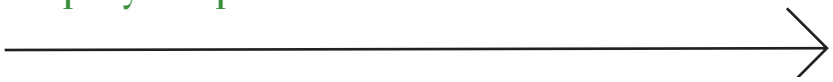
SAP ABAP on HANA

Step by Step Guide to Create BAPI In ABAP



Step By Step Guide to Create BAPI in ABAP

What is BAPI?

Business Application Programming Interface(BAPI) are standardized programming interfaces (methods) enabling external applications to access business processes and data in the SAP System.

They **provide stable and standardized methods** to achieve seamless integration between the SAP System and external applications, legacy systems and add-ons.

BAPIs are defined in the **BOR(Business object repository)** as methods of SAP business object types that carry out specific business functions. They are implemented as **RFC-enabled function** modules and are created in the Function Builder of the ABAP Workbench.

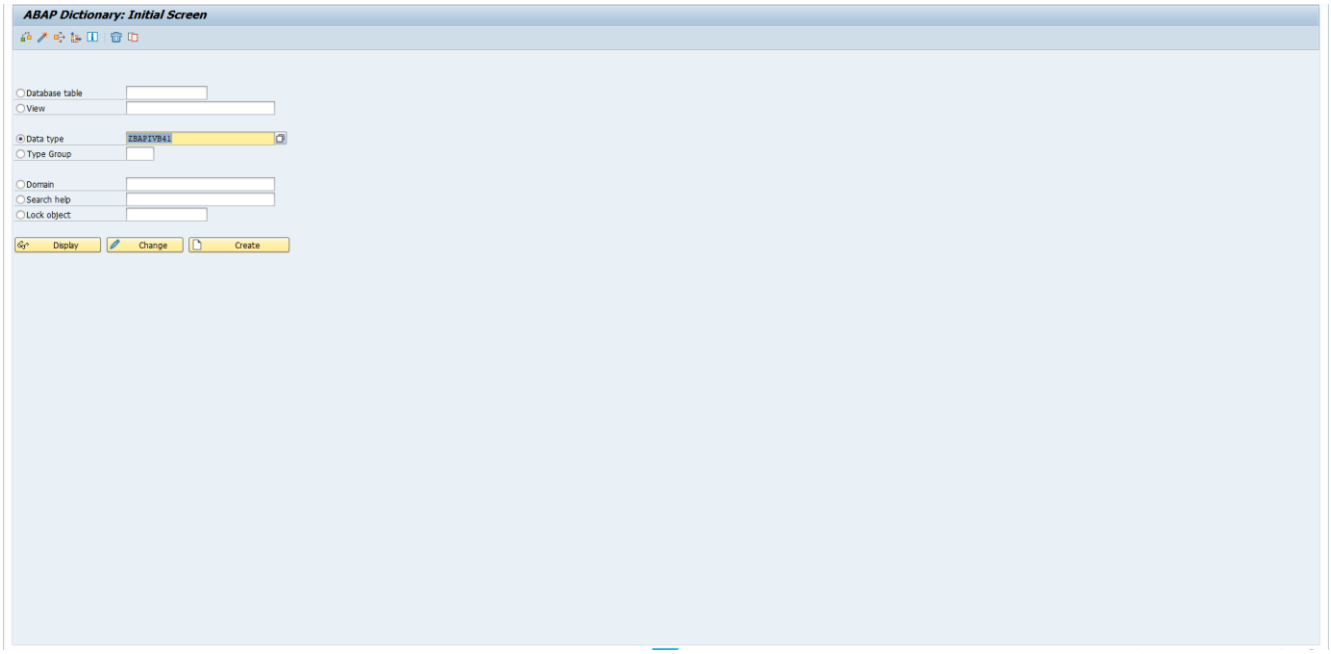
Some BAPIs and methods provide basic functions and can be used for most SAP Business Objects. These are called **STANDARDIZED BAPI's.**

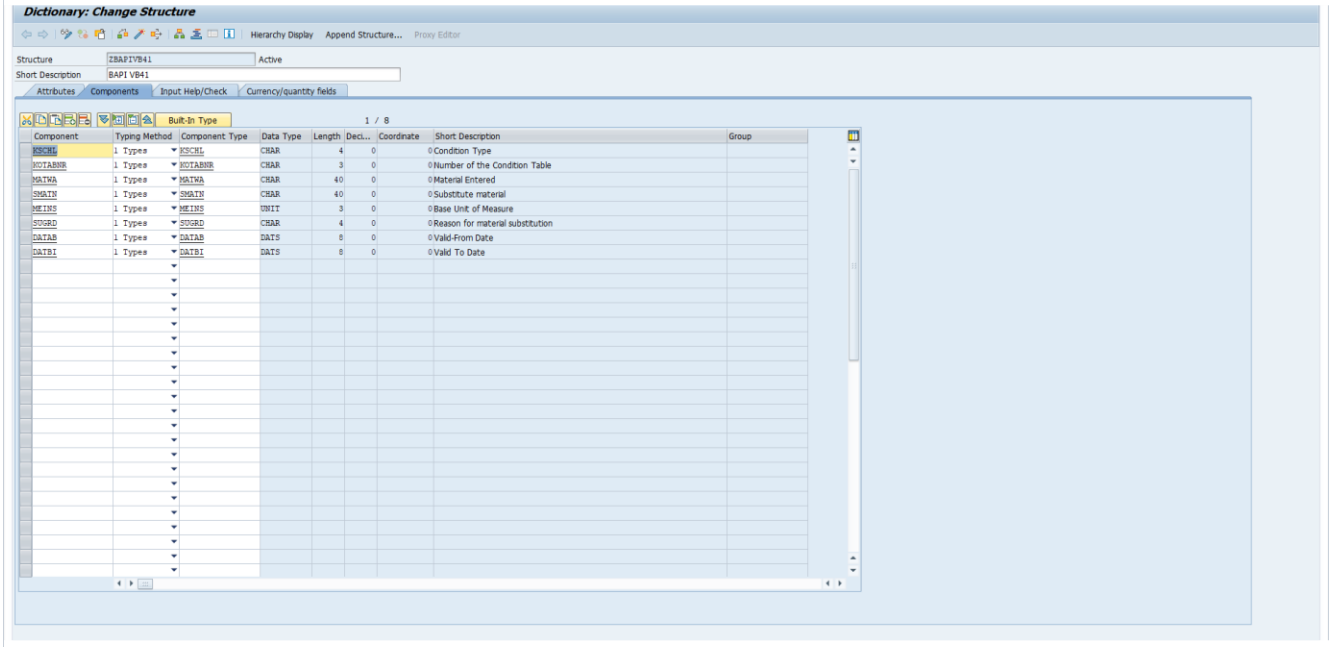
List of Standardized BAPIs:

- BAPIs for Reading Data – GetList() , GetDetail() , GetStatus() , ExistenceCheck()
- BAPIs for Creating or Changing Data- Create() ,Change(),Delete() and Undelete() ,
- BAPIs for Mass Processing -ChangeMultiple(), CreateMultiple(), DeleteMultiple().

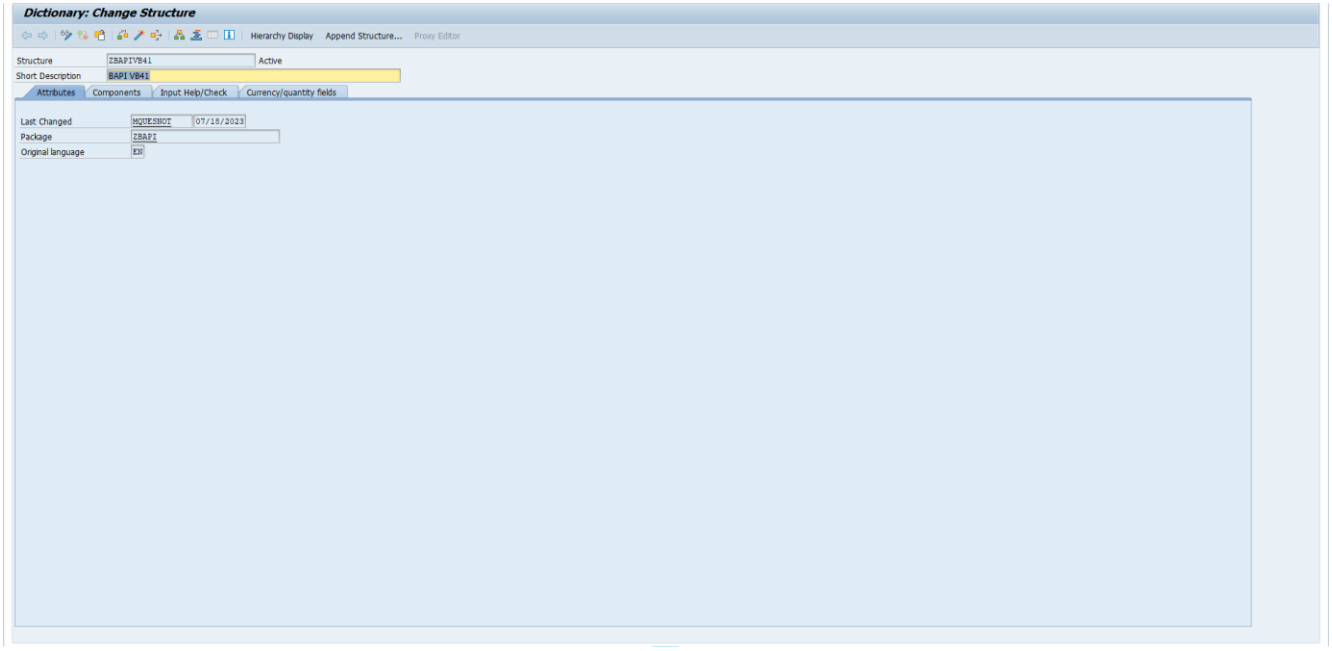
Stage1: Creating a structure in SE11

Step 1: Go to transaction SE11 and create a structure as shown or as per your requirement.


ABAP Dictionary: Initial Screen
Database table
View
Data type
ZBAPIVB41
Type Group
Domain
Search help
Lock object
Display
Change
Create

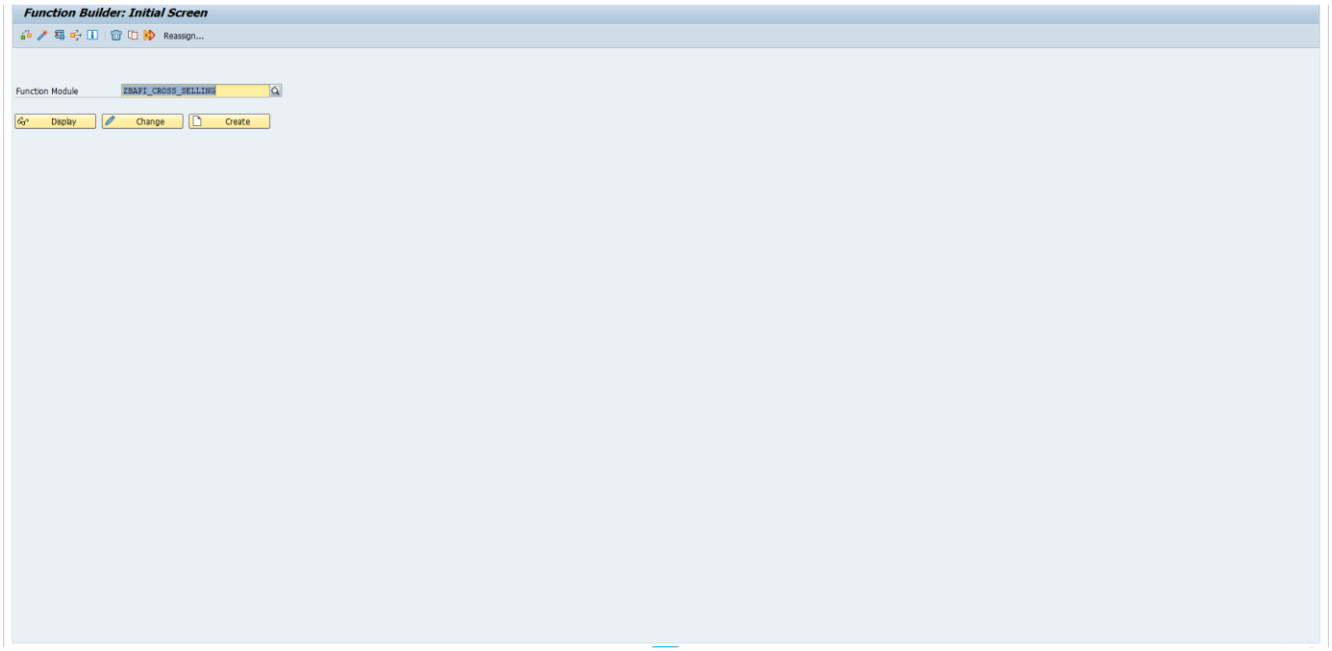

Dictionary: Change Structure
Hierarchy Display
Append Structure...
Proxy Editor
Structure
ZBAPIVB41
Active
Short Description
BAPI VB41
Attributes
Components
Input Help/Check
Currency/quantity fields
Built-In Type
1 / 8
Component
Typing Method
Component Type
Data Type
Length
Deci...
Coordinate
Short Description
Group
KSCHL 1 Types KSCHL CHAR 4 0 0 Condition Type
KOTABNR 1 Types KOTABNR CHAR 3 0 0 Number of the Condition Table
MATWA 1 Types MATWA CHAR 40 0 0 Material Entered
SMATN 1 Types SMATN CHAR 40 0 0 Substitute material
MEINS 1 Types MEINS UNIT 3 0 0 Base Unit of Measure
SUGRD 1 Types SUGRD CHAR 4 0 0 Reason for material substitution
DATAB 1 Types DATAB DATS 8 0 0 Valid-From Date
DATBI 1 Types DATBI DATS 8 0 0 Valid To Date

Do not forget to save it in a **package**. Check the structure (ctrl + F2) and activate (ctrl + F3) the structure.

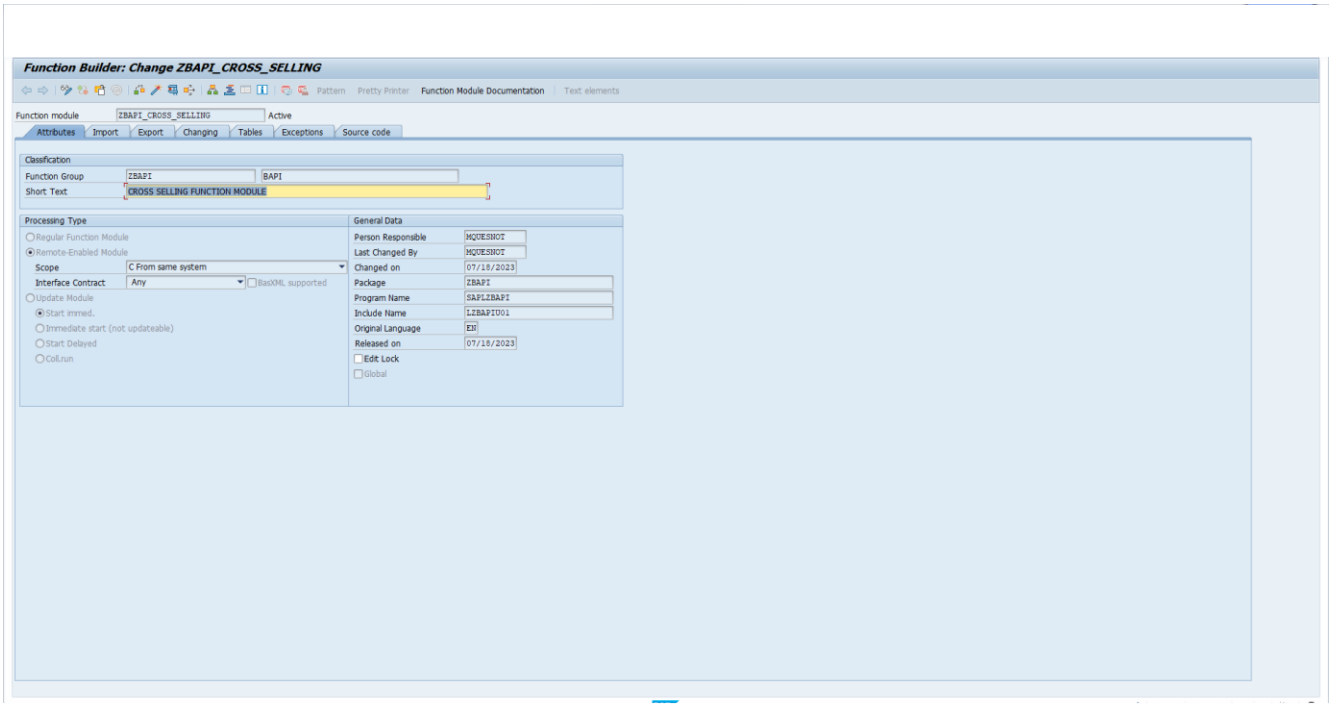


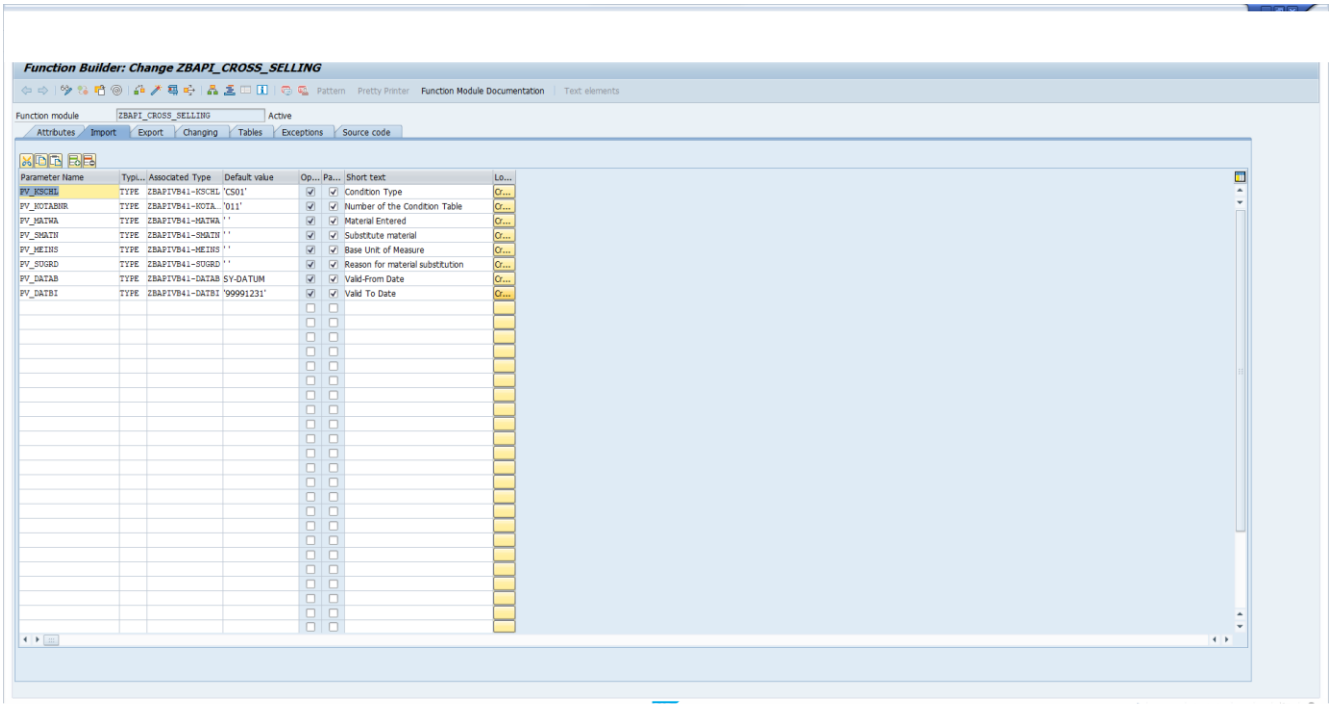
Stage2: Creating the function module in SE37

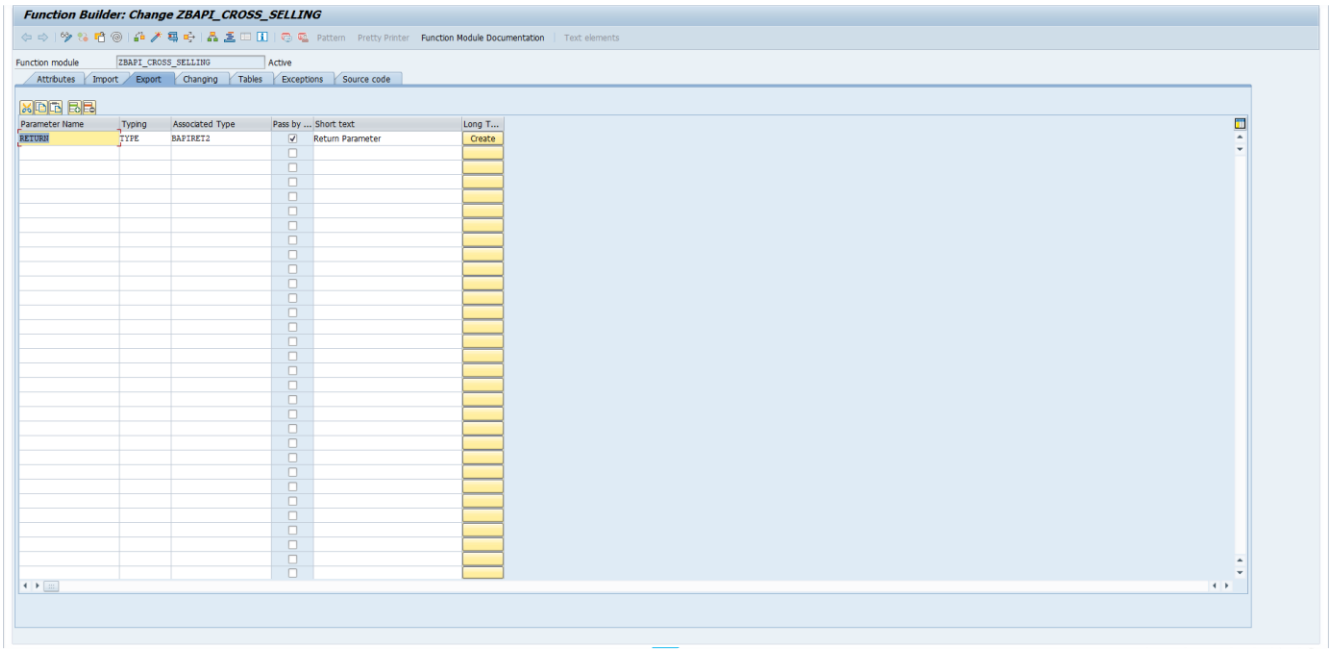
Step 2: Go to transaction SE37 where you create function modules.

