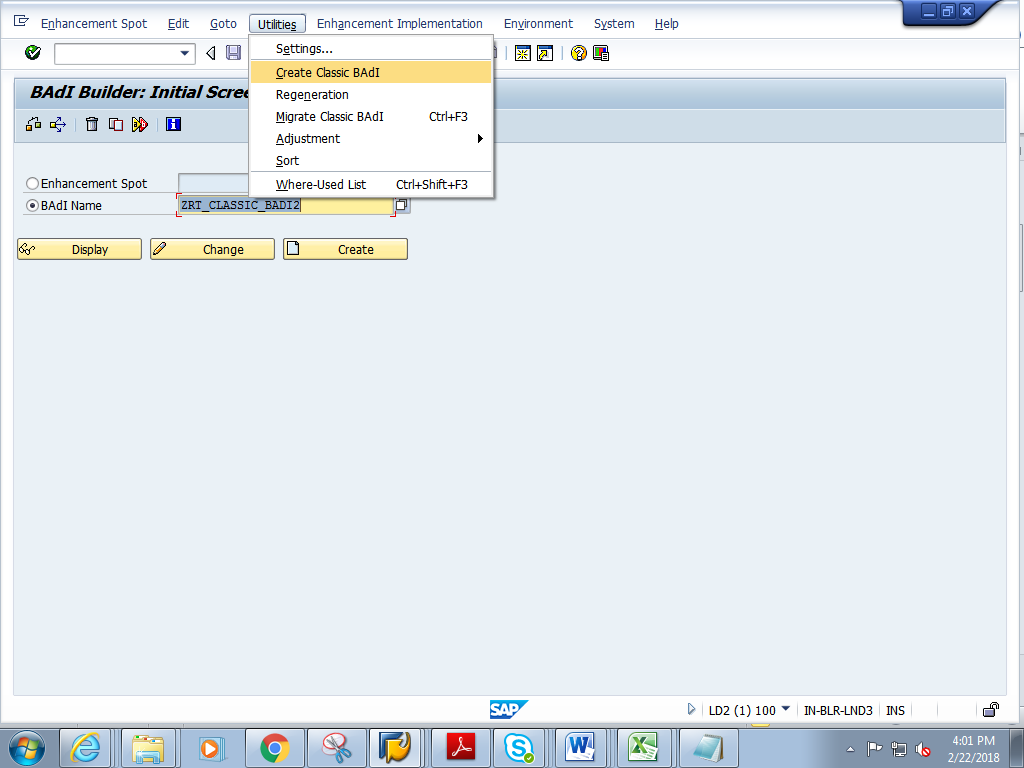
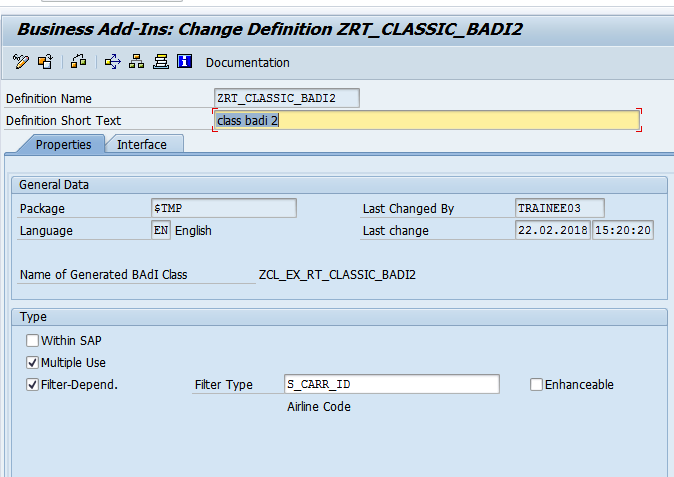
**Classic BADI with FILTER**

1. Go to tcode SE18.Give a BADI name starting with Z/X. Go to Utilities->Create Classic BADI.

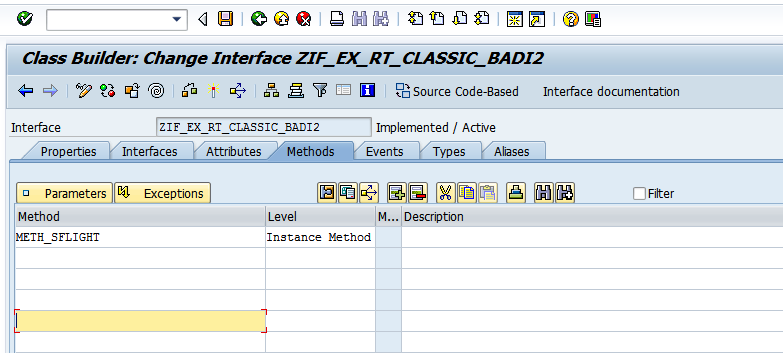
(Do not click create)



1. Give a short description. **Check the Filter Depend**. **In the Filter type, give the Data Element name Eg. S\_CARR\_ID .**  
   Click on the interface tab.



