abaplog.com/Uploads/screen%20seq%20cat%20screens.png)*Click on images to see with original size.*](http://www.abaplog.com/Uploads/screen%20seq%20cat%20screens.png)
* Go back to screen sequences. Select BUP001 and double click on Screen Sequence -> Screens. Set item number like 9000000 and set screen we created. Save changes. [[](http://www.abaplog.com/Uploads/screen%20seq%20screens.png)*Click on images to see with original size.*](http://www.abaplog.com/Uploads/screen%20seq%20screens.png)
* In my example I will create a new tab in FI Customer role(FLCU00).  Go to Bussiness Partner->Control->Divisibility->BP Views.
* Select FLCU00. Assing Data Sets with BP View -> Data Sets. Set Data Set name we created. Save changes. [[](http://www.abaplog.com/Uploads/bp%20views.png)*Click on images to see with original size.*](http://www.abaplog.com/Uploads/bp%20views.png) [[](http://www.abaplog.com/Uploads/bp%20view%20dataset.png)*Click on images to see with original size.*](http://www.abaplog.com/Uploads/bp%20view%20dataset.png)
* Double click BP View -> Calling Applications. Set Application we created..Save changes. [[](http://www.abaplog.com/Uploads/bp%20view%20app.png)*Click on images to see with original size.*](http://www.abaplog.com/Uploads/bp%20view%20app.png)

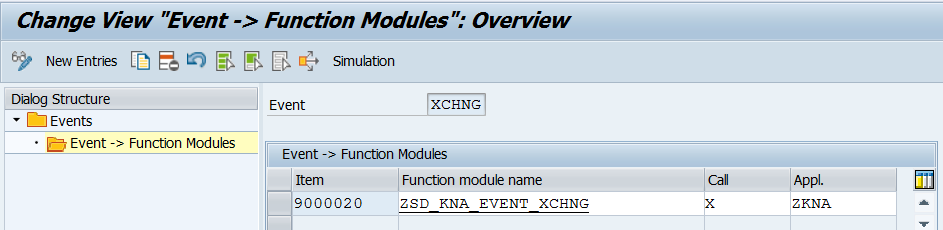
Now we can see additional tab in BP under FLCU00. Events must be defined for this app.

* Go to Bussiness Partner->Control->Events->Bussiness Data Toolset. To set first data we should use ISDAT event. Select ISDAT and double click on Events -> Function Modules. Here we should create a new function in the same fg.  ZSD\_KNA1\_EVENT\_ISDAT. Set Call as ‘X’. And set app name. Save changes.

1. FUNCTION zsd\_kna\_event\_isdat.
2. \*"----------------------------------------------------------------------
3. \*"\*"Local Interface:
4. \*"----------------------------------------------------------------------
6. CALL FUNCTION 'CVIC\_BUPA\_KNA1\_GET'
7. IMPORTING
8. e\_kna1 = gs\_kna1.
10. kna1-zz\_field1 = gs\_kna1-zz\_field1.
12. ENDFUNCTION.

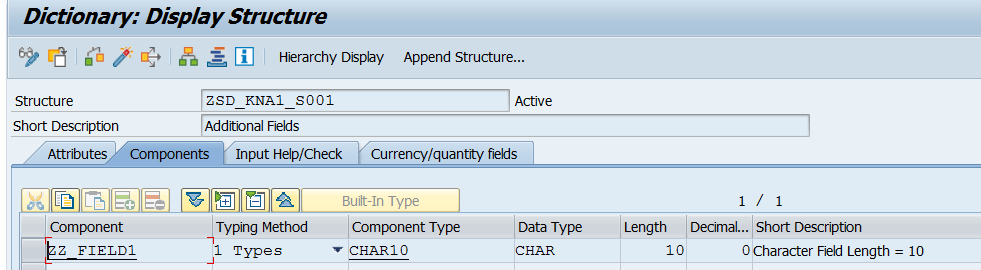
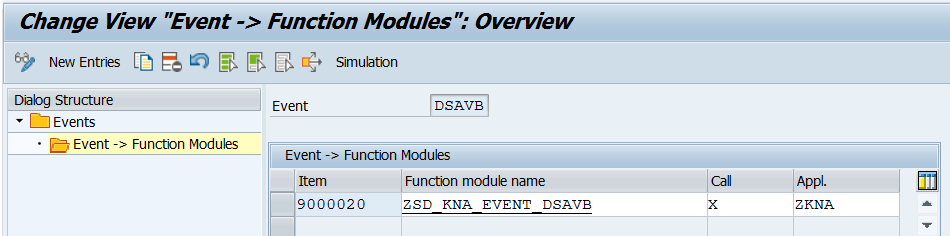
[[](http://www.abaplog.com/Uploads/isdat.png)*Click on images to see with original size.*](http://www.abaplog.com/Uploads/isdat.png)