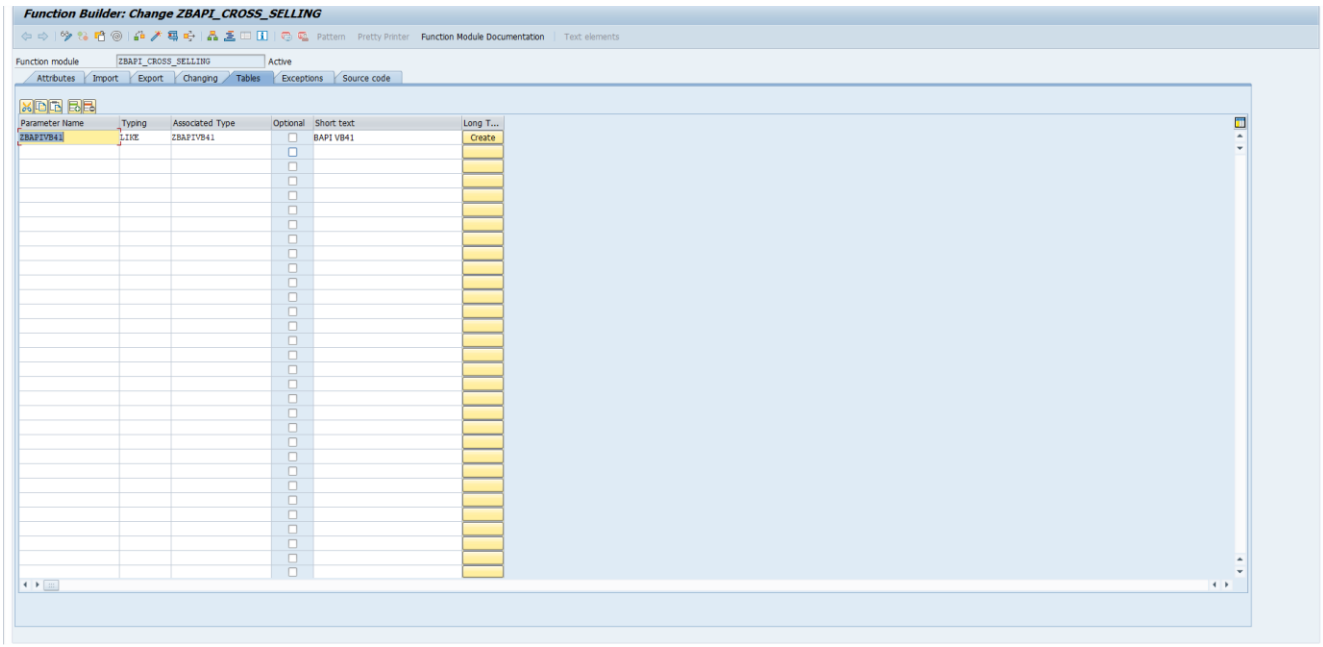


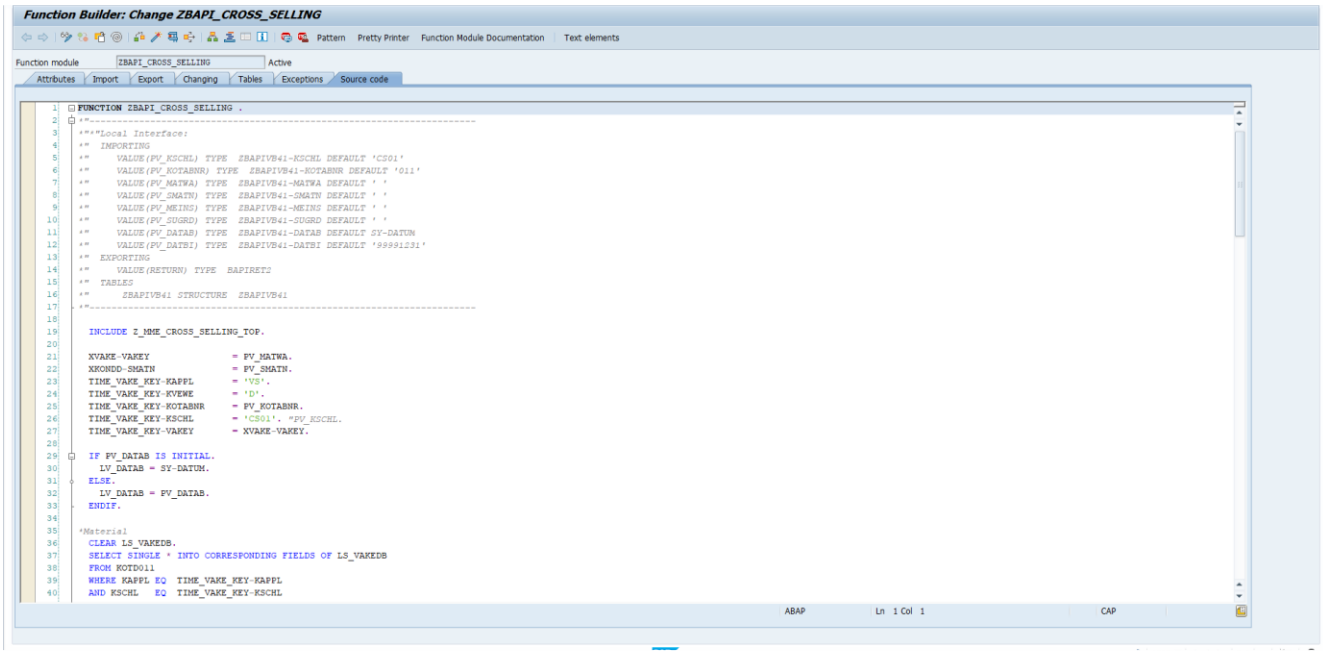
In the attributes tab, make the processing type as ‘Remote-Enabled Module’.




Function Builder: Change ZBAPI_CROSS_SELLING
Pattern
Pretty Printer
Function Module Documentation
Text elements
Function module
ZBAPI_CROSS_SELLING
Active
Attributes
Import
Export
Changing
Tables
Exceptions
Source code
Parameter Name
Typi...
Associated Type
Default value
Op...
Pa...
Short text
Lo...
PV_KSCHL TYPE ZBAPIVB41-KSCHL 'CS01' Condition Type Cr...
PV_KOTABNR TYPE ZBAPIVB41-KOTA... '011' Number of the Condition Table Cr...
PV_MATWA TYPE ZBAPIVB41-MATWA '' Material Entered Cr...
PV_SMATN TYPE ZBAPIVB41-SMATN '' Substitute material Cr...
PV_MEINS TYPE ZBAPIVB41-MEINS '' Base Unit of Measure Cr...
PV_SUGRD TYPE ZBAPIVB41-SUGRD '' Reason for material substitution Cr...
PV_DATAB TYPE ZBAPIVB41-DATAB SY-DATUM Valid-From Date Cr...
PV_DATBI TYPE ZBAPIVB41-DATBI '99991231' Valid To Date Cr...


Function Builder: Change ZBAPI_CROSS_SELLING
Pattern
Pretty Printer
Function Module Documentation
Text elements
Function module
ZBAPI_CROSS_SELLING
Active
Attributes
Import
Export
Changing
Tables
Exceptions
Source code
Parameter Name
Typing
Associated Type
Pass by ...
Short text
Long T...
RETURN TYPE BAPIRET2 Return Parameter Create

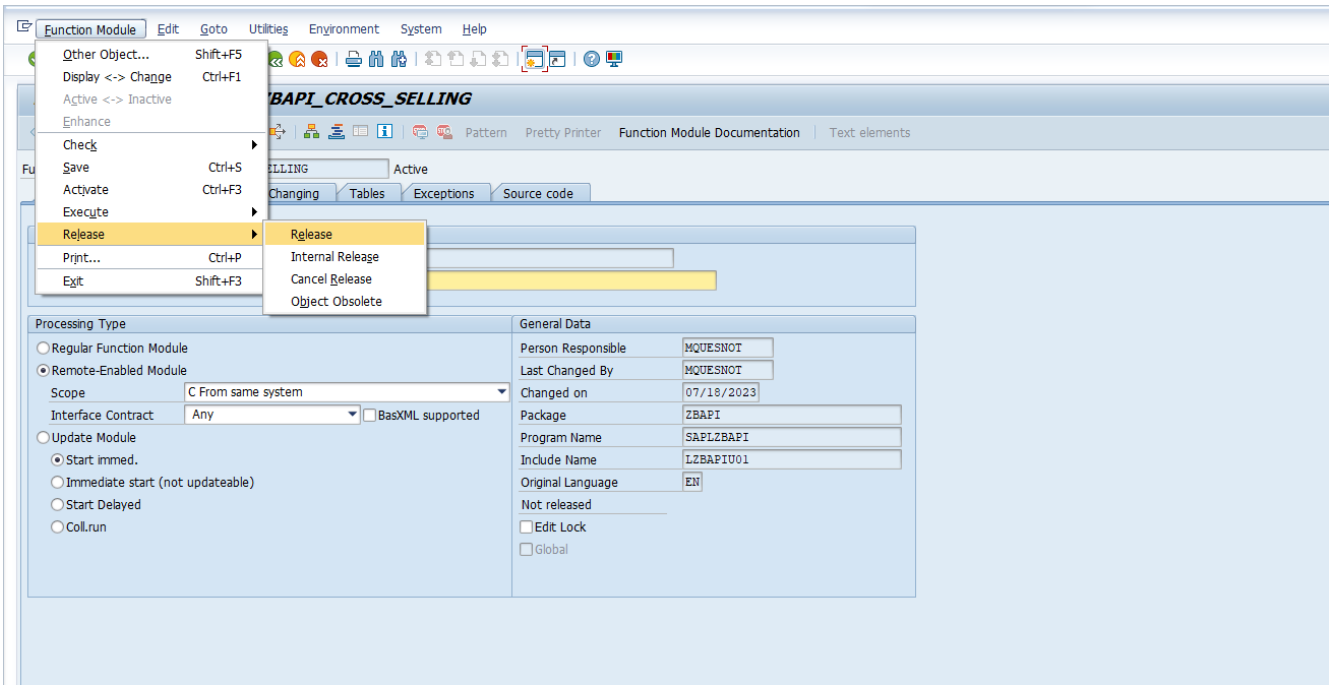

Function Builder: Change ZBAPI_CROSS_SELLING
Pattern
Pretty Printer
Function Module Documentation
Text elements
Function module
ZBAPI_CROSS_SELLING
Active
Attributes
Import
Export
Changing
Tables
Exceptions
Source code
Parameter Name
Typing
Associated Type
Optional
Short text
Long T...
ZBAPIVB41
LIKE
ZBAPIVB41
BAPI VB41
Create

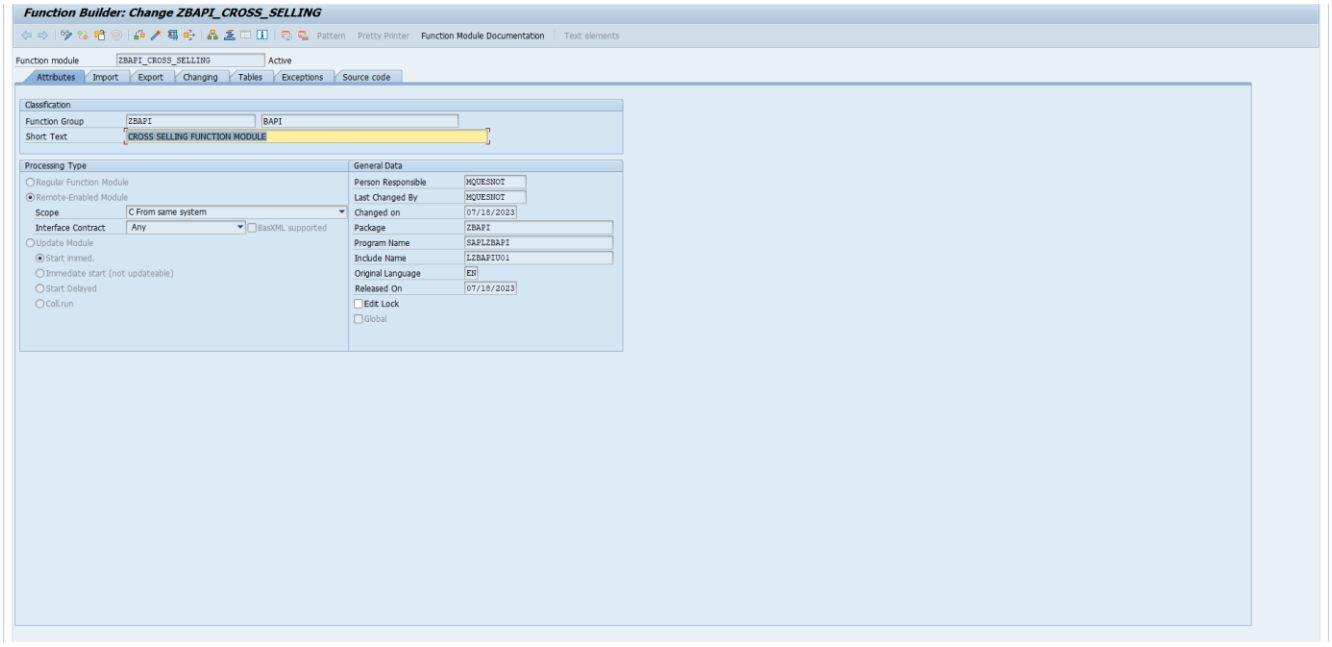

Function Builder: Change ZBAPI_CROSS_SELLING
Pattern
Pretty Printer
Function Module Documentation
Text elements
Function module
ZBAPI_CROSS_SELLING
Active
Attributes
Import
Export
Changing
Tables
Exceptions
Source code
FUNCTION ZBAPI_CROSS_SELLING .
*"--
""Local Interface:
*" IMPORTING
*"   VALUE(PV_KSCHL) TYPE ZBAPIVB41-KSCHL DEFAULT 'CS01'
*"   VALUE(PV_KOTABNR) TYPE ZBAPIVB41-KOTABNR DEFAULT '011'
*"   VALUE(PV_MATWA) TYPE ZBAPIVB41-MATWA DEFAULT ' '
*"   VALUE(PV_SMATN) TYPE ZBAPIVB41-SMATN DEFAULT ' '
*"   VALUE(PV_MEINS) TYPE ZBAPIVB41-MEINS DEFAULT ' '
*"   VALUE(PV_SUGRD) TYPE ZBAPIVB41-SUGRD DEFAULT ' '
*"   VALUE(PV_DATAB) TYPE ZBAPIVB41-DATAB DEFAULT SY-DATUM
*"   VALUE(PV_DATBI) TYPE ZBAPIVB41-DATBI DEFAULT '99991231'
*" EXPORTING
*"   VALUE(RETURN) TYPE BAPIRET2
*" TABLES
*"   ZBAPIVB41 STRUCTURE ZBAPIVB41
*"--
INCLUDE Z_MME_CROSS_SELLING_TOP.
XVAKE-VAKEY = PV_MATWA.
XKONDD-SMATN = PV_SMATN.
TIME_VAKE_KEY-KAPPL = 'V0'.
TIME_VAKE_KEY-KVEWE = 'D'.
TIME_VAKE_KEY-KOTABNR = PV_KOTABNR.
TIME_VAKE_KEY-KSCHL = 'CS01'. "PV_KSCHL.
TIME_VAKE_KEY-VAKEY = XVAKE-VAKEY.
IF PV_DATAB IS INITIAL.
LV_DATAB = SY-DATUM.
ELSE.
LV_DATAB = PV_DATAB.
ENDIF.
*Material
CLEAR LS_VAKEDB.
SELECT SINGLE * INTO CORRESPONDING FIELDS OF LS_VAKEDB
FROM KOTD011
WHERE KAPPL EQ TIME_VAKE_KEY-KAPPL
AND KSCHL EQ TIME_VAKE_KEY-KSCHL
ABAP
Ln 1 Col 1
CAP

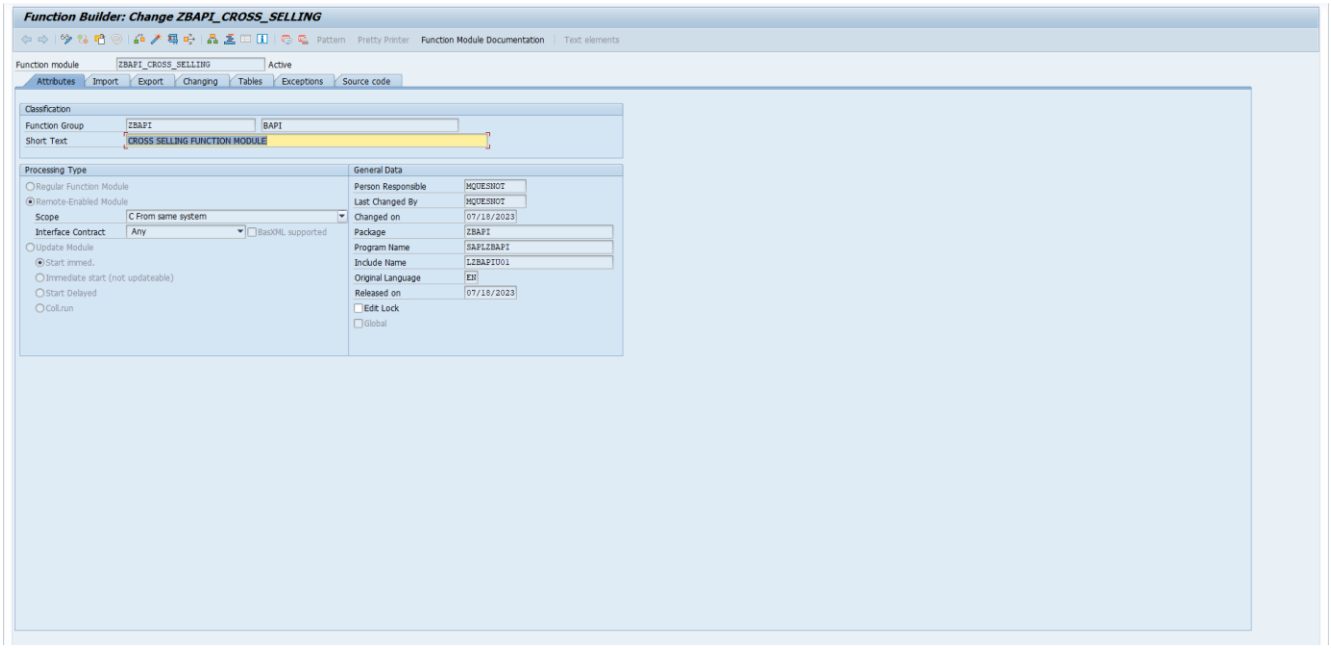
```
FUNCTION ZBAPI_CROSS_SELLING .
*"----------------------------------------------------------------------
*"*"Local Interface:
*"  IMPORTING
*"     VALUE(PV_KSCHL) TYPE  ZBAPIVB41-KSCHL DEFAULT 'CS01'
*"     VALUE(PV_KOTABNR) TYPE  ZBAPIVB41-KOTABNR DEFAULT '011'
*"     VALUE(PV_MATWA) TYPE  ZBAPIVB41-MATWA DEFAULT ' '
*"     VALUE(PV_SMATN) TYPE  ZBAPIVB41-SMATN DEFAULT ' '
*"     VALUE(PV_MEINS) TYPE  ZBAPIVB41-MEINS DEFAULT ' '
*"     VALUE(PV_SUGRD) TYPE  ZBAPIVB41-SUGRD DEFAULT ' '
*"     VALUE(PV_DATAB) TYPE  ZBAPIVB41-DATAB DEFAULT SY-DATUM
*"     VALUE(PV_DATBI) TYPE  ZBAPIVB41-DATBI DEFAULT '99991231'
*"  EXPORTING
*"     VALUE(RETURN) TYPE  BAPIRET2
*"  TABLES
*"      ZBAPIVB41 STRUCTURE  ZBAPIVB41
*"----------------------------------------------------------------------

  INCLUDE Z_MME_CROSS_SELLING_TOP.

  XVAKE-VAKEY               = PV_MATWA.
  XKONDD-SMATN              = PV_SMATN.
  TIME_VAKE_KEY-KAPPL       = 'VS'.
  TIME_VAKE_KEY-KVEWE        = 'D'.
  TIME_VAKE_KEY-KOTABNR     = PV_KOTABNR.
  TIME_VAKE_KEY-KSCHL       = 'CS01'. "PV_KSCHL.
  TIME_VAKE_KEY-VAKEY       = XVAKE-VAKEY.

  IF PV_DATAB IS INITIAL.
    LV_DATAB = SY-DATUM.
  ELSE.
    LV_DATAB = PV_DATAB.
  ENDIF.

*Material
  CLEAR LS_VAKEDB.
  SELECT SINGLE * INTO CORRESPONDING FIELDS OF LS_VAKEDB
  FROM KOTD011
  WHERE KAPPL EQ  TIME_VAKE_KEY-KAPPL
  AND KSCHL   EQ  TIME_VAKE_KEY-KSCHL
  AND MATNR   EQ  PV_MATWA
  AND DATBI   GE  LV_DATAB
  AND DATAB   LE  LV_DATAB.

  IF SY-SUBRC = 0.

    SELECT SINGLE * FROM KONDD WHERE KNUMH = NV_NEW_NUMBER AND SMATN = PV_SMATN.

    IF SY-SUBRC = 0.
      EXIT.
    ELSE.
      NV_NEW_NUMBER = LS_VAKEDB-KNUMH.
    ENDIF.
```

```
  ELSE.

    CALL FUNCTION 'NUMBER_GET_NEXT'
      EXPORTING
        NR_RANGE_NR = '01'
        OBJECT      = NUMMERVORG
      IMPORTING
        NUMBER      = NV_NEW_NUMBER.

  ENDIF.

*Variable Key for Condition Maintenance: Internal Structure
  XVAKE-KVEWE   =  TIME_VAKE_KEY-KVEWE.
  XVAKE-KOTABNR =  TIME_VAKE_KEY-KOTABNR .
  XVAKE-KAPPL   =  TIME_VAKE_KEY-KAPPL.
  XVAKE-KSCHL   =  TIME_VAKE_KEY-KSCHL.
  XVAKE-DATBI   =  PV_DATBI.
  XVAKE-DATAB    =  PV_DATAB.
  XVAKE-KNUMH   =  NV_NEW_NUMBER.
  XVAKE-UPDKZ   = 'I'.
  APPEND  XVAKE.

*Condition Tables: Update Structure
  MOVE-CORRESPONDING XVAKE TO TIME_VAKE_DB.
  TIME_VAKE_DB-XXSTATE    = 'I'.
  TIME_VAKE_DB-XXPROCINFO = 'U'.
  TIME_VAKE_DB-XXDBACTION = 'I'.
  APPEND TIME_VAKE_DB.

*Change Document Structure; Generated by RSSCD000
*Condition Record: Old and New Validity Periods
  XKONDAT-KNUMH  = NV_NEW_NUMBER.
  XKONDAT-DATAN  = SY-DATUM.
  XKONDAT-DATAB = PV_DATAB.
  XKONDAT-DATBI = PV_DATBI.
  XKONDAT-KZ = 'U'.
  APPEND XKONDAT.

*Change Document Structure; Generated by RSSCD000
*Condition Record: Old and New Validity Periods
  YKONDAT-KNUMH  = NV_NEW_NUMBER.
  YKONDAT-DATAN  = PV_DATBI.
  YKONDAT-KZ = 'U'.
  APPEND YKONDAT.

*Material Substitution - Posting Structure
  XKONDD-MANDT = SYST-MANDT.
  XKONDD-KNUMH = NV_NEW_NUMBER.
  IF LS_VAKEDB-KNUMH IS INITIAL.
    XKONDD-UPDKZ = 'I'.
  ELSE.
    XKONDD-UPDKZ = ' '.
  ENDIF.
  APPEND XKONDD.

*Conditions: Additional Mat.for Mat.Determination - Posting
  XKONDDP-MANDT = SYST-MANDT.
```

```
  XKONDDP-KPOSN = 1.
  XKONDDP-KNUMH = NV_NEW_NUMBER.
  XKONDDP-SMATN = XKONDD-SMATN.
  IF LS_VAKEDB-KNUMH IS INITIAL.
    XKONDDP-UPDKZ = ' '.
    XKONDDP-KPOSN = 1.
  ELSE.
    CLEAR LV_COUNT.
    SELECT COUNT(*) INTO LV_COUNT FROM KONDD WHERE KNUMH = NV_NEW_NUMBER.
    XKONDDP-UPDKZ = 'I'.
    XKONDDP-KPOSN = LV_COUNT + 1.
    CLEAR : TIME_VAKE_DB, XKONDAT, YKONDAT.
  ENDIF.
  APPEND XKONDDP.

  IF LS_VAKEDB-KNUMH IS INITIAL.
* Condition Maintenance - General
    CALL FUNCTION 'RV_KONDITION_SICHERN' IN UPDATE TASK
      TABLES
        VAKEDB_TAB = TIME_VAKE_DB.
  ENDIF.

  BOOLE_T = 'X'.

*Material substitution - update function
  CALL FUNCTION 'RV_MAT_SUB_SAVE' IN UPDATE TASK
    IMPORTING
      USE_OUTBOUND_CALL = BOOLE_T
      CD_KVEWE          = TIME_VAKE_KEY-KVEWE
      CD_KOTABNR        = TIME_VAKE_KEY-KOTABNR
      CD_KAPPL          = TIME_VAKE_KEY-KAPPL
      CD_KSCHL          = TIME_VAKE_KEY-KSCHL
    TABLES
      X_KONDDVB         = XKONDD
      Y_KONDDVB         = YKONDD
      X_KONDDPVB        = XKONDDP
      Y_KONDDPVB        = YKONDDP
      X_KONDATVB        = XKONDAT
      Y_KONDATVB        = YKONDAT
      DB_TIME           = TIME_VAKE_DB
      DB_XVAKE          = XVAKE.

  CALL FUNCTION 'BAPI_TRANSACTION_COMMIT'
* EXPORTING
*   WAIT          =
* IMPORTING
*   RETURN        =
    .

*Material Substitution - Data Division
  SELECT SINGLE * FROM KONDD WHERE KNUMH = NV_NEW_NUMBER AND SMATN = PV_SMATN.


ENDFUNCTION.
```