1. In the interface, add the method METH\_FLIGHT as shown below. Click on Parameters



1. Give 2 parameters I\_CARRID and E\_SFLIGHT

(Note: ZBADI\_SFLI is a table type of line type SFLIGHT)

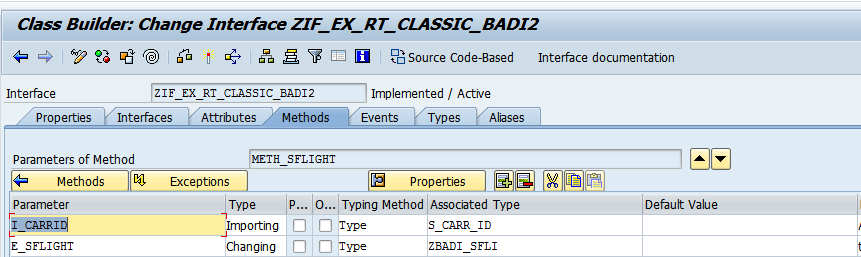
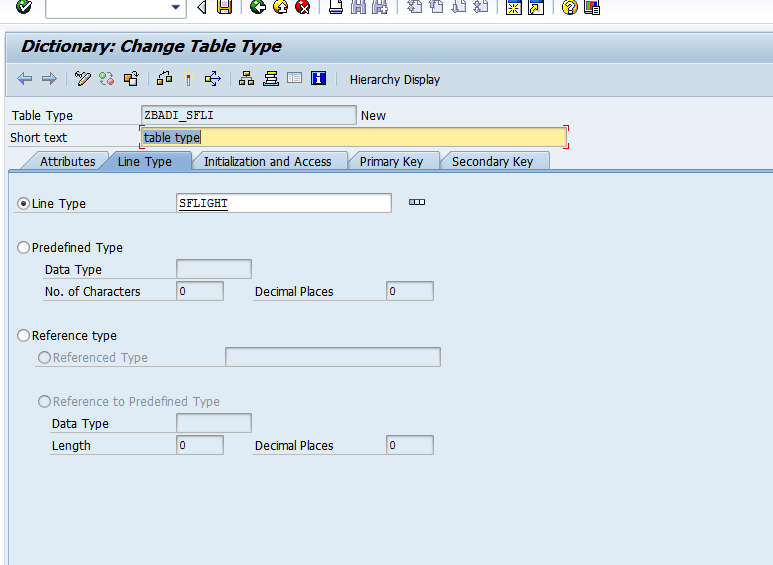
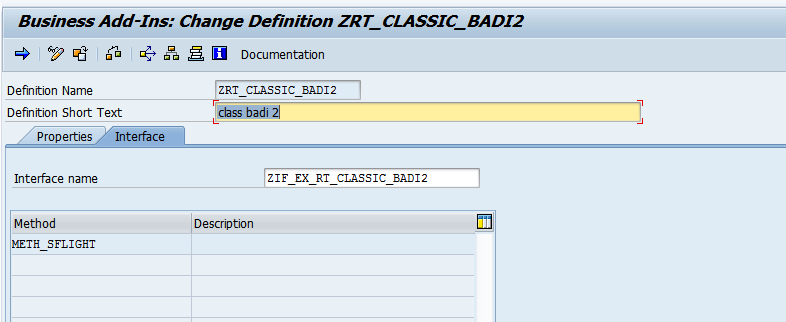