1. Turn back to Events. Now we should check if anything is changed in the screen with XCHNG event. Double click on fms. Here we should create a new function in the same fg.  ZSD\_KNA1\_EVENT\_XCHNG. Set Call as ‘X’. And set app name. Save changes.
   1. FUNCTION zsd\_kna\_event\_xchng.
   2. \*"----------------------------------------------------------------------
   3. \*"\*"Local Interface:
   4. \*" EXPORTING
   5. \*" REFERENCE(E\_XCHNG) TYPE BOOLE\_D
   6. \*"----------------------------------------------------------------------
   8. IF gs\_kna1-zz\_field1 NE kna1-zz\_field1.
   9. e\_xchng = 'X'.
   10. ENDIF.
   12. ENDFUNCTION.

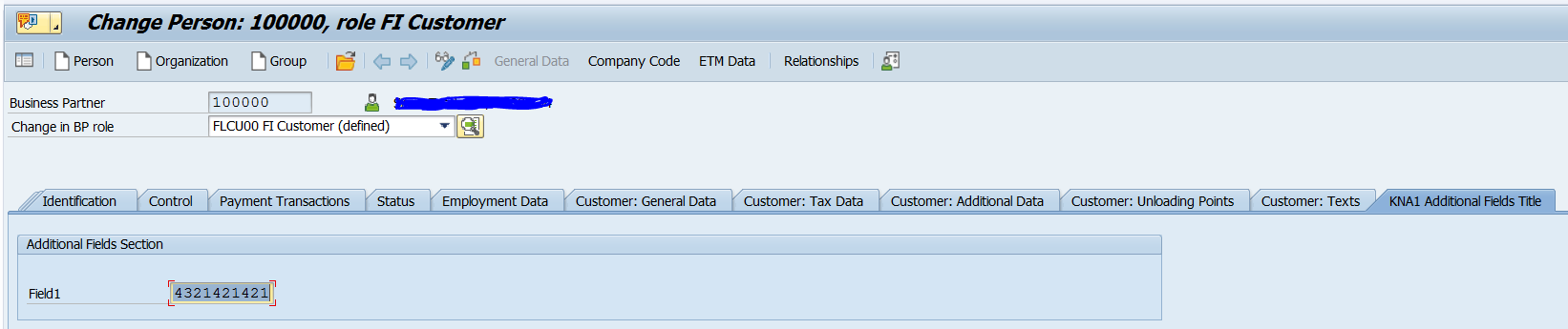
[[](http://www.abaplog.com/Uploads/xchng.png)*Click on images to see with original size.*](http://www.abaplog.com/Uploads/xchng.png)

1. Turn back to Events. Now we should set changed values with DSAVB event. Double click on fms. Here we should create a new function in the same fg.  ZSD\_KNA1\_EVENT\_DSAVB. Z Structure is created with Z fields. Set Call as ‘X’. And set app name. Save changes.
   1. FUNCTION zsd\_kna\_event\_dsavb.
   2. \*"--------------------------------------------------------------------
   3. \*"\*"Local Interface:
   4. \*"--------------------------------------------------------------------
   6. DATA : ls\_kna1 TYPE kna1.
   7. ls\_kna1 = kna1.
   8. MOVE-CORRESPONDING gs\_kna1 TO kna1.
   9. kna1-zz\_field1 = ls\_kna1-zz\_field1.

   12. CALL FUNCTION 'CVIC\_BUPA\_KNA1\_COLLECT'
   13. EXPORTING
   14. i\_subname = 'ZSD\_KNA1\_S001'
   15. i\_kna1 = kna1.
   17. ENDFUNCTION.

[[](http://www.abaplog.com/Uploads/s001.png)*Click on images to see with original size.*](http://www.abaplog.com/Uploads/s001.png)                                                [[](http://www.abaplog.com/Uploads/dsavb.png)*Click on images to see with original size.*](http://www.abaplog.com/Uploads/dsavb.png)

Now everything looks fine. Lets test it.

[[](http://www.abaplog.com/Uploads/bp.png)*Click on images to see with original size.*](http://www.abaplog.com/Uploads/bp.png)

Now we can get current value and change it. With the same solution we can display/update fields in LFA1, KNB1 etc. We only need to find the right fm's for them.  Hopefully it will be useful for you.