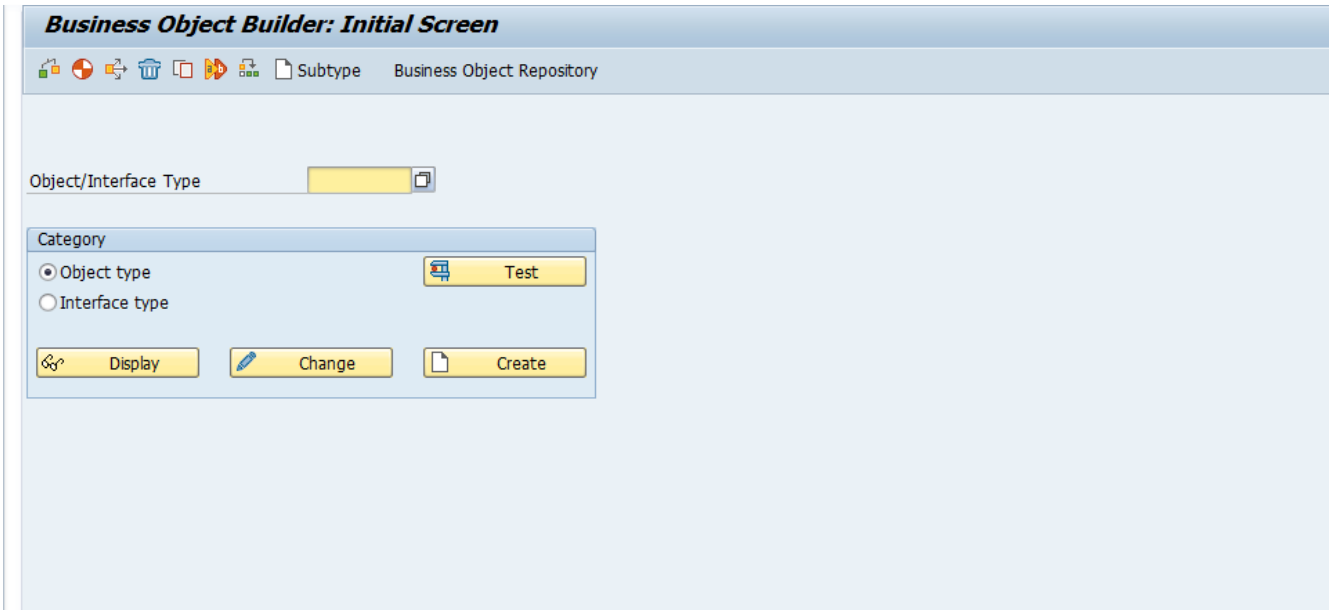

Function Module
Edit
Goto
Utilities
Environment
System
Help
Other Object... Shift+F5
Display <-> Change Ctrl+F1
Active <-> Inactive
Enhance
Check
Save Ctrl+S
Activate Ctrl+F3
Execute
Release
Print... Ctrl+P
Exit Shift+F3
Release
Internal Release
Cancel Release
Object Obsolete
BAPI_CROSS_SELLING
Pattern
Pretty Printer
Function Module Documentation
Text elements
Active
Changing
Tables
Exceptions
Source code
Processing Type
Regular Function Module
Remote-Enabled Module
Scope
C From same system
Interface Contract
Any
BasXML supported
Update Module
Start immed.
Immediate start (not updateable)
Start Delayed
Coll.run
General Data
Person Responsible
MQUESNOT
Last Changed By
MQUESNOT
Changed on
07/18/2023
Package
ZBAPI
Program Name
SAPLZBAPI
Include Name
LZBAPIU01
Original Language
EN
Not released
Edit Lock
Global

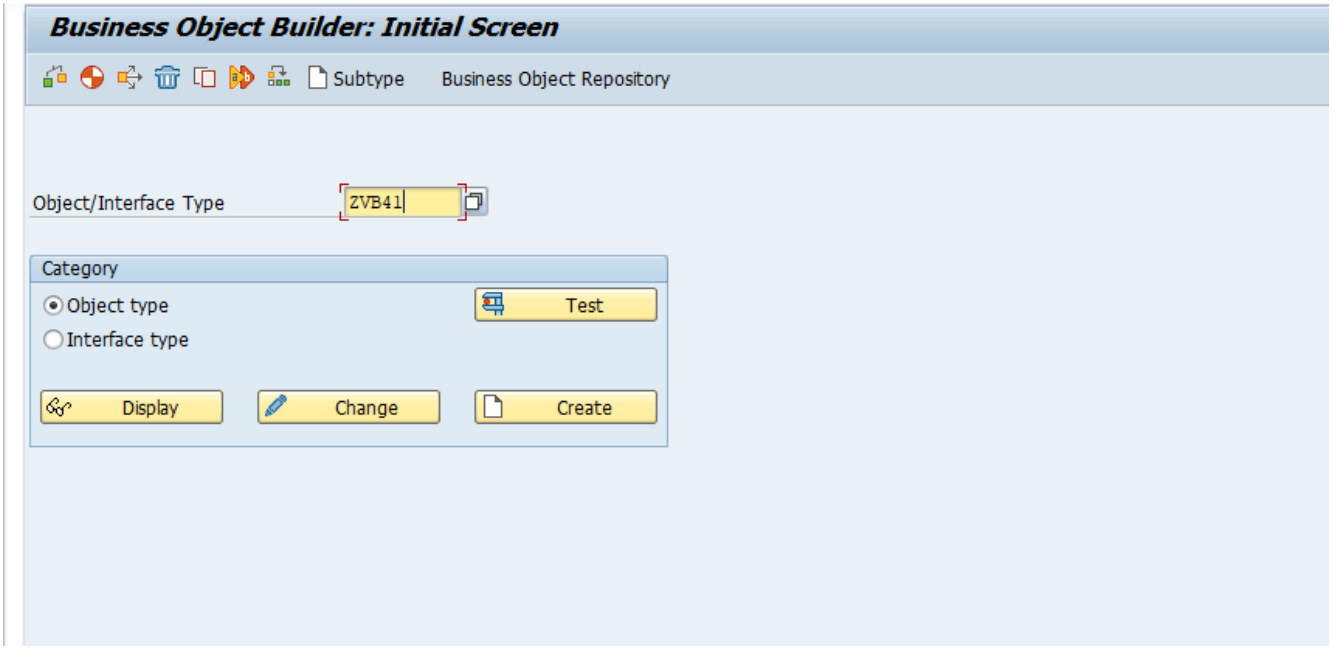

Function Builder: Change ZBAPI_CROSS_SELLING
Function module
ZBAPI_CROSS_SELLING
Active
Attributes
Import
Export
Changing
Tables
Exceptions
Source code
Classification
Function Group
ZBAPI
BAPI
Short Text
CROSS SELLING FUNCTION MODULE
Processing Type
Regular Function Module
Remote-Enabled Module
Scope
C From same system
Interface Contract
Any
BasXML supported
Update Module
Start immed.
Immediate start (not updateable)
Start Delayed
Coll.run
General Data
Person Responsible
MQUESNOT
Last Changed By
MQUESNOT
Changed on
07/18/2023
Package
ZBAPI
Program Name
SAPLZBAPI
Include Name
LZBAPIU01
Original Language
EN
Released On
07/18/2023
Edit Lock
Global

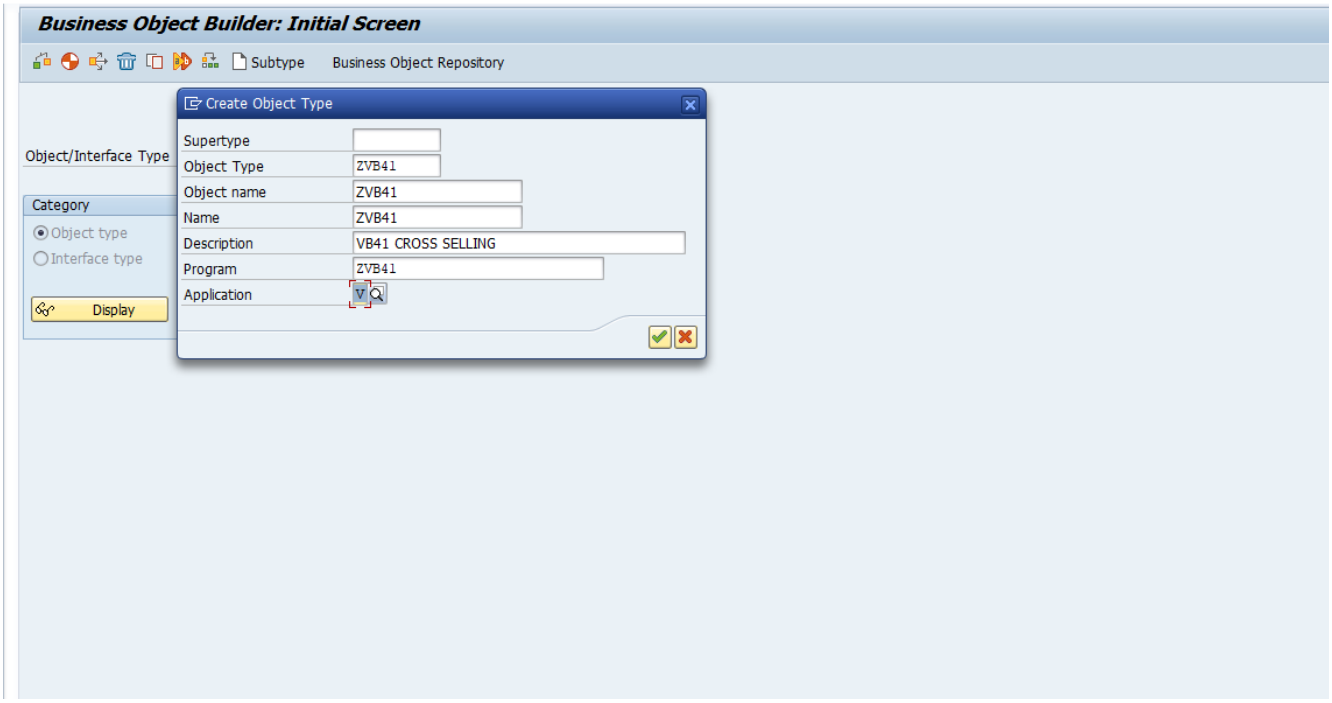


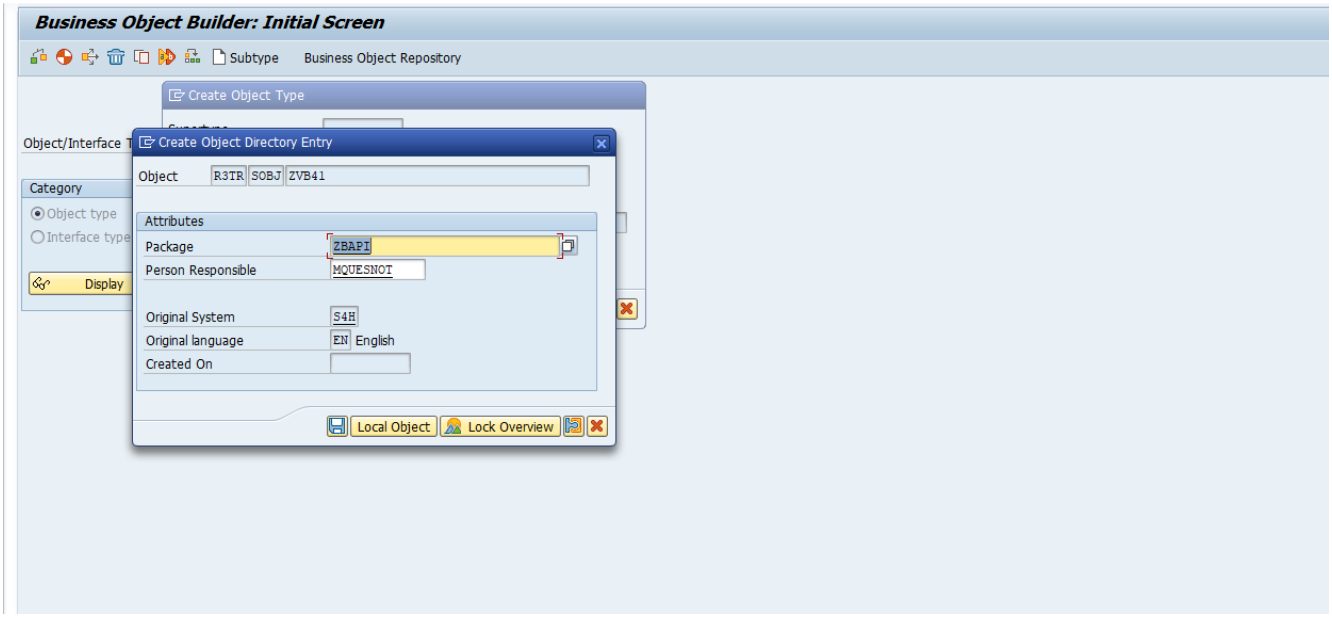
How to create a BAPI

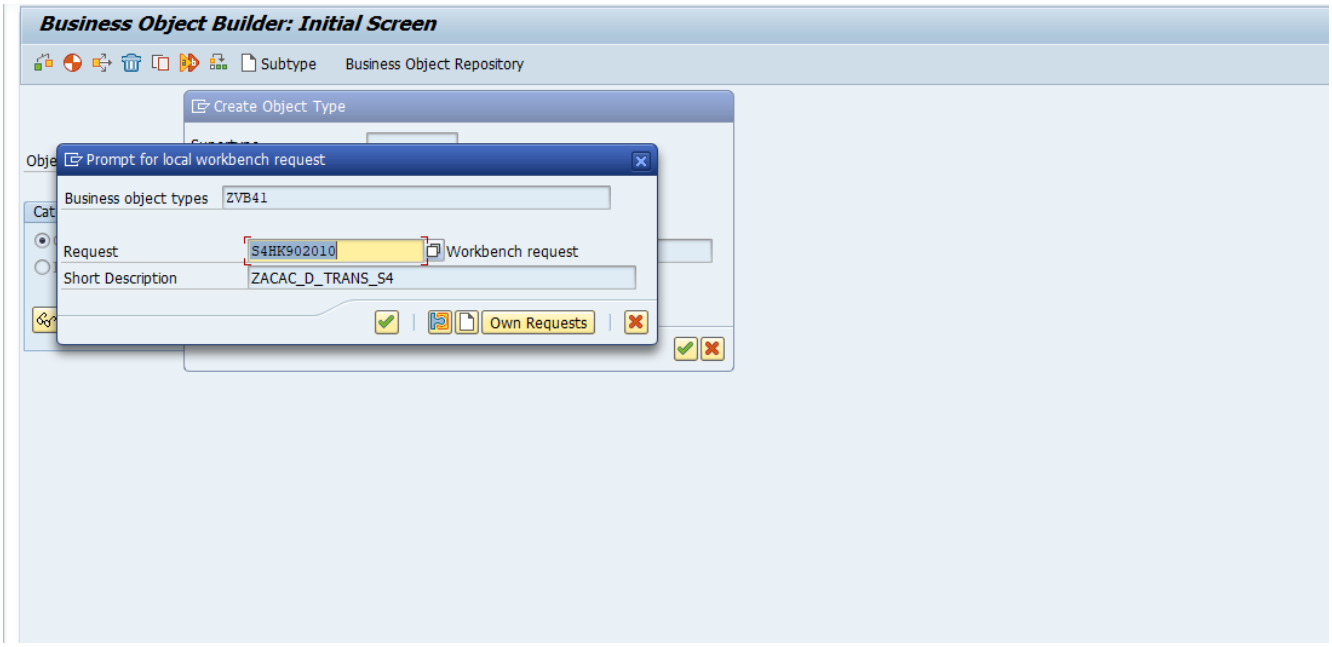
Step 1.Go to transaction swo1 (Tools->Business Framework -> BAPI Development ->Business Object builder) .Select the business object, according to the functional requirement for which the BAPI is being created.