Table type :ZBADI\_SFLI

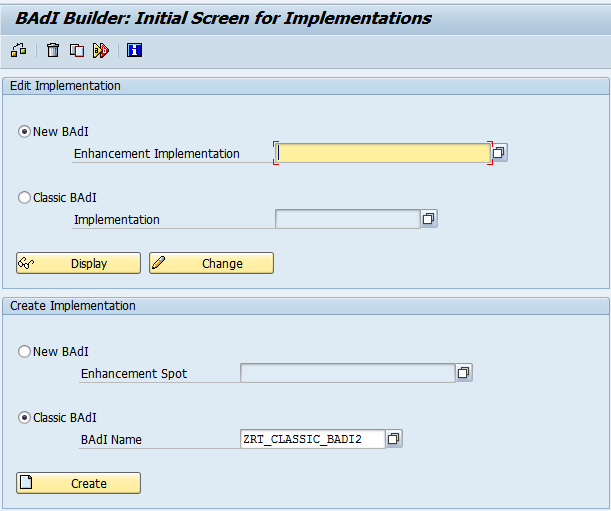


1. Save and activate the BADI

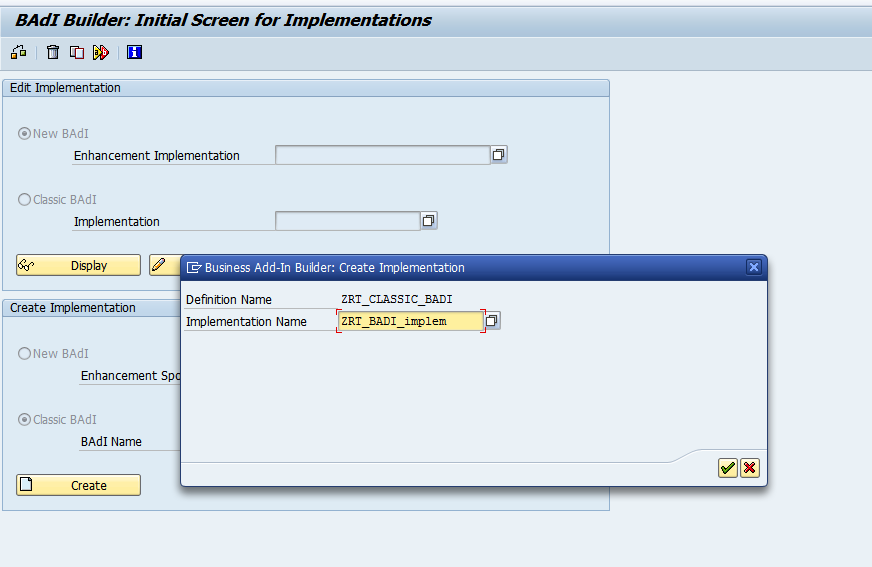


1. Go to tcode SE19. In the create implementation section, choose Classic BADI->give the name of the BADI created .

Click on create.