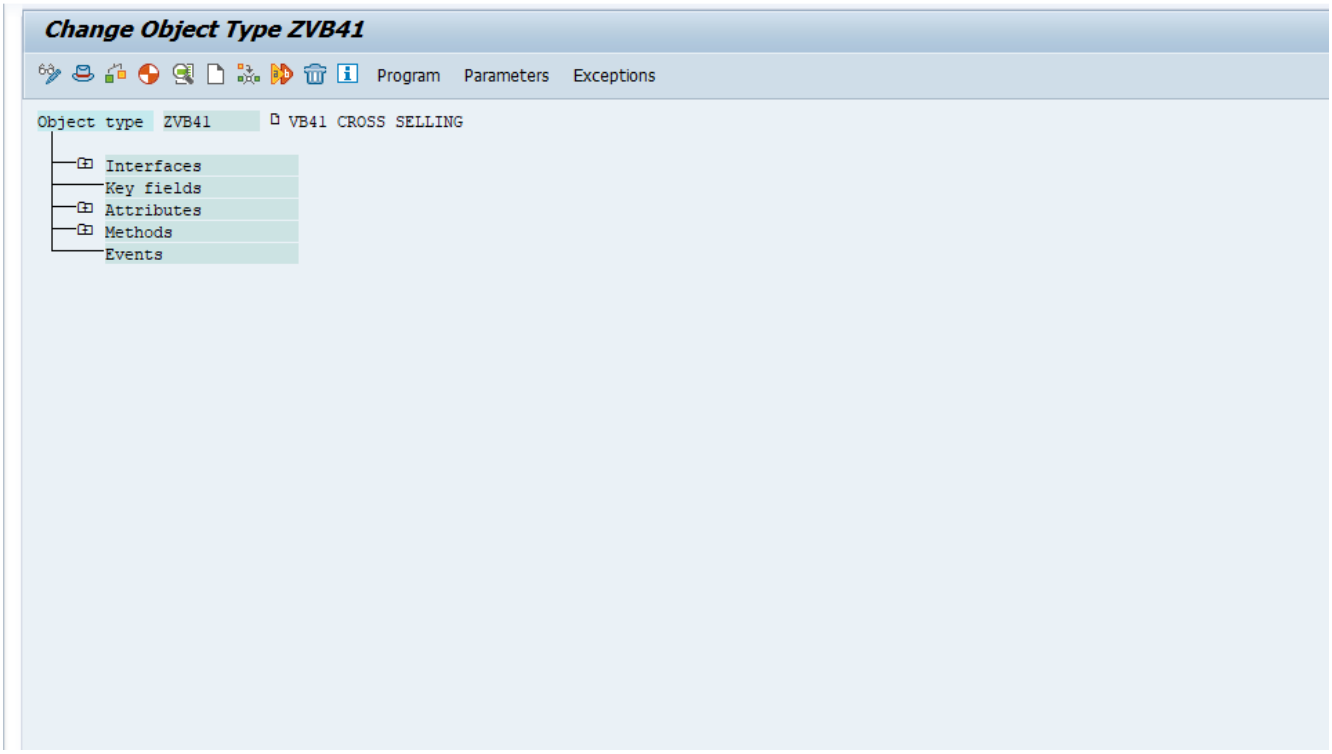

Business Object Builder: Initial Screen
Subtype
Business Object Repository
Object/Interface Type
Category
Object type
Interface type
Test
Display
Change
Create


Business Object Builder: Initial Screen
Subtype
Business Object Repository
Object/Interface Type
ZVB41
Category
Object type
Interface type
Test
Display
Change
Create

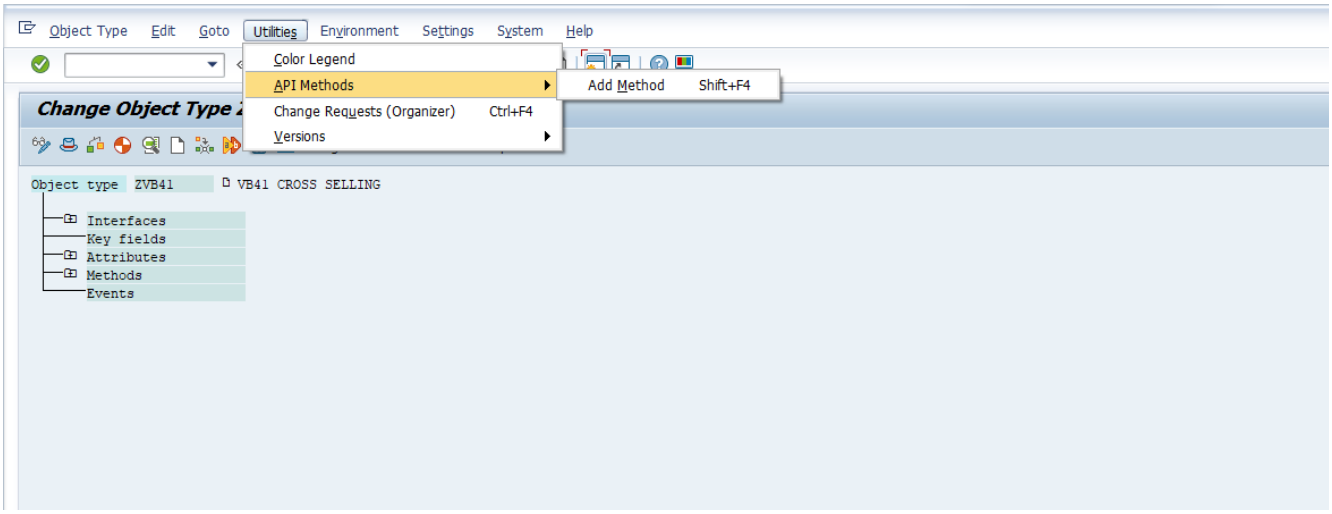

Business Object Builder: Initial Screen
Subtype
Business Object Repository
Create Object Type
Supertype
Object Type
ZVB41
Object name
ZVB41
Name
ZVB41
Description
VB41 CROSS SELLING
Program
ZVB41
Application
V
Object/Interface Type
Category
Object type
Interface type
Display

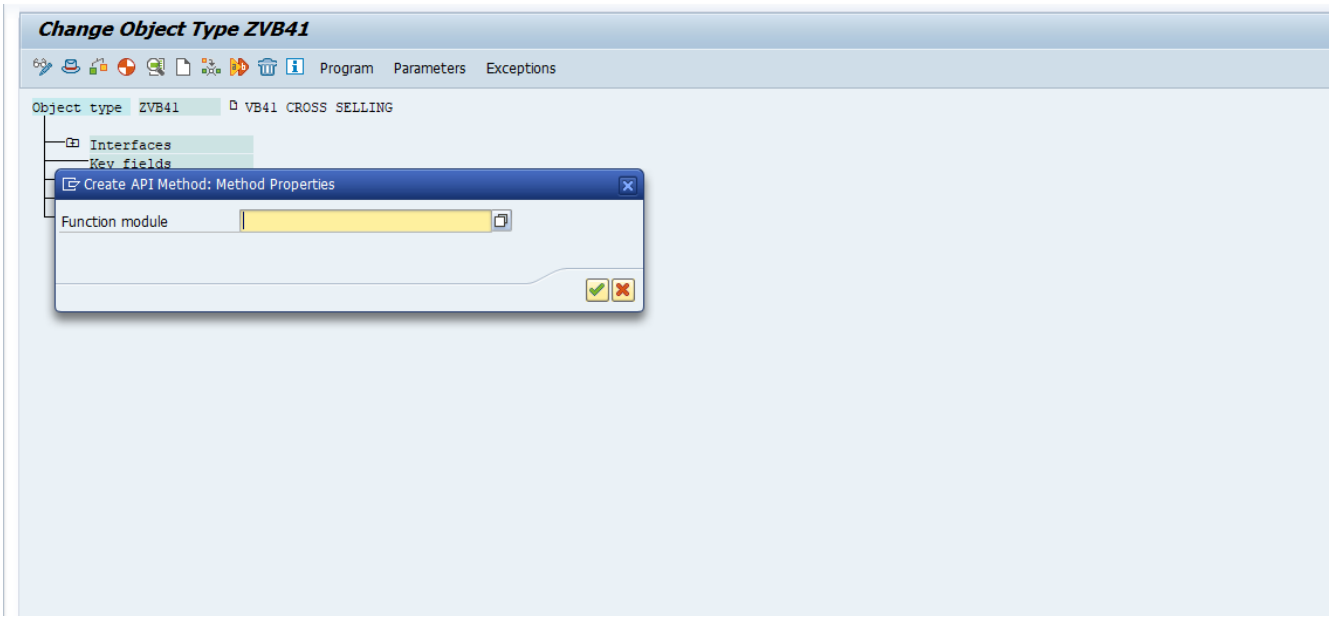

Business Object Builder: Initial Screen
Subtype
Business Object Repository
Create Object Type
Create Object Directory Entry
Object
R3TR
SOBJ
ZVB41
Attributes
Package
ZBAPI
Person Responsible
MQUESNOI
Original System
S4H
Original language
EN
English
Created On
Local Object
Lock Overview
Category
Object type
Interface type
Display

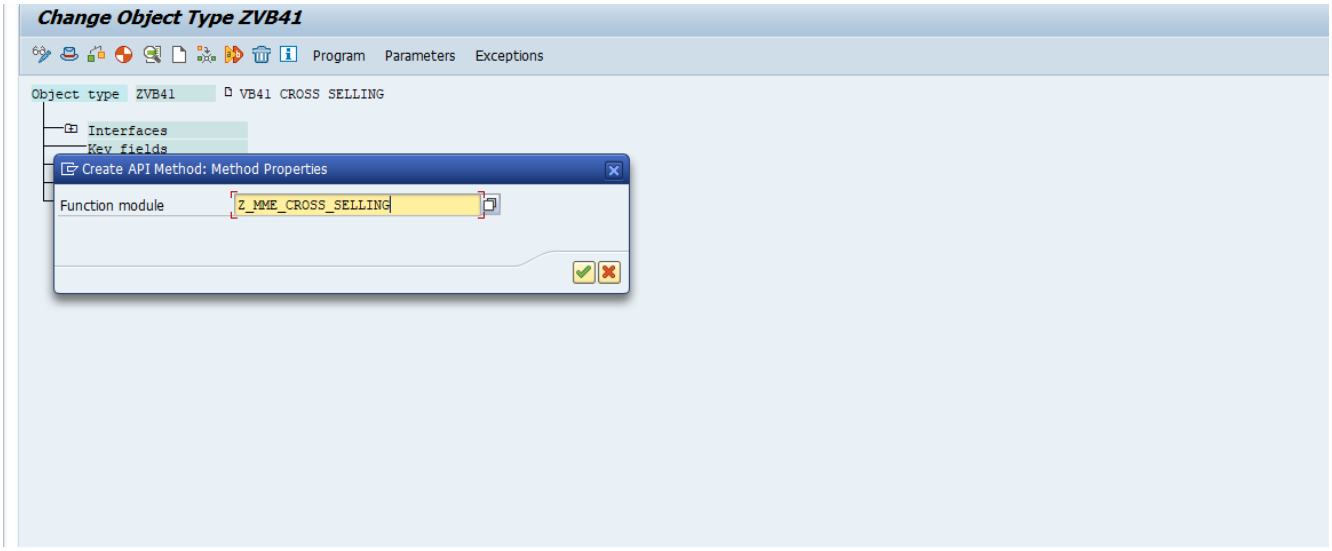

Business Object Builder: Initial Screen
Subtype
Business Object Repository
Create Object Type
Prompt for local workbench request
Business object types
ZVB41
Request
S4HK902010
Workbench request
Short Description
ZACAC_D_TRANS_S4
Own Requests


Change Object Type ZVB41
Program
Parameters
Exceptions
Object type
ZVB41
VB41 CROSS SELLING
Interfaces
Key fields
Attributes
Methods
Events

Step2. Open the business object in change mode. Then Select Utilities ->API Methods ->Add method. Then enter the name of the function module and select Continue.

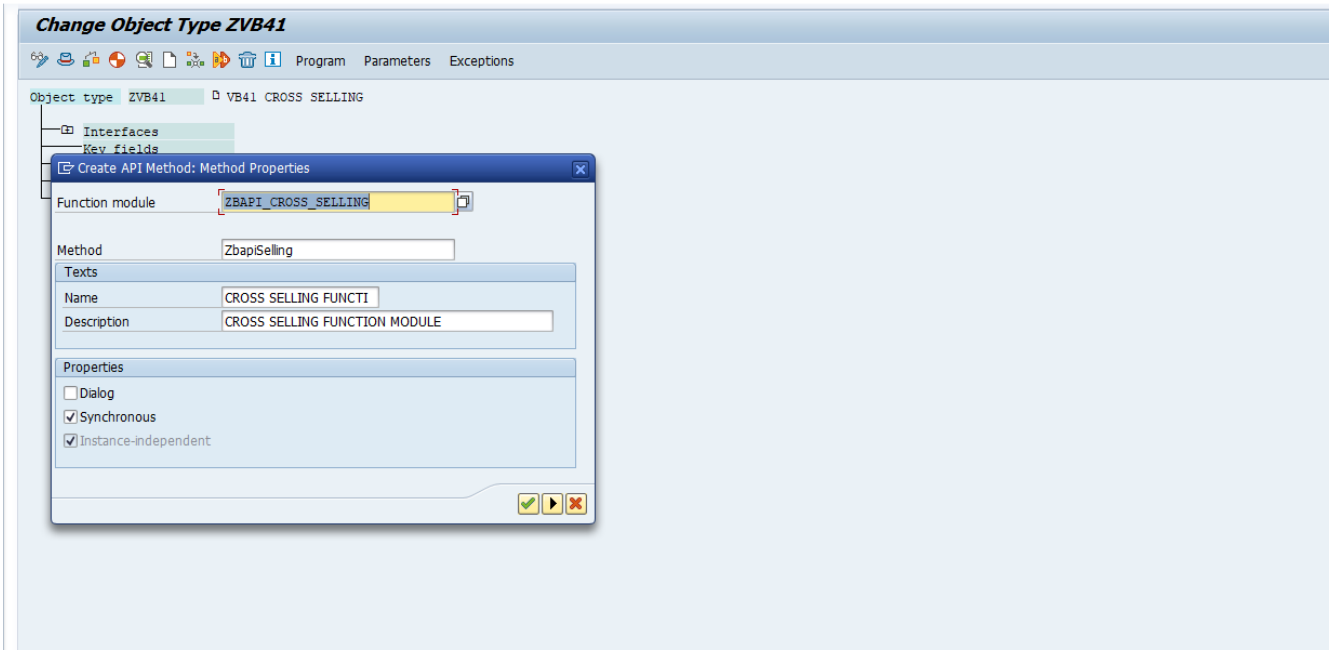


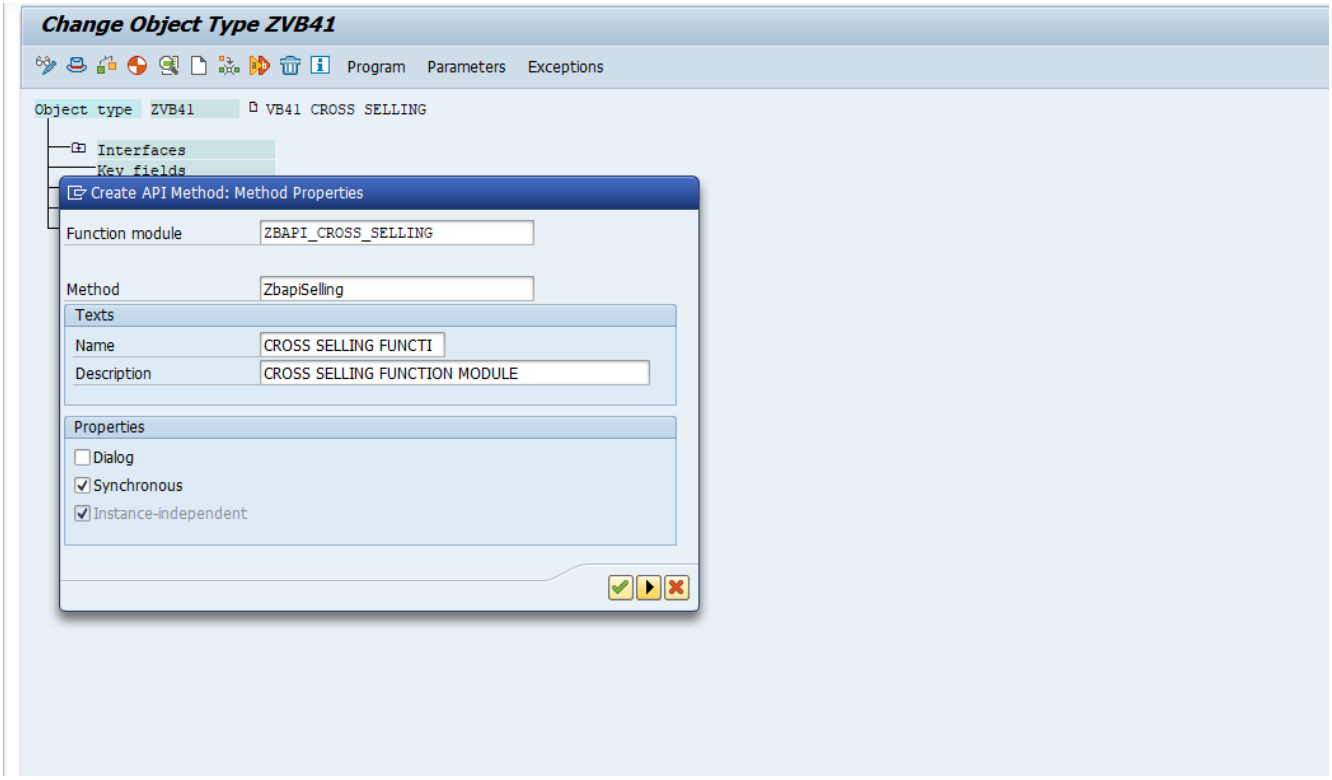


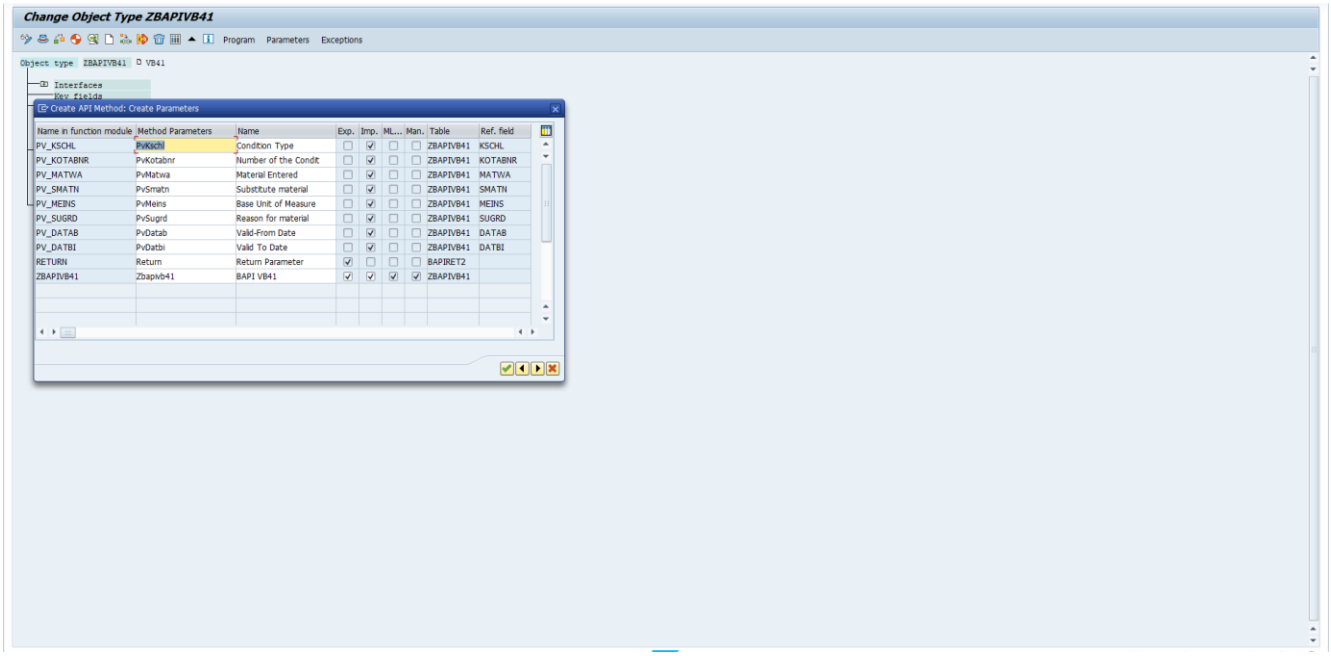


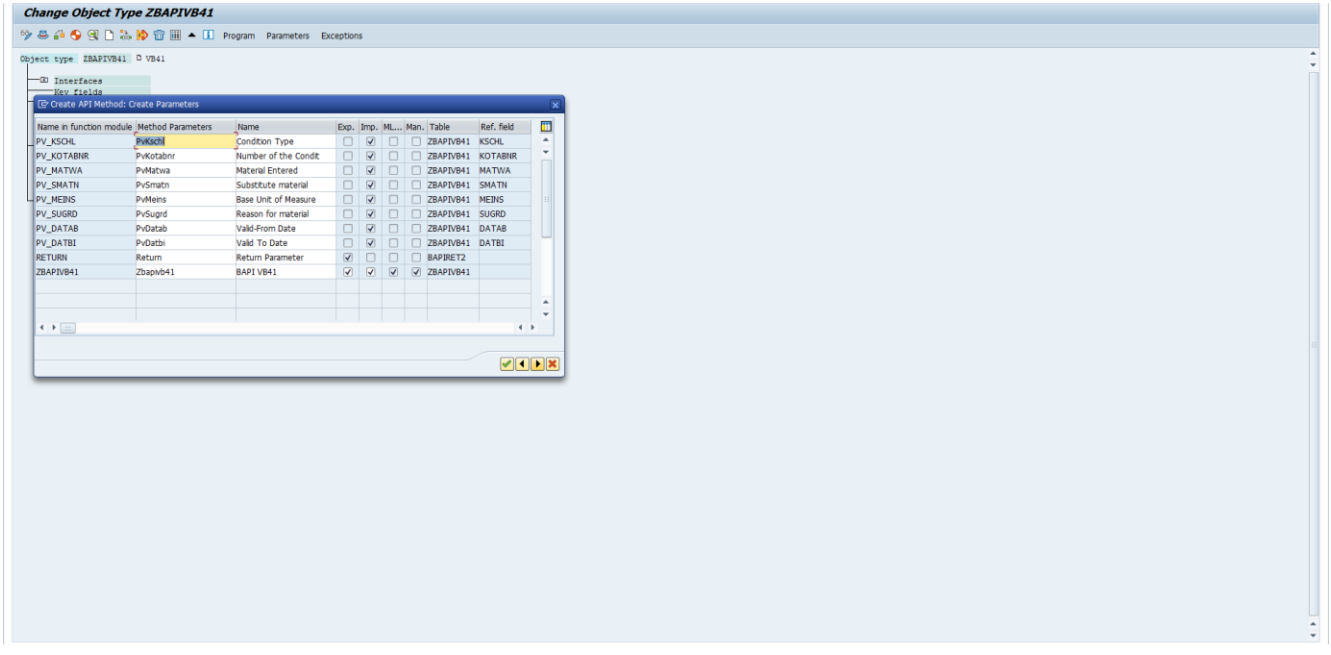
Step 3.In the next dialog box, following information needs to be specified :

- Method : Suggest an appropriate name for the method,
- Texts : Enter description for the BAPI,
- Radio buttons : Dialog, Synchronous, Instance-independent . BAPI ‘s are usually implemented synchronously.