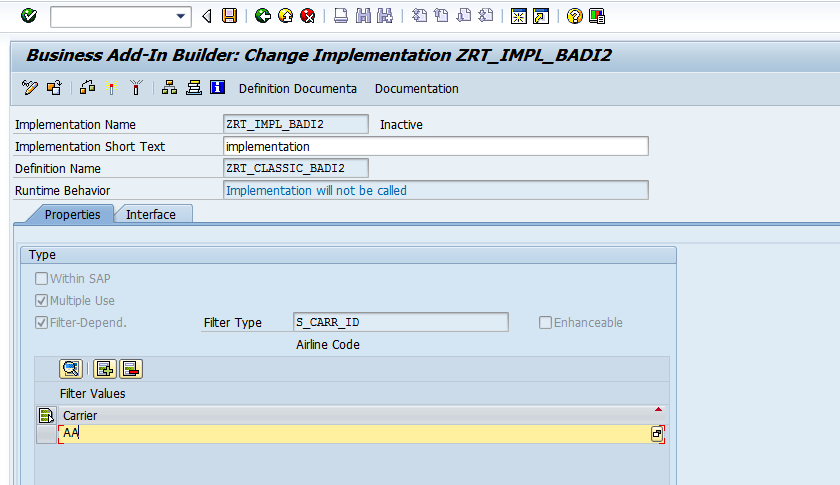


1. Give a suitable implementation name starting with Z/Y. Click on continue

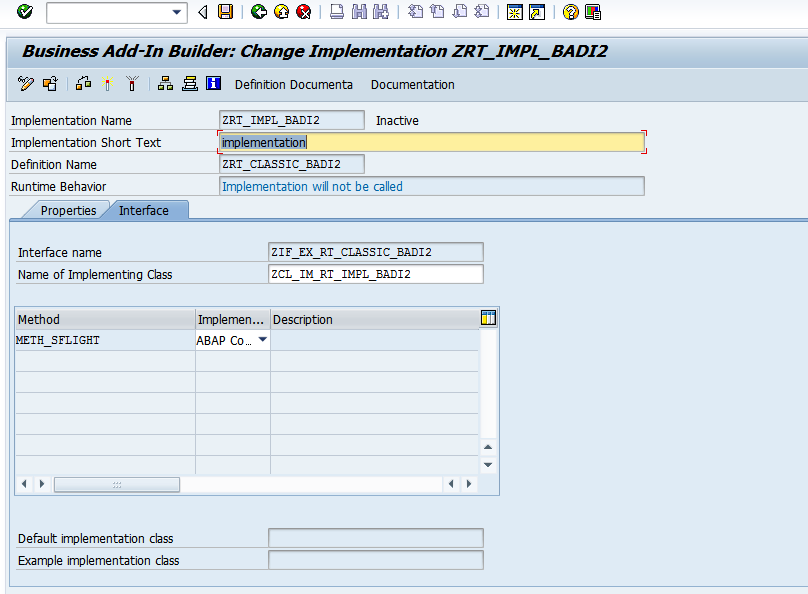


1. Give short description. **Click on the Plus symbol and add the value of CARRID Eg. AA.**

Click on the interface tab.

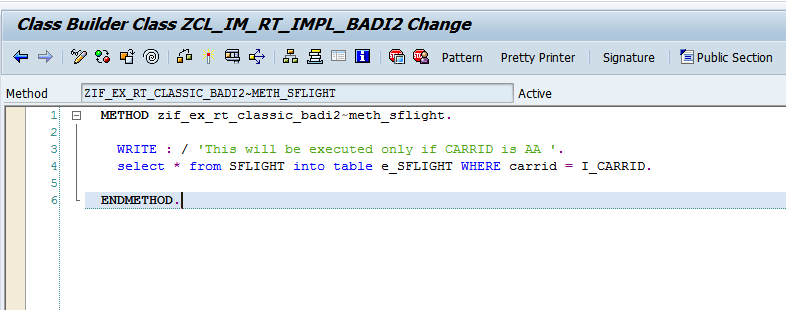


1. In the interface tab, double click on class and create it.

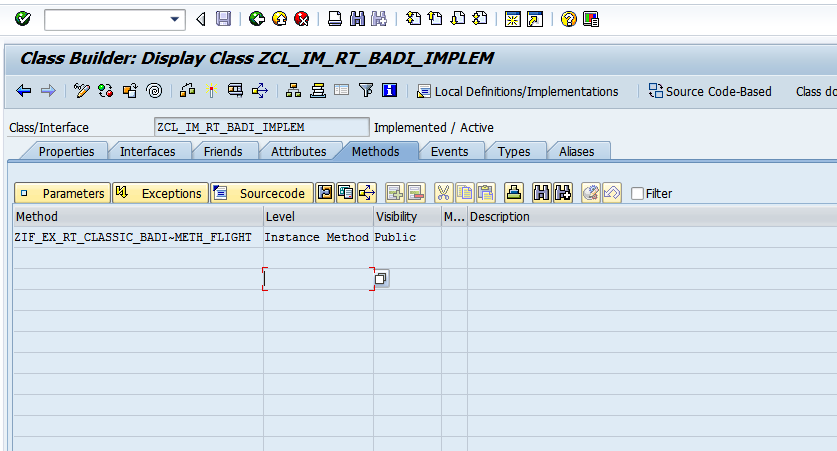


  select  carrid connid fldate seatsmax from sflight  
       into table ex\_sflight  
        where  carrid = i\_Carrid.

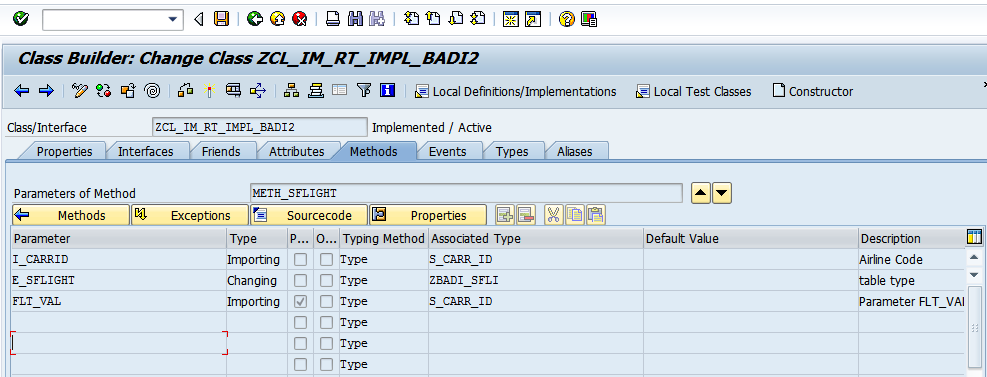
1. The class opens up. In the method METH\_FLIGHT, add the below logic