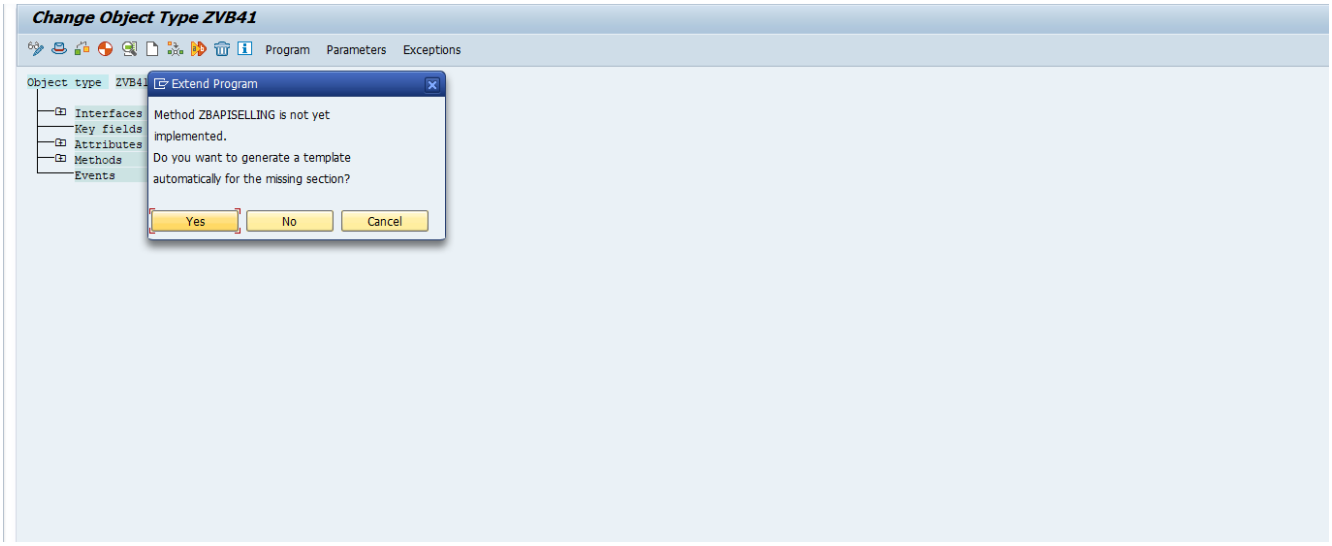

Change Object Type ZVB41
Program
Parameters
Exceptions
Object type ZVB41
VB41 CROSS SELLING
Interfaces
Key fields
Create API Method: Method Properties
Function module
ZBAPI_CROSS_SELLING
Method
ZbapiSelling
Texts
Name
CROSS SELLING FUNCTI
Description
CROSS SELLING FUNCTION MODULE
Properties
Dialog
Synchronous
Instance-independent

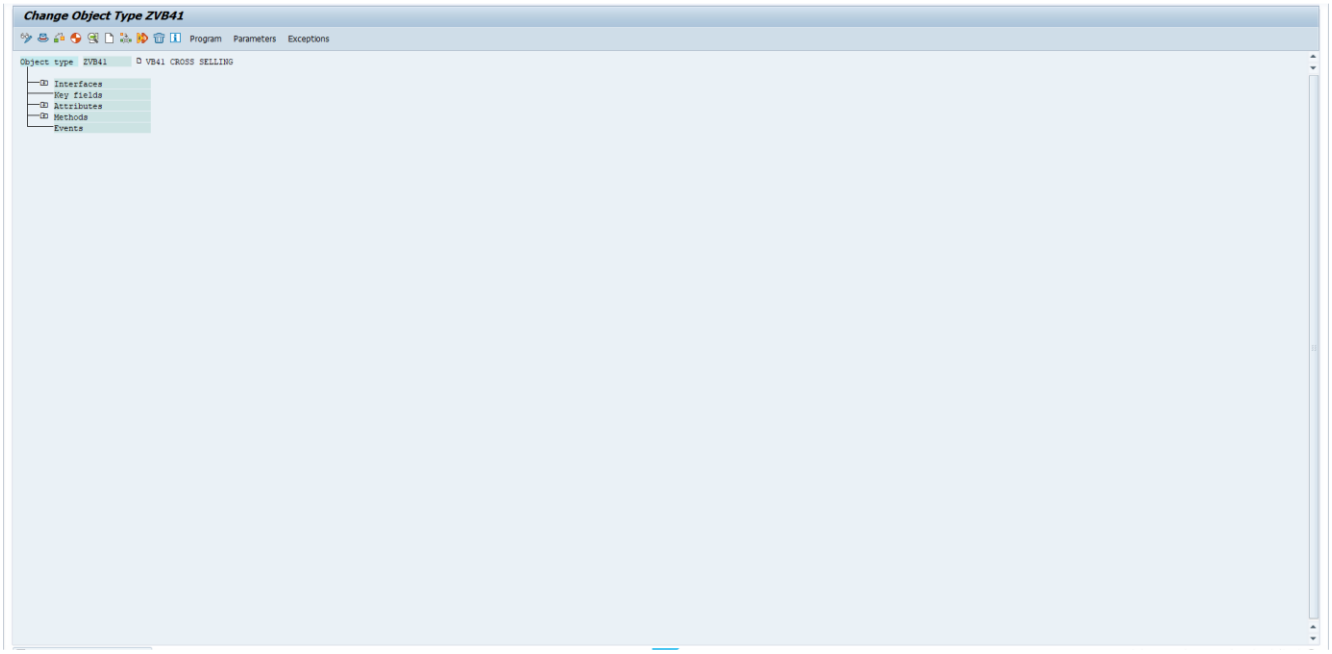

Change Object Type ZVB41
Program
Parameters
Exceptions
Object type ZVB41
VB41 CROSS SELLING
Interfaces
Key fields
Create API Method: Method Properties
Function module
ZBAPI_CROSS_SELLING
Method
ZbapiSelling
Texts
Name
CROSS SELLING FUNCTI
Description
CROSS SELLING FUNCTION MODULE
Properties
Dialog
Synchronous
Instance-independent

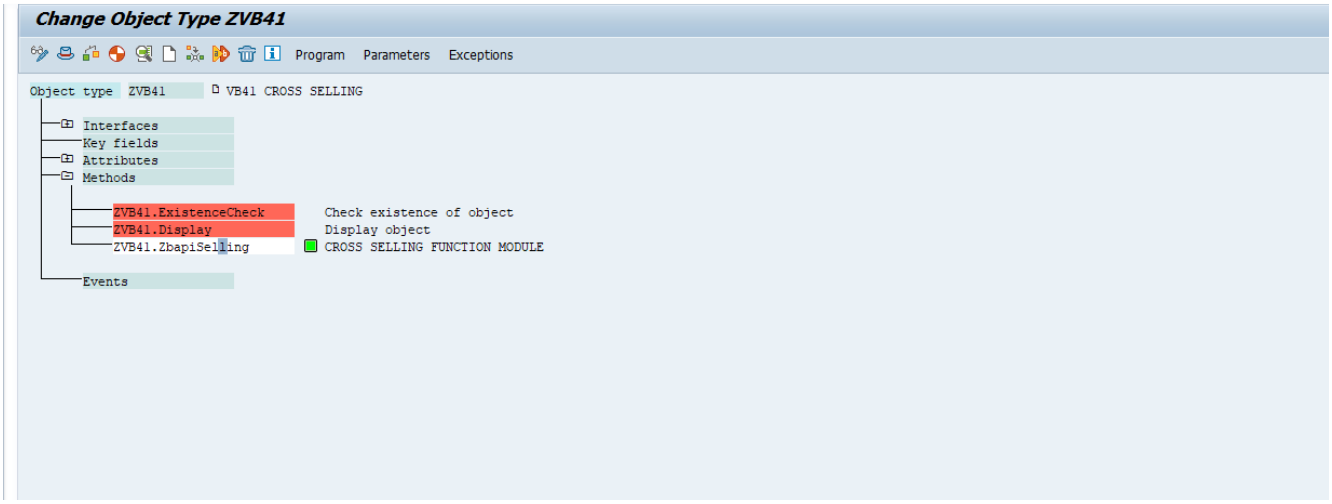

Change Object Type ZBAPIVB41
Program
Parameters
Exceptions
Object type ZBAPIVB41 VB41
Interfaces
Key fields
Create API Method: Create Parameters
Name in function module | Method Parameters | Name | Exp. | Imp. | ML... | Man. | Table | Ref. field
PV_KSCHL | PvKschl | Condition Type | ZBAPIVB41 | KSCHL
PV_KOTABNR | PvKotabnr | Number of the Condit | ZBAPIVB41 | KOTABNR
PV_MATWA | PvMatwa | Material Entered | ZBAPIVB41 | MATWA
PV_SMATN | PvSmatn | Substitute material | ZBAPIVB41 | SMATN
PV_MEINS | PvMeins | Base Unit of Measure | ZBAPIVB41 | MEINS
PV_SUGRD | PvSugrd | Reason for material | ZBAPIVB41 | SUGRD
PV_DATAB | PvDatab | Valid-From Date | ZBAPIVB41 | DATAB
PV_DATBI | PvDatbi | Valid To Date | ZBAPIVB41 | DATBI
RETURN | Return | Return Parameter | BAPIRET2
ZBAPIVB41 | Zbapivb41 | BAPI VB41 | ZBAPIVB41

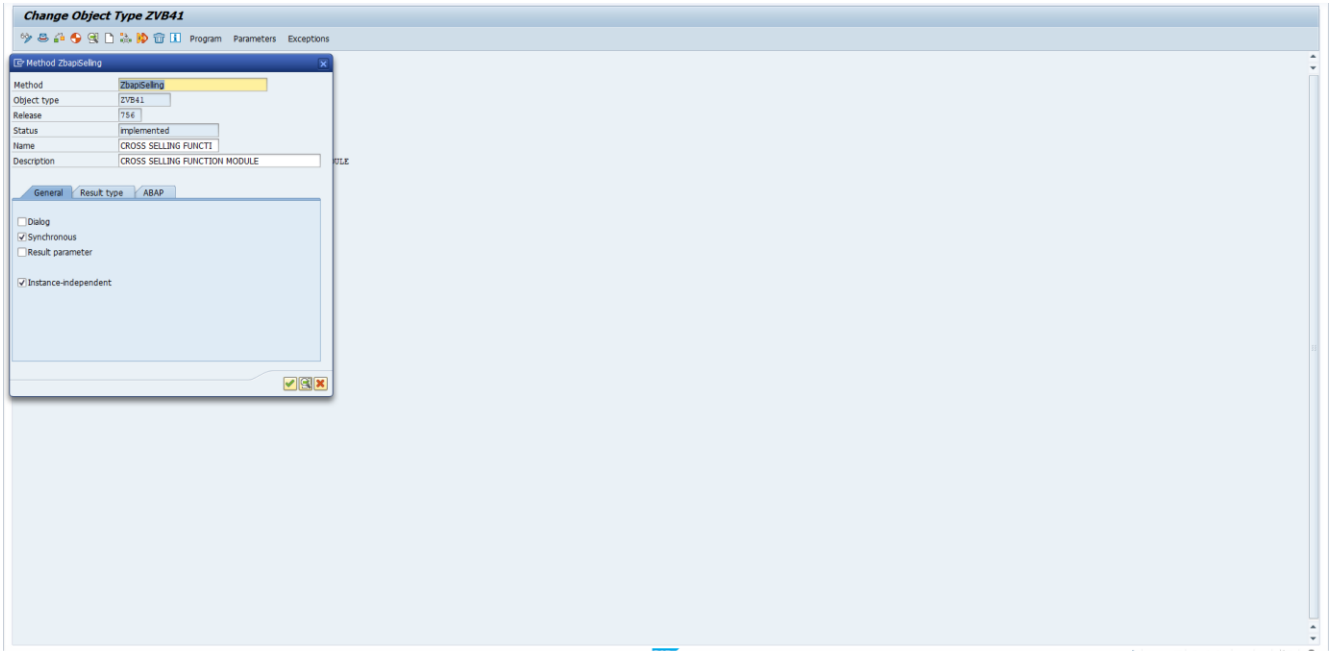

Change Object Type ZBAPIVB41
Program
Parameters
Exceptions
Object type ZBAPIVB41 VB41
Interfaces
Key fields
Create API Method: Create Parameters
Name in function module | Method Parameters | Name | Exp. | Imp. | ML... | Man. | Table | Ref. field
PV_KSCHL | PvKschl | Condition Type | ZBAPIVB41 | KSCHL
PV_KOTABNR | PvKotabnr | Number of the Condit | ZBAPIVB41 | KOTABNR
PV_MATWA | PvMatwa | Material Entered | ZBAPIVB41 | MATWA
PV_SMATN | PvSmatn | Substitute material | ZBAPIVB41 | SMATN
PV_MEINS | PvMeins | Base Unit of Measure | ZBAPIVB41 | MEINS
PV_SUGRD | PvSugrd | Reason for material | ZBAPIVB41 | SUGRD
PV_DATAB | PvDatab | Valid-From Date | ZBAPIVB41 | DATAB
PV_DATBI | PvDatbi | Valid To Date | ZBAPIVB41 | DATBI
RETURN | Return | Return Parameter | BAPIRET2
ZBAPIVB41 | Zbapivb41 | BAPI VB41 | ZBAPIVB41

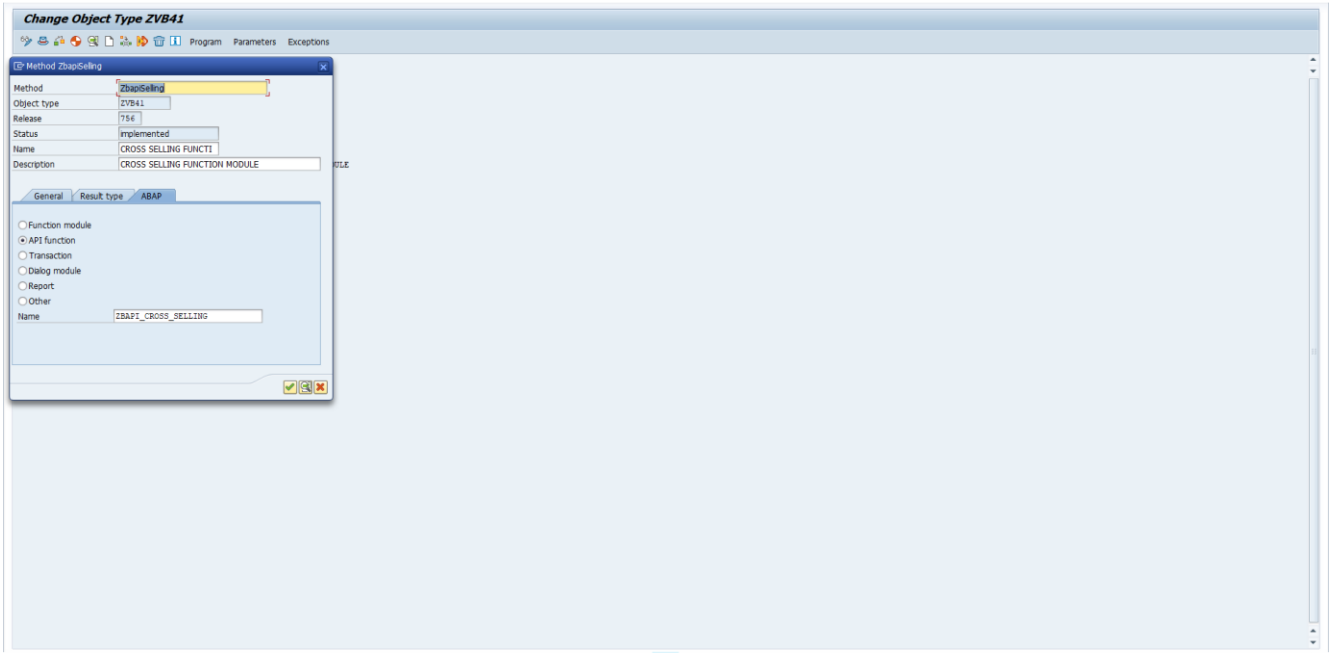
Step4. To create the method select Yes in the next dialog box.



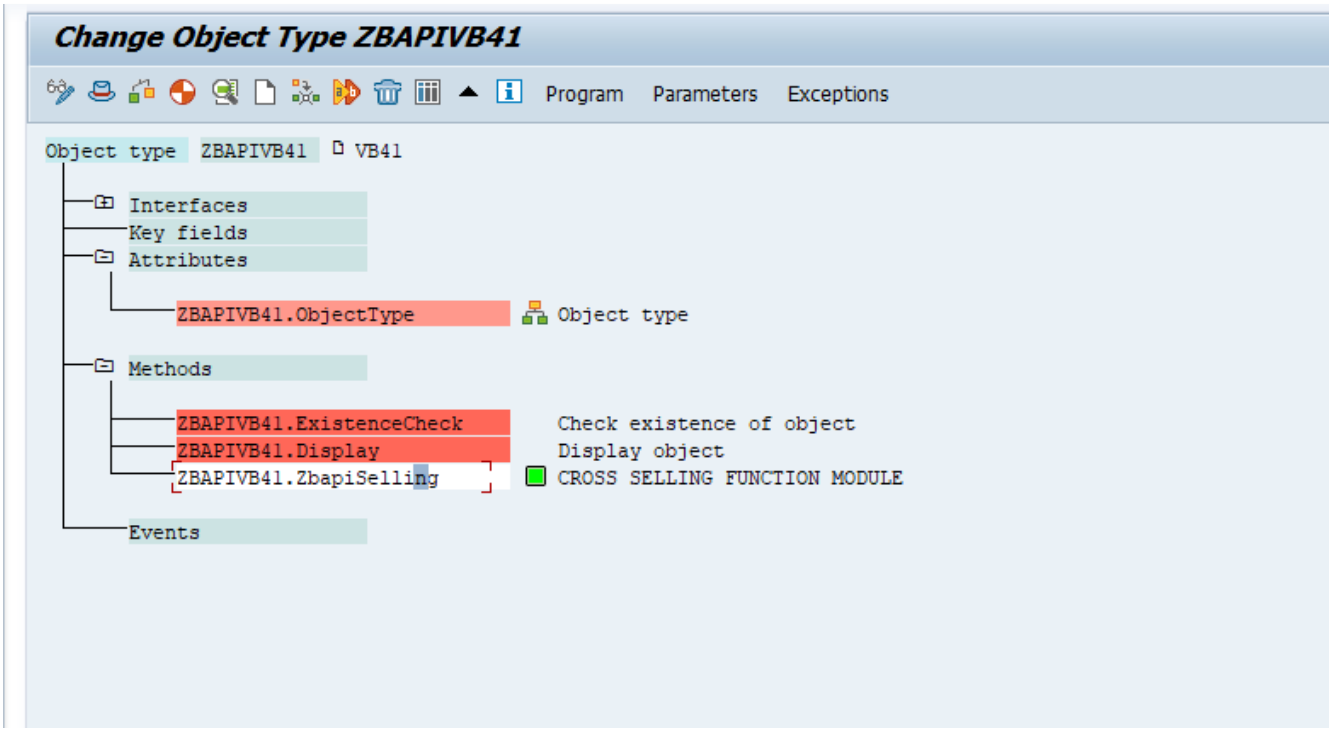


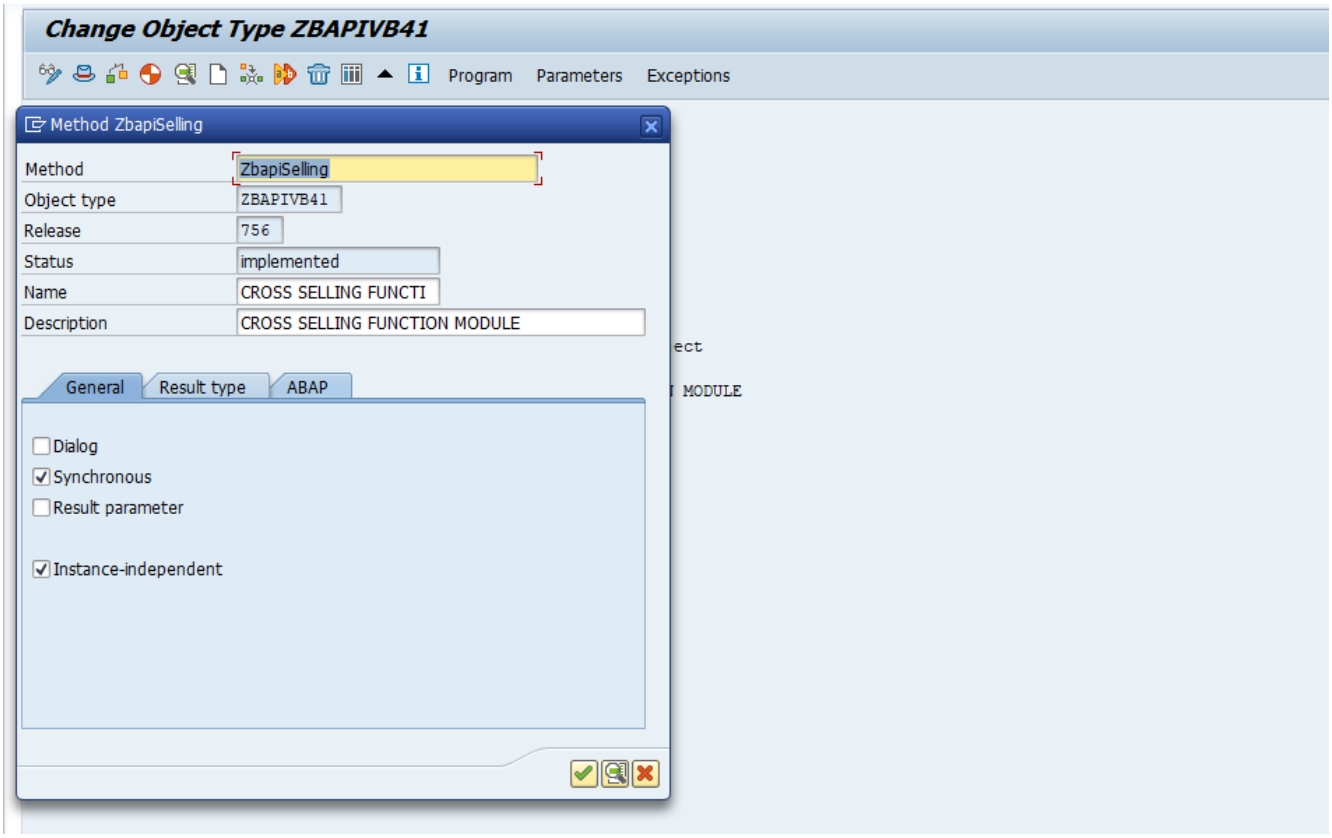

Change Object Type ZVB41
Program
Parameters
Exceptions
Object type ZVB41
VB41 CROSS SELLING
Interfaces
Key fields
Attributes
Methods
ZVB41.ExistenceCheck
Check existence of object
ZVB41.Display
Display object
ZVB41.ZbapiSelling
CROSS SELLING FUNCTION MODULE
Events

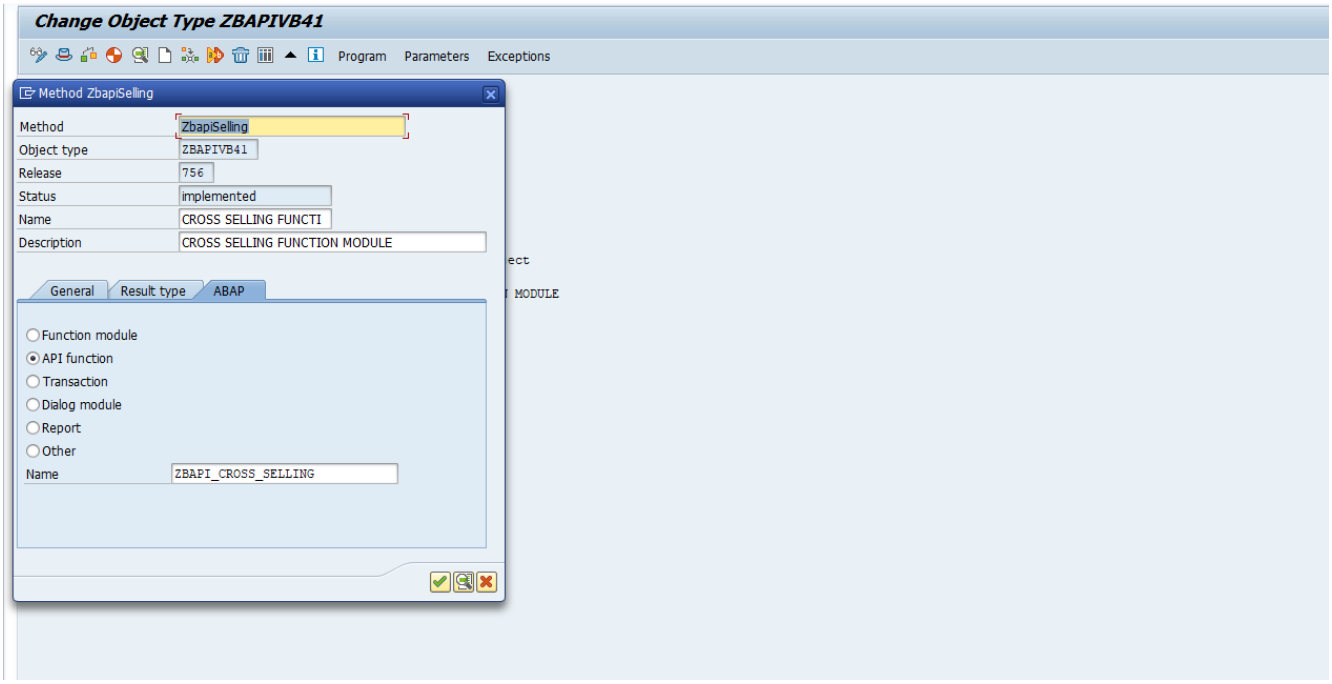

Change Object Type ZVB41
Program
Parameters
Exceptions
Method ZbapiSelling
Method
ZbapiSelling
Object type
ZVB41
Release
756
Status
implemented
Name
CROSS SELLING FUNCTI
Description
CROSS SELLING FUNCTION MODULE
General
Result type
ABAP
Dialog
Synchronous
Result parameter
Instance-independent

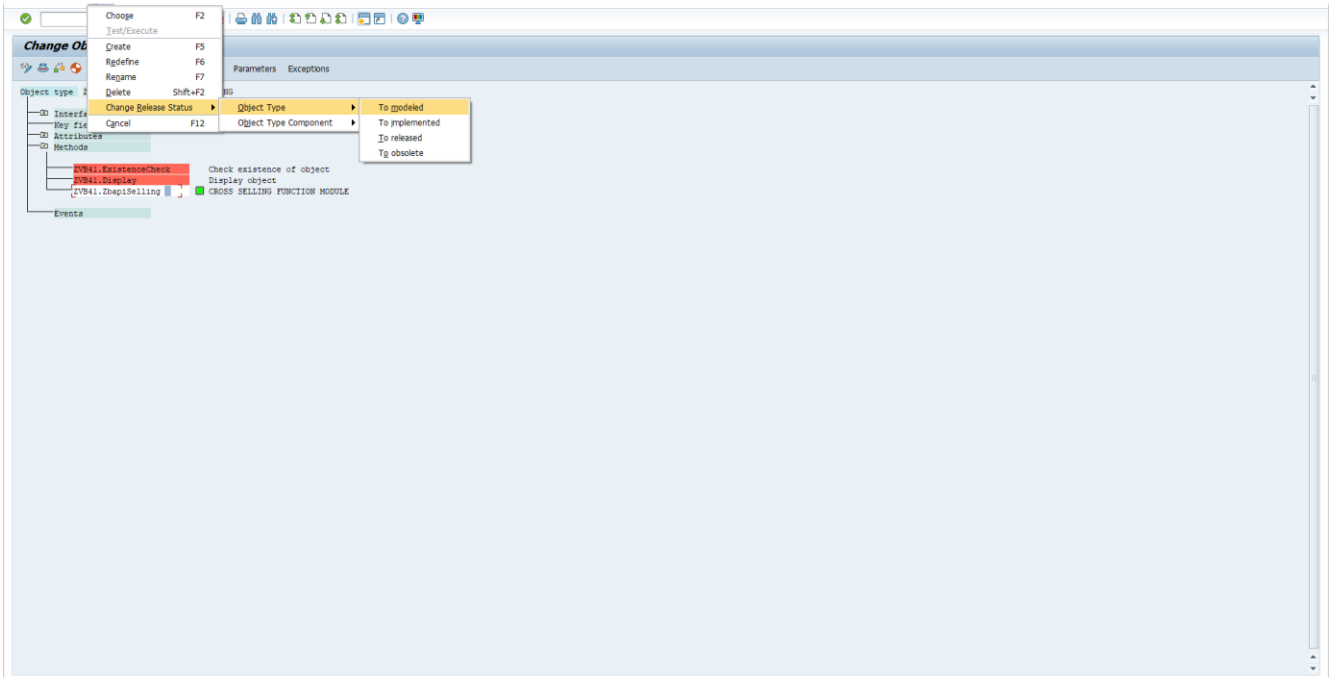


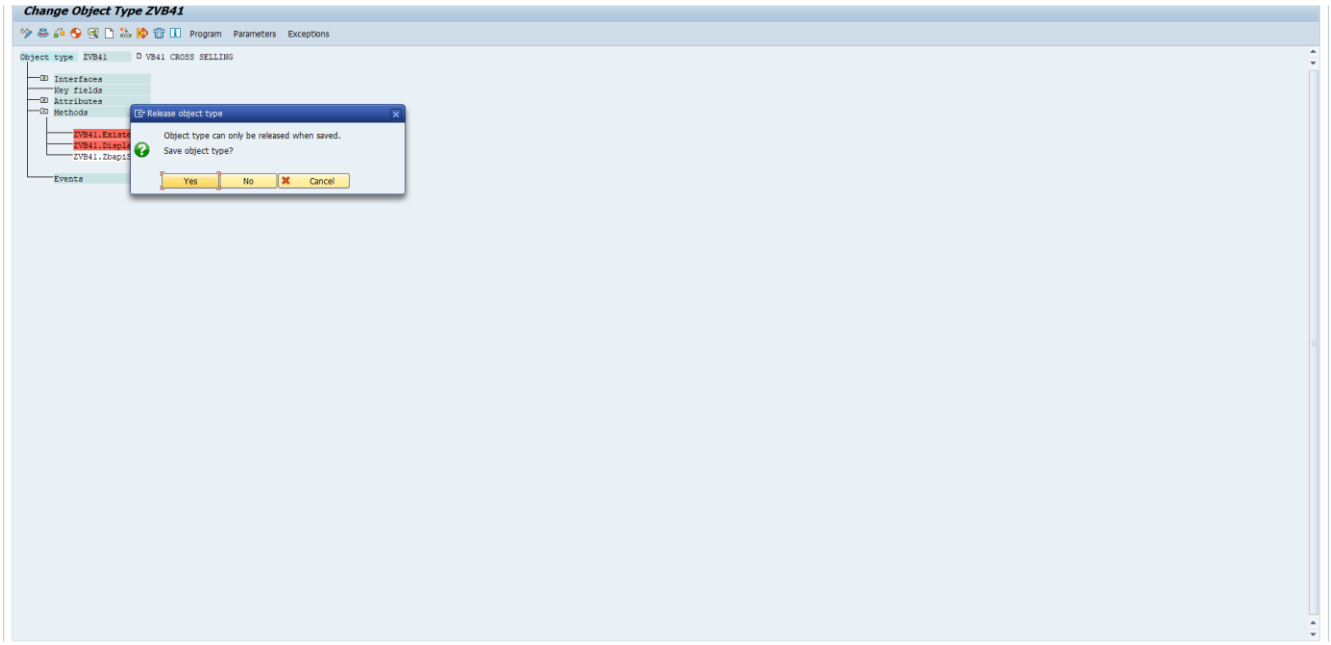
Step5. After the program has been generated and executed, check the program in the method just created. Thus , a BAPI is created.

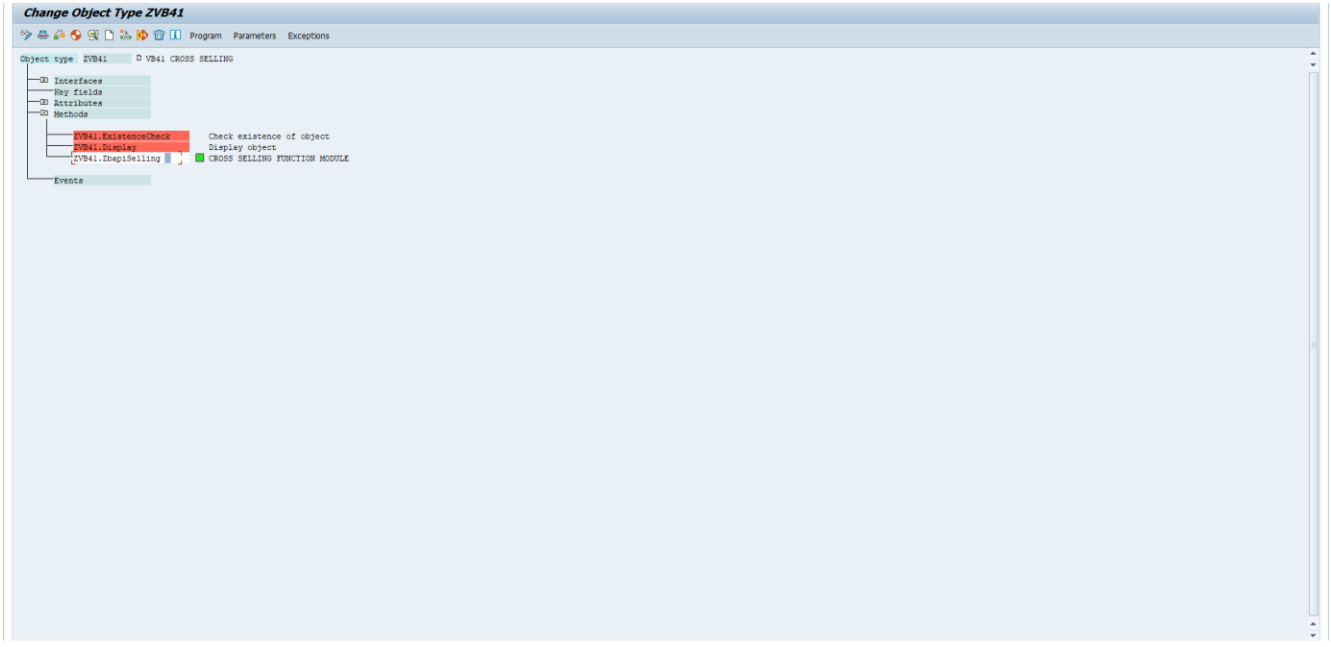

Change Object Type ZBAPIVB41
Program
Parameters
Exceptions
Object type
ZBAPIVB41
VB41
Interfaces
Key fields
Attributes
ZBAPIVB41.ObjectType
Object type
Methods
ZBAPIVB41.ExistenceCheck
Check existence of object
ZBAPIVB41.Display
Display object
ZBAPIVB41.ZbapiSelling
CROSS SELLING FUNCTION MODULE
Events

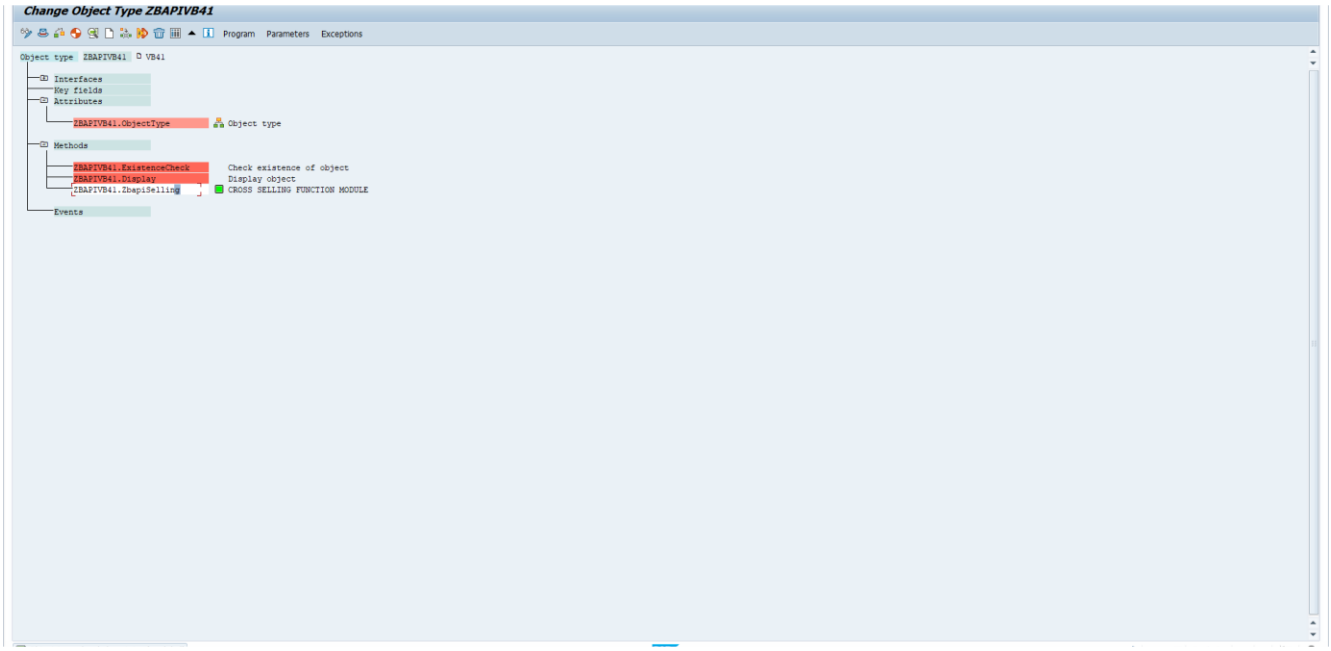

Change Object Type ZBAPIVB41
Program
Parameters
Exceptions
Method ZbapiSelling
Method
ZbapiSelling
Object type
ZBAPIVB41
Release
756
Status
implemented
Name
CROSS SELLING FUNCTI
Description
CROSS SELLING FUNCTION MODULE
General
Result type
ABAP
Dialog
Synchronous
Result parameter
Instance-independent
ect
MODULE

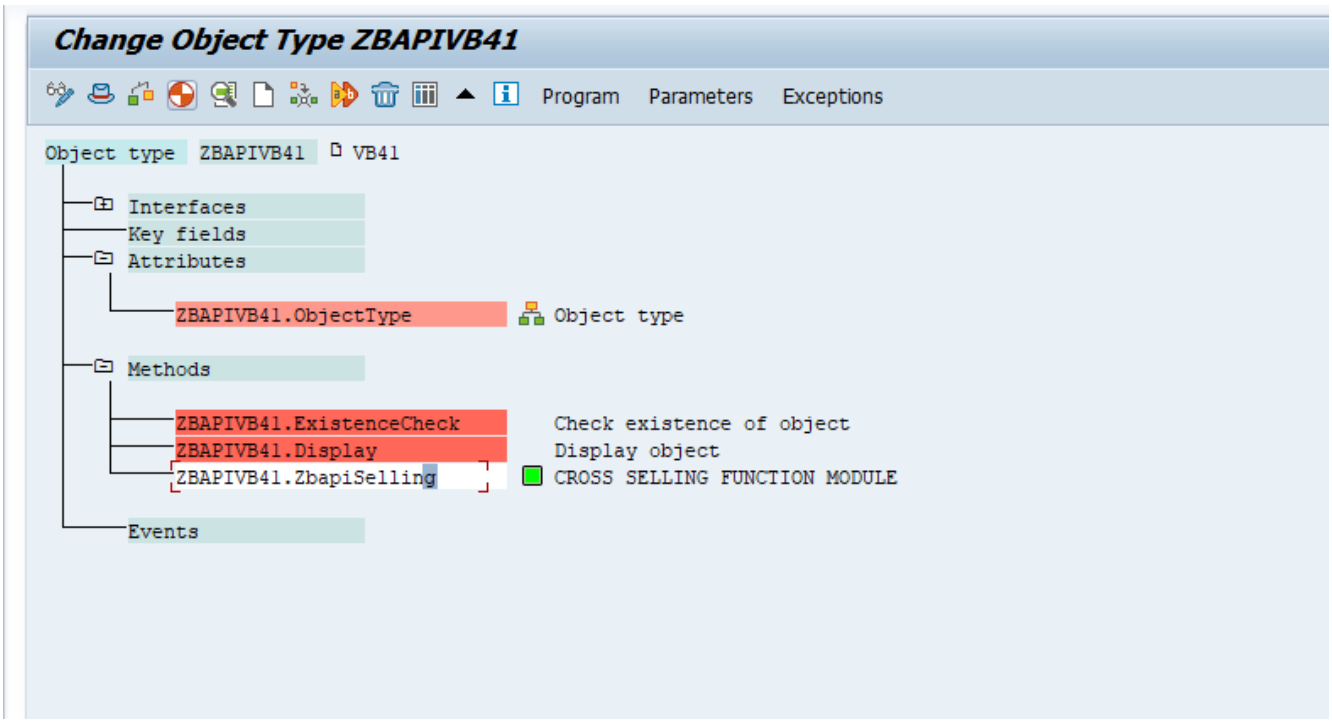

Change Object Type ZBAPIVB41
Program
Parameters
Exceptions
Method ZbapiSelling
Method
ZbapiSelling
Object type
ZBAPIVB41
Release
756
Status
implemented
Name
CROSS SELLING FUNCTI
Description
CROSS SELLING FUNCTION MODULE
General
Result type
ABAP
Function module
API function
Transaction
Dialog module
Report
Other
Name
ZBAPI_CROSS_SELLING

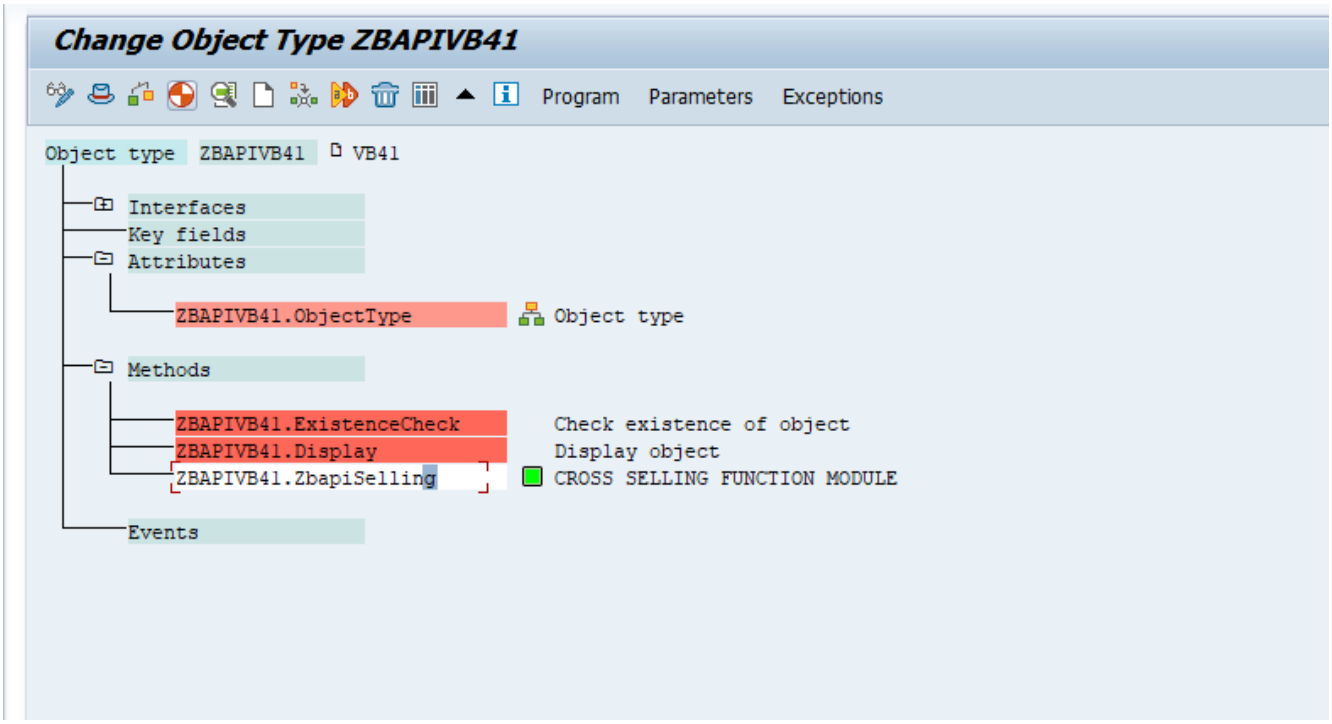

Choose F2
Test/Execute
Create F5
Redefine F6
Rename F7
Delete Shift+F2
Change Release Status
Cancel F12
Object Type
Object Type Component
To modeled
To implemented
To released
To obsolete
Parameters
Exceptions
Methods
ZVB41.ExistenceCheck
ZVB41.Display
ZVB41.ZbapiSelling
Check existence of object
Display object
CROSS SELLING FUNCTION MODULE
Events