1. Save the class and go back.



1. Save and activate the BADI Implementation. You will see a new importing parameter FLT\_VAL gets populated

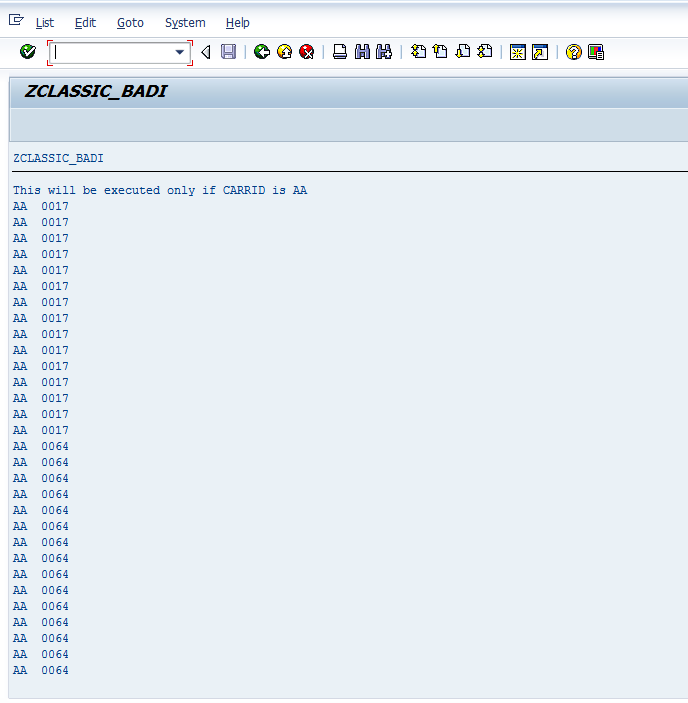


1. Create a Report in se38 and add the below logic

*\*&---------------------------------------------------------------------\**  
*\*& Report ZCLASSIC\_BADI\_PRG*  
*\*&---------------------------------------------------------------------\**  
*\*&*  
*\*&---------------------------------------------------------------------\**

REPORT zrt\_classic\_badi\_prg.  
  
INCLUDE zrtassic\_badi\_prg\_top.  
  
  
PARAMETERS p\_carrid TYPE s\_carr\_id.  
  
  
START-OF-SELECTION.  
  
  DATA obj1 TYPE REF TO  zif\_ex\_rt\_classic\_badi2.*"Interface from se19*  
  
  
  CALL METHOD cl\_exithandler=>get\_instance  
    EXPORTING  
      exit\_name = 'ZRT\_CLASSIC\_BADI2' *"Name of badi definition name*  
*\*     null\_instance\_accepted        = SEEX\_FALSE*  
*\*  IMPORTING*  
*\*     act\_imp\_existing              =*  
    CHANGING  
      instance  = obj1  
*\*  EXCEPTIONS*  
*\*     no\_reference                  = 1*  
*\*     no\_interface\_reference        = 2*  
*\*     no\_exit\_interface             = 3*  
*\*     class\_not\_implement\_interface = 4*  
*\*     single\_exit\_multiply\_active   = 5*  
*\*     cast\_error                    = 6*  
*\*     exit\_not\_existing             = 7*  
*\*     data\_incons\_in\_exit\_managem   = 8*  
*\*     others    = 9*  
    .  
  IF sy-subrc <> 0.  
*\* Implement suitable error handling here*  
  ENDIF.  
  
  
  
  CALL METHOD obj1->meth\_sflight  
    EXPORTING  
      i\_carrid  = p\_carrid  
**flt\_val   = p\_carrid  
    CHANGING  
      e\_sflight = it\_tab.**  
  
  
  LOOP AT it\_tab INTO wa\_tab.  
    WRITE : / wa\_tab-carrid,  
              wa\_tab-connid.  
  ENDLOOP.  
  
  IF it\_tab IS INITIAL.  
    WRITE : / 'BADI was not executed'.  
  ENDIF.

10) The expected output is as follows when CARRID = AA



11) When CARRID is not equal to AA. The BADI is not executed.