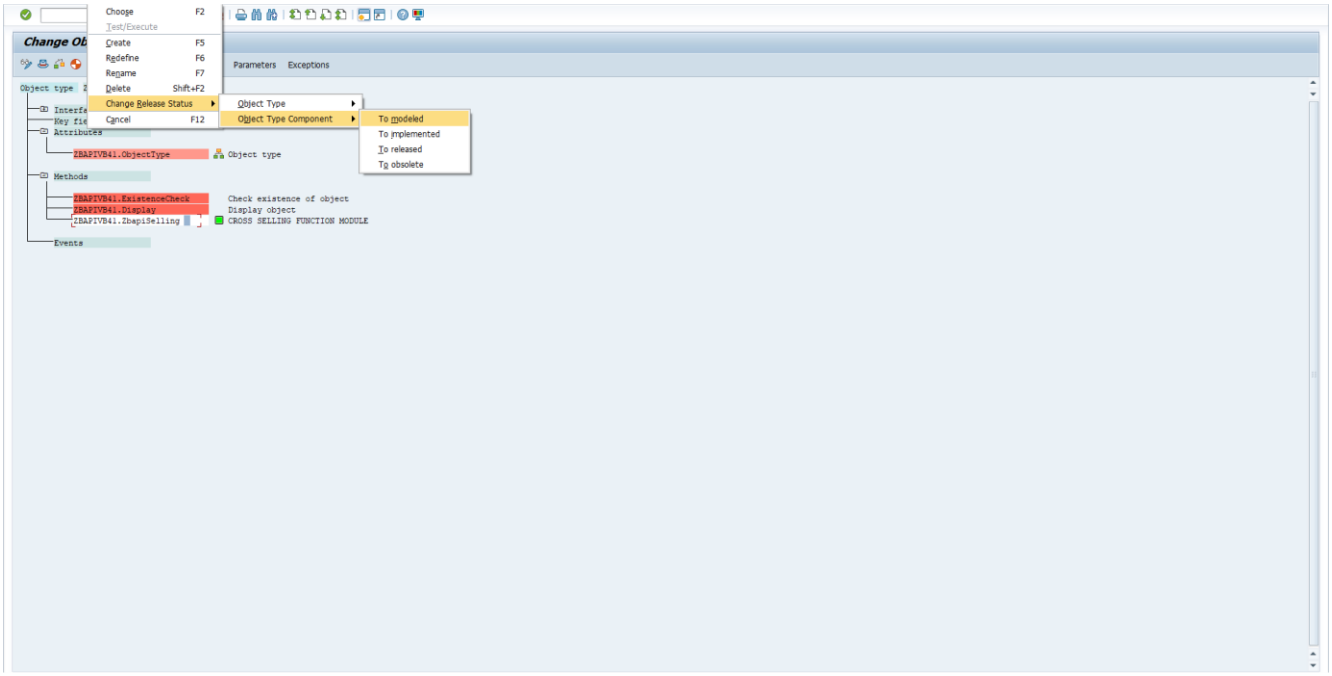

Change Object Type ZVB41
Program
Parameters
Exceptions
Object type ZVB41
VB41 CROSS SELLING
Interfaces
Key fields
Attributes
Methods
Events
Release object type
Object type can only be released when saved.
Save object type?
Yes
No
Cancel

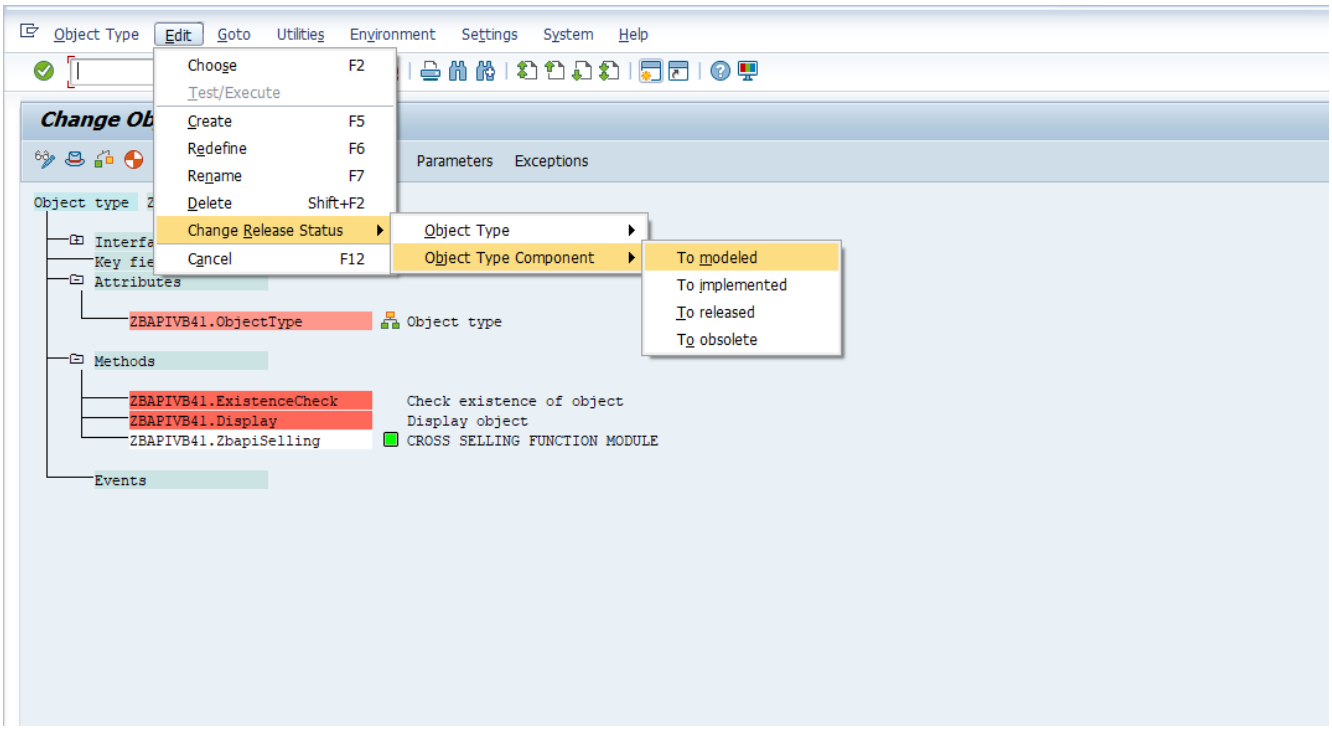

Change Object Type ZVB41
Program
Parameters
Exceptions
Object type ZVB41
VB41 CROSS SELLING
Interfaces
Key fields
Attributes
Methods
ZVB41.ExistenceCheck
Check existence of object
ZVB41.Display
Display object
ZVB41.ZbapiSelling
CROSS SELLING FUNCTION MODULE
Events

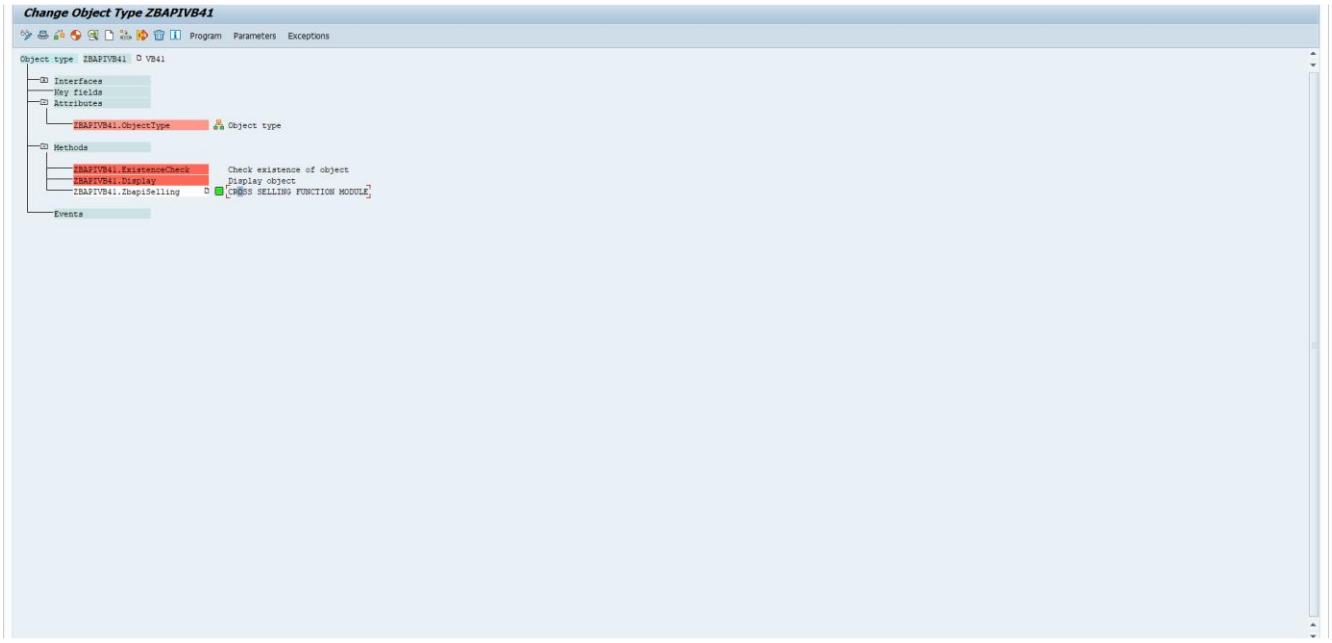

Change Object Type ZBAPIVB41
Program
Parameters
Exceptions
Object type
ZBAPIVB41
VB41
Interfaces
Key fields
Attributes
ZBAPIVB41.ObjectType
Object type
Methods
ZBAPIVB41.ExistenceCheck
Check existence of object
ZBAPIVB41.Display
Display object
ZBAPIVB41.ZbapiSelling
CROSS SELLING FUNCTION MODULE
Events

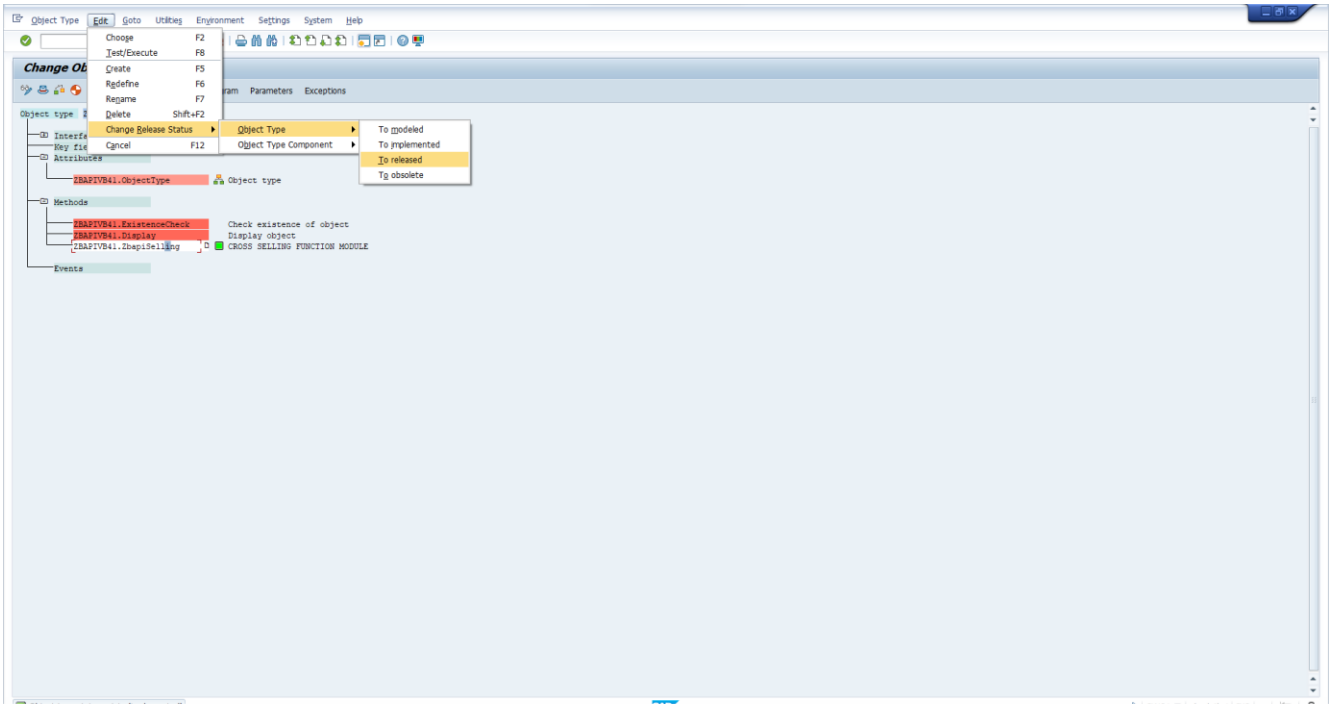

Change Object Type ZBAPIVB41
Program
Parameters
Exceptions
Object type
ZBAPIVB41
VB41
Interfaces
Key fields
Attributes
ZBAPIVB41.ObjectType
Object type
Methods
ZBAPIVB41.ExistenceCheck
Check existence of object
ZBAPIVB41.Display
Display object
ZBAPIVB41.ZbapiSelling
CROSS SELLING FUNCTION MODULE
Events

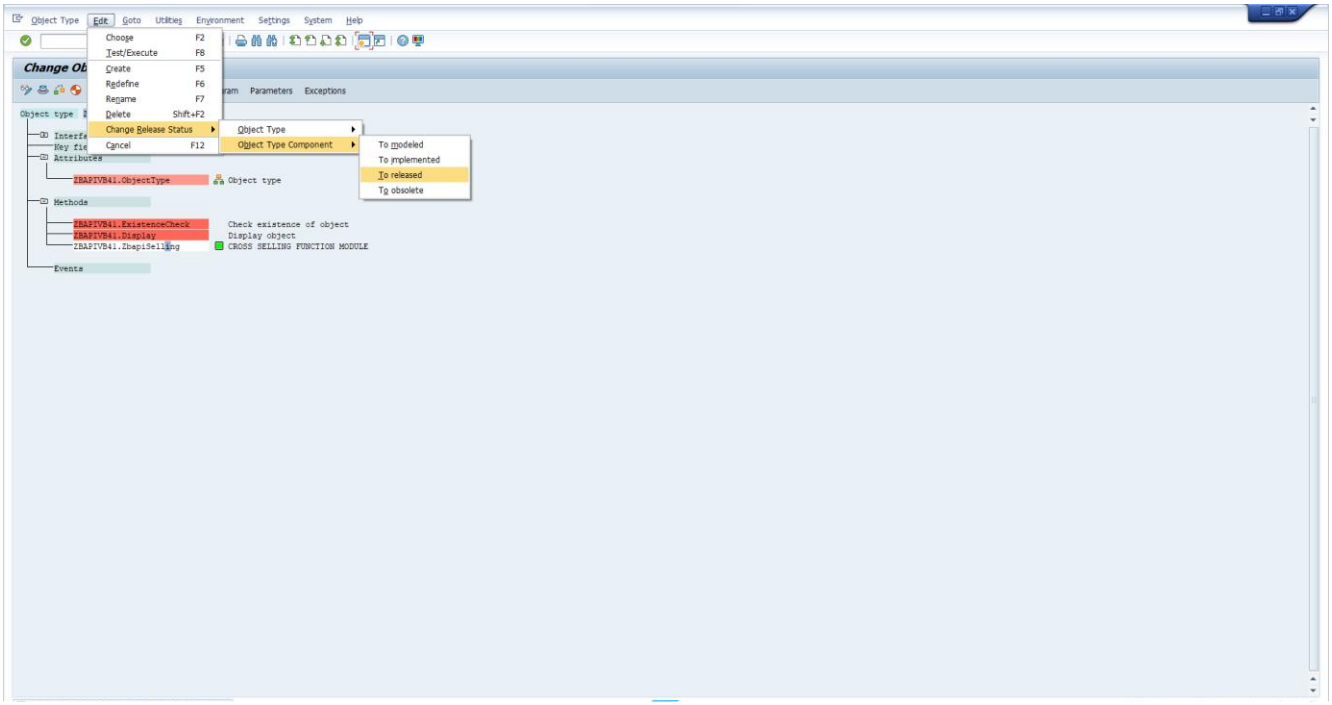

Change Object Type ZBAPIVB41
Program
Parameters
Exceptions
Object type
ZBAPIVB41
VB41
Interfaces
Key fields
Attributes
ZBAPIVB41.ObjectType
Object type
Methods
ZBAPIVB41.ExistenceCheck
Check existence of object
ZBAPIVB41.Display
Display object
ZBAPIVB41.ZbapiSelling
CROSS SELLING FUNCTION MODULE
Events

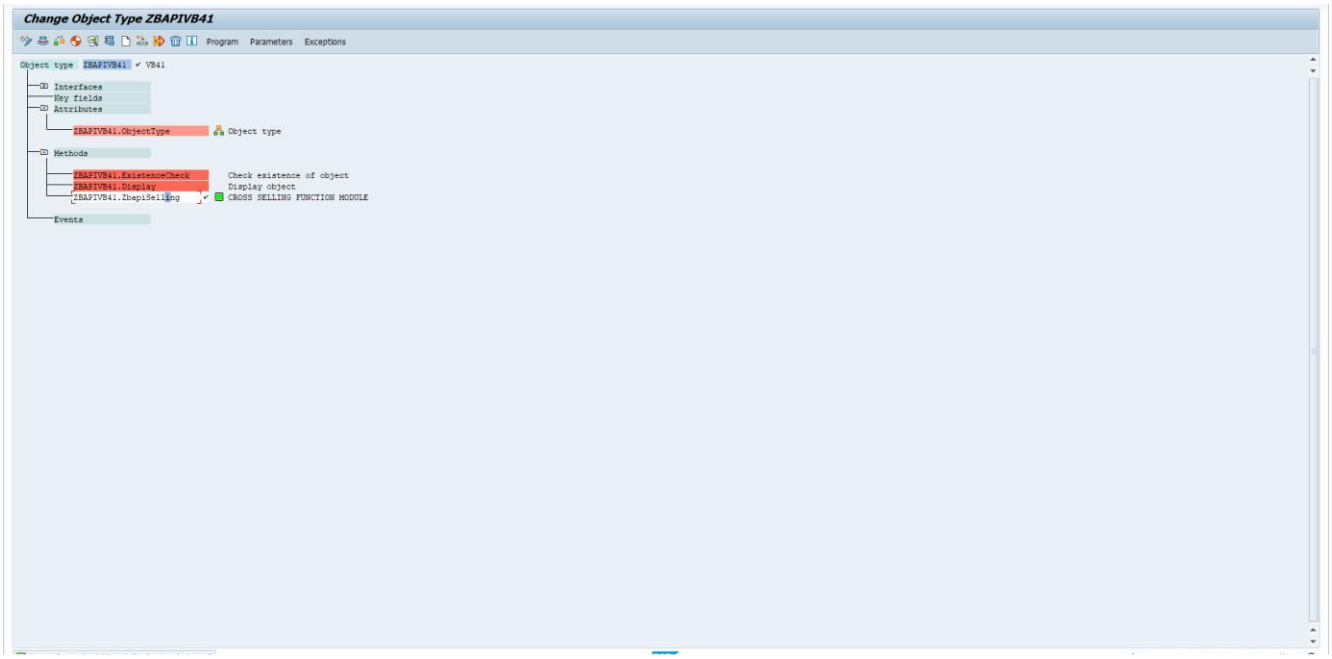

Choose F2
Test/Execute
Create F5
Redefine F6
Rename F7
Delete Shift+F2
Change Release Status
Cancel F12
Object Type
Object Type Component
To modeled
To implemented
To released
To obsolete
Parameters Exceptions
Object type
Attributes
ZBAPIVB41.ObjectType
Object type
Methods
ZBAPIVB41.ExistenceCheck
Check existence of object
ZBAPIVB41.Display
Display object
ZBAPIVB41.ZbapiSelling
CROSS SELLING FUNCTION MODULE
Events

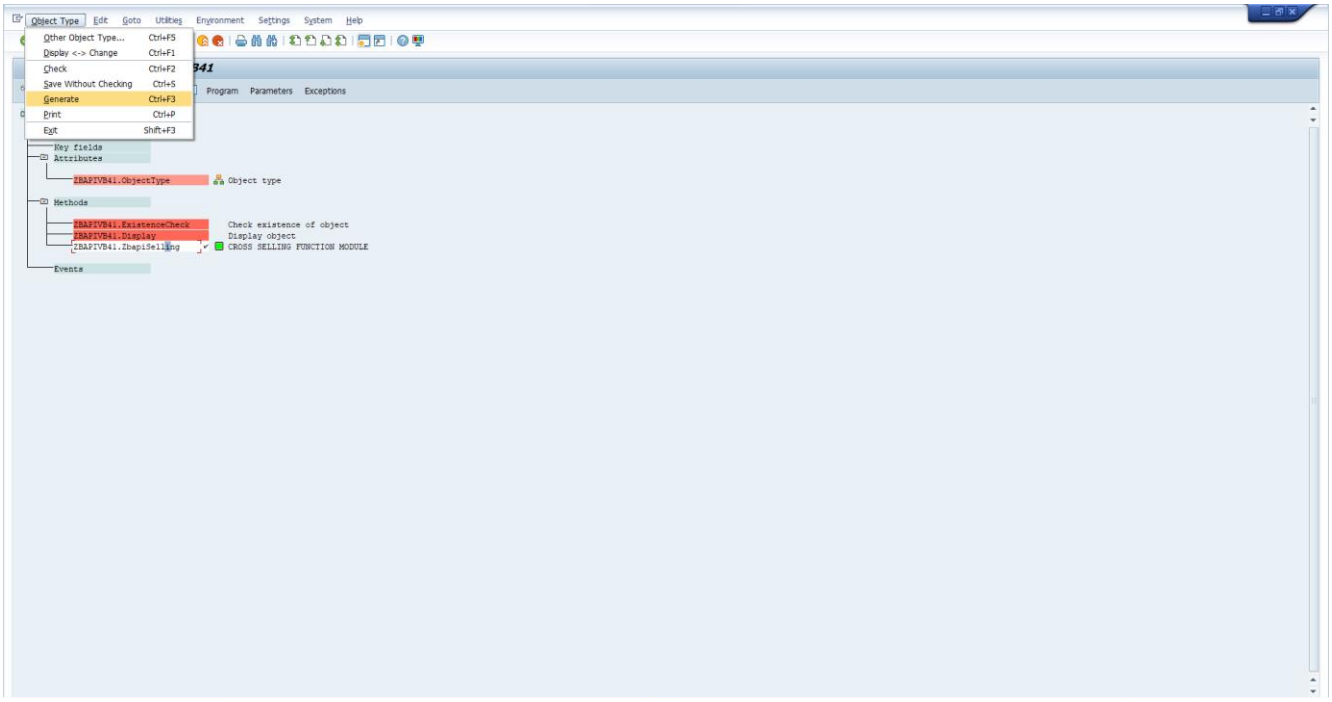

Object Type
Edit
Goto
Utilities
Environment
Settings
System
Help
Choose F2
Test/Execute
Create F5
Redefine F6
Rename F7
Delete Shift+F2
Change Release Status
Cancel F12
Object Type
Object Type Component
To modeled
To implemented
To released
To obsolete
Parameters Exceptions
Object type
Attributes
ZBAPIVB41.ObjectType
Object type
Methods
ZBAPIVB41.ExistenceCheck
Check existence of object
ZBAPIVB41.Display
Display object
ZBAPIVB41.ZbapiSelling
CROSS SELLING FUNCTION MODULE
Events

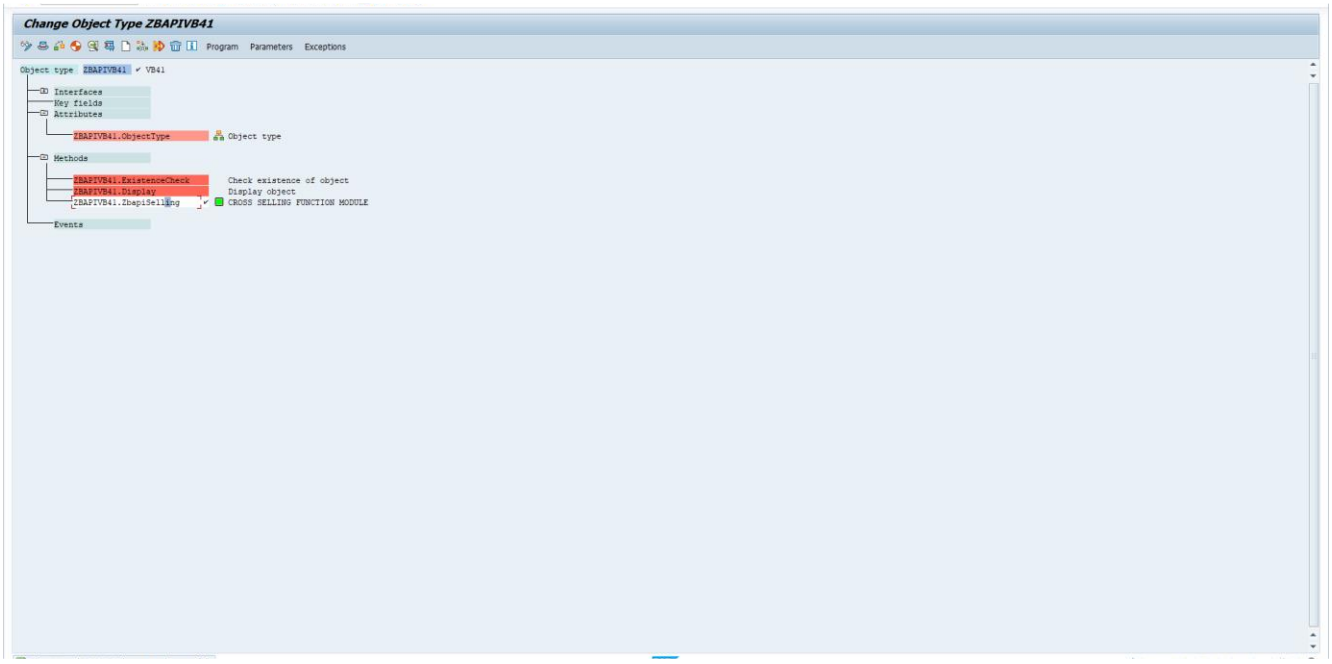

Change Object Type ZBAPIVB41
Program
Parameters
Exceptions
Object type ZBAPIVB41 VB41
Interfaces
Key fields
Attributes
ZBAPIVB41.ObjectType
Object type
Methods
ZBAPIVB41.ExistenceCheck
Check existence of object
ZBAPIVB41.Display
Display object
ZBAPIVB41.ZbapiSelling
CROSS SELLING FUNCTION MODULE
Events


Object Type
Edit
Goto
Utilities
Environment
Settings
System
Help
Choose F2
Test/Execute F8
Create F5
Redefine F6
Rename F7
Delete Shift+F2
Change Release Status
Cancel F12
Object Type
Object Type Component
To modeled
To implemented
To released
To obsolete
Parameters
Exceptions
Attributes
ZBAPIVB41.ObjectType
Object type
Methods
ZBAPIVB41.ExistenceCheck
Check existence of object
ZBAPIVB41.Display
Display object
ZBAPIVB41.ZbapiSelling
CROSS SELLING FUNCTION MODULE
Events


Object Type
Edit
Goto
Utilities
Environment
Settings
System
Help
Choose F2
Test/Execute F8
Create F5
Redefine F6
Rename F7
Delete Shift+F2
Change Release Status
Cancel F12
Object Type
Object Type Component
To modeled
To implemented
To released
To obsolete
Parameters
Exceptions
Object type
Interfa
Key fie
Attributes
ZBAPIVB41.ObjectType
Object type
Methods
ZBAPIVB41.ExistenceCheck
Check existence of object
ZBAPIVB41.Display
Display object
ZBAPIVB41.ZbapiSelling
CROSS SELLING FUNCTION MODULE
Events

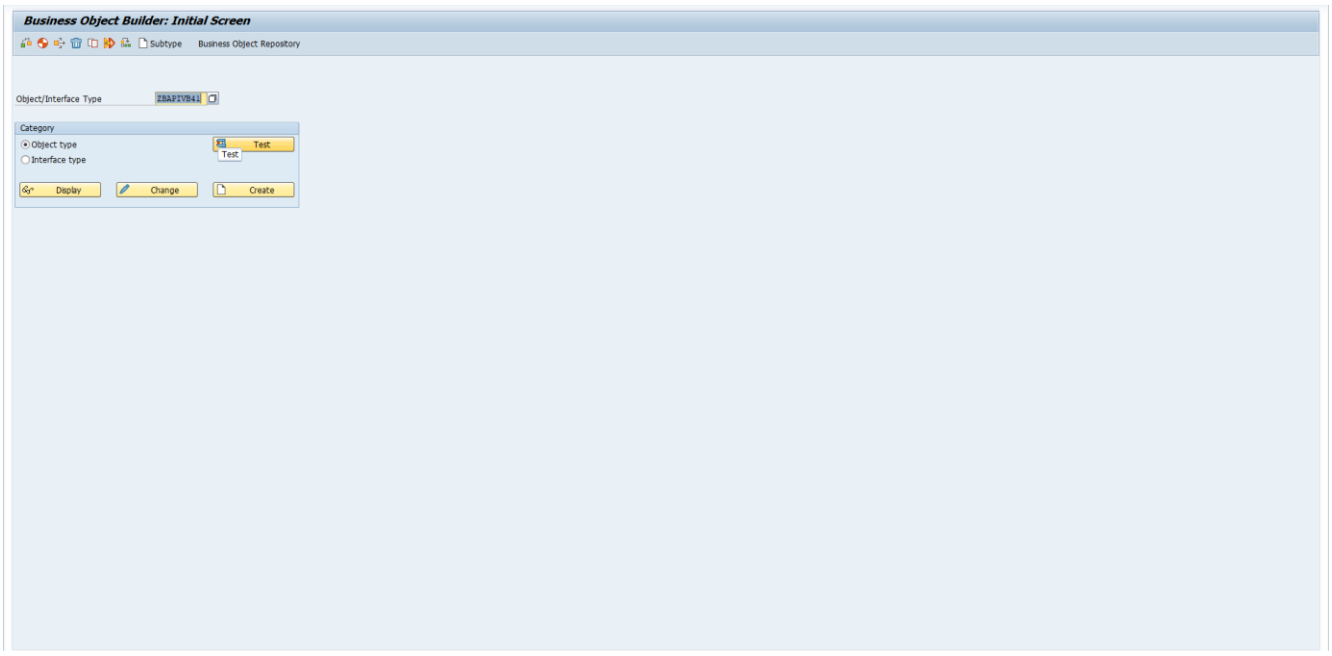

Change Object Type ZBAPIVB41
Program
Parameters
Exceptions
Object type ZBAPIVB41 VB41
Interfaces
Key fields
Attributes
ZBAPIVB41.ObjectType
Object type
Methods
ZBAPIVB41.ExistenceCheck
Check existence of object
ZBAPIVB41.Display
Display object
ZBAPIVB41.ZbapiSelling
CROSS SELLING FUNCTION MODULE
Events

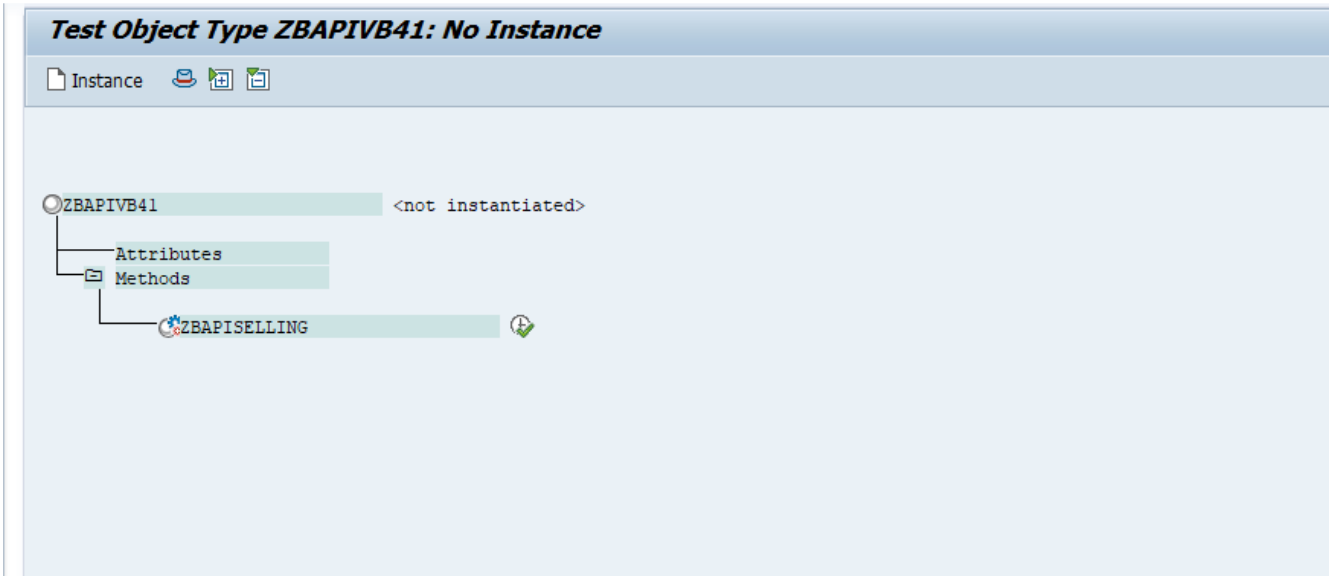

Object Type
Edit
Goto
Utilities
Environment
Settings
System
Help
Other Object Type... Ctrl+F5
Display <-> Change Ctrl+F1
Check Ctrl+F2
Save Without Checking Ctrl+S
Generate Ctrl+F3
Print Ctrl+P
Exit Shift+F3
Program
Parameters
Exceptions
Key fields
Attributes
ZBAPIVB41.ObjectType
Object type
Methods
ZBAPIVB41.ExistenceCheck
Check existence of object
ZBAPIVB41.Display
Display object
ZBAPIVB41.ZbapiSelling
CROSS SELLING FUNCTION MODULE
Events

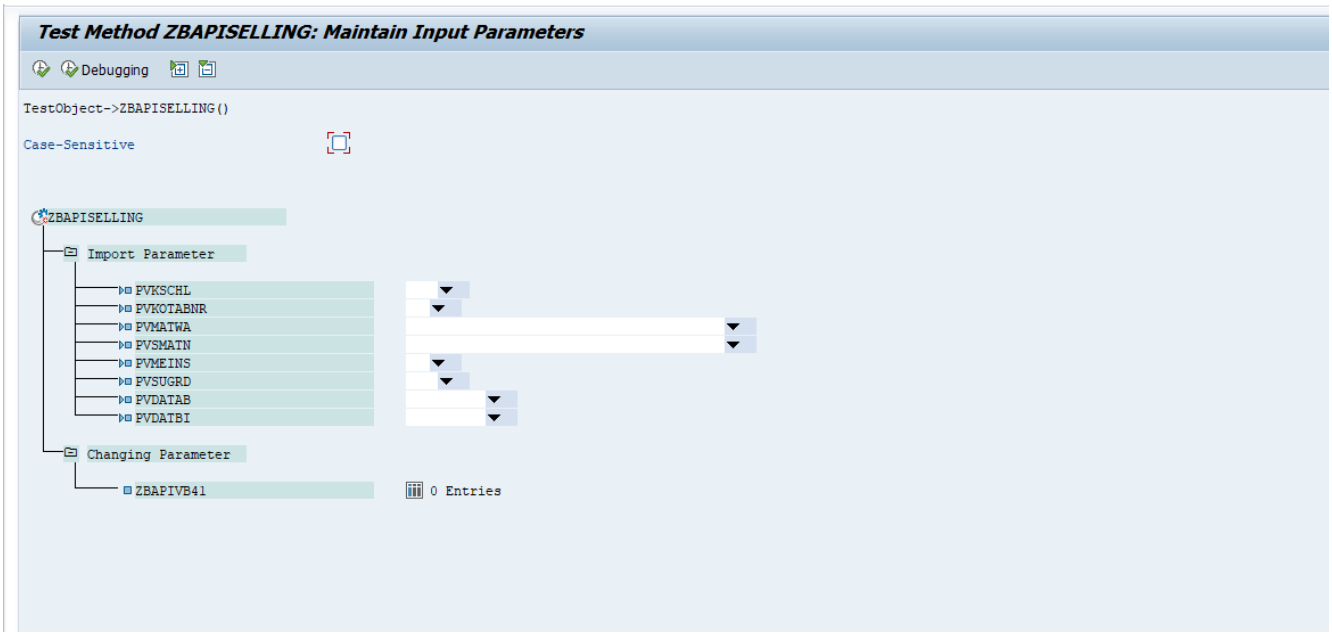

Change Object Type ZBAPIVB41
Program
Parameters
Exceptions
Object type ZBAPIVB41
VB41
Interfaces
Key fields
Attributes
ZBAPIVB41.ObjectType
Object type
Methods
ZBAPIVB41.ExistenceCheck
Check existence of object
ZBAPIVB41.Display
Display object
ZBAPIVB41.ZbapiSelling
CROSS SELLING FUNCTION MODULE
Events

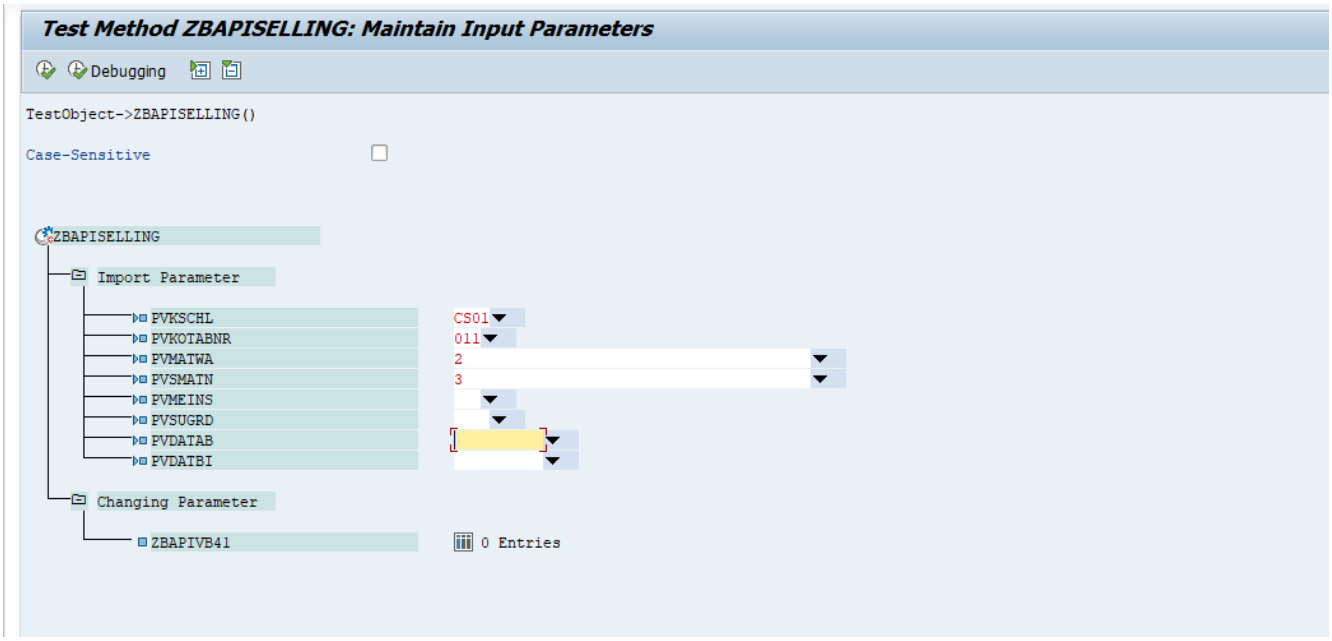
Testing the BAPI

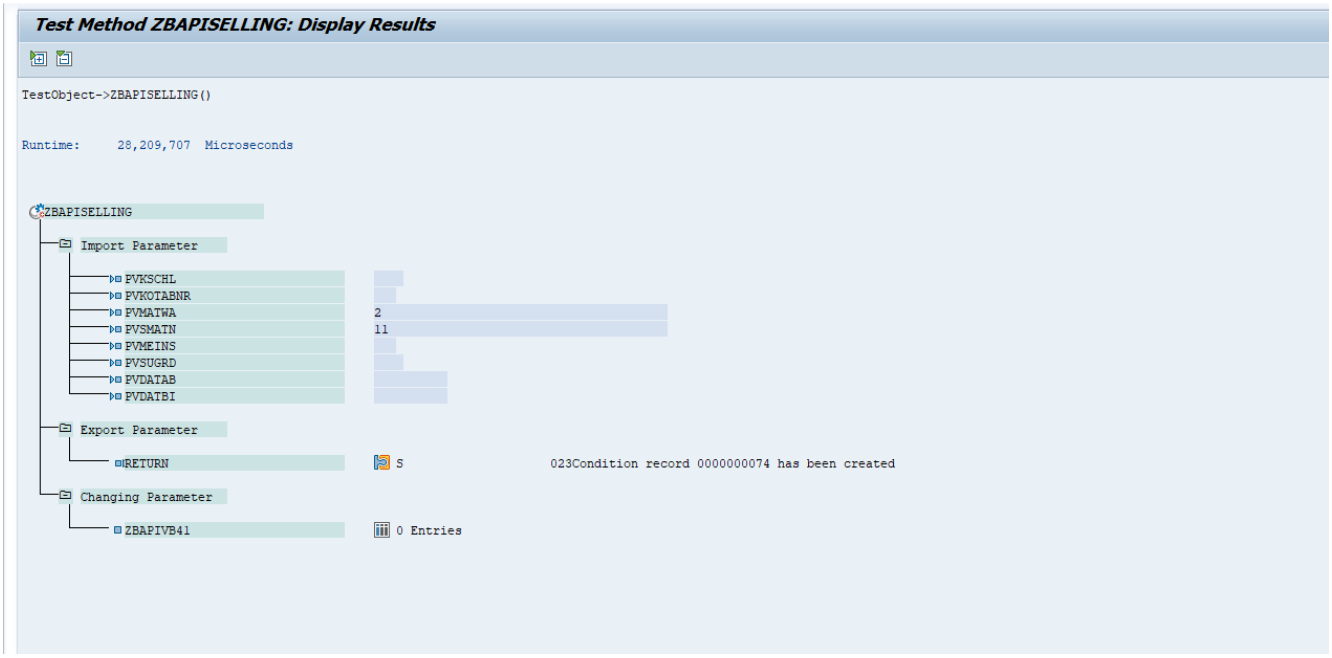
You can test the BAPI by Testing the individual method of the Business Object in the Business Object Builder. .




Test Object Type ZBAPIVB41: No Instance
Instance
ZBAPIVB41
<not instantiated>
Attributes
Methods
ZBAPISELLING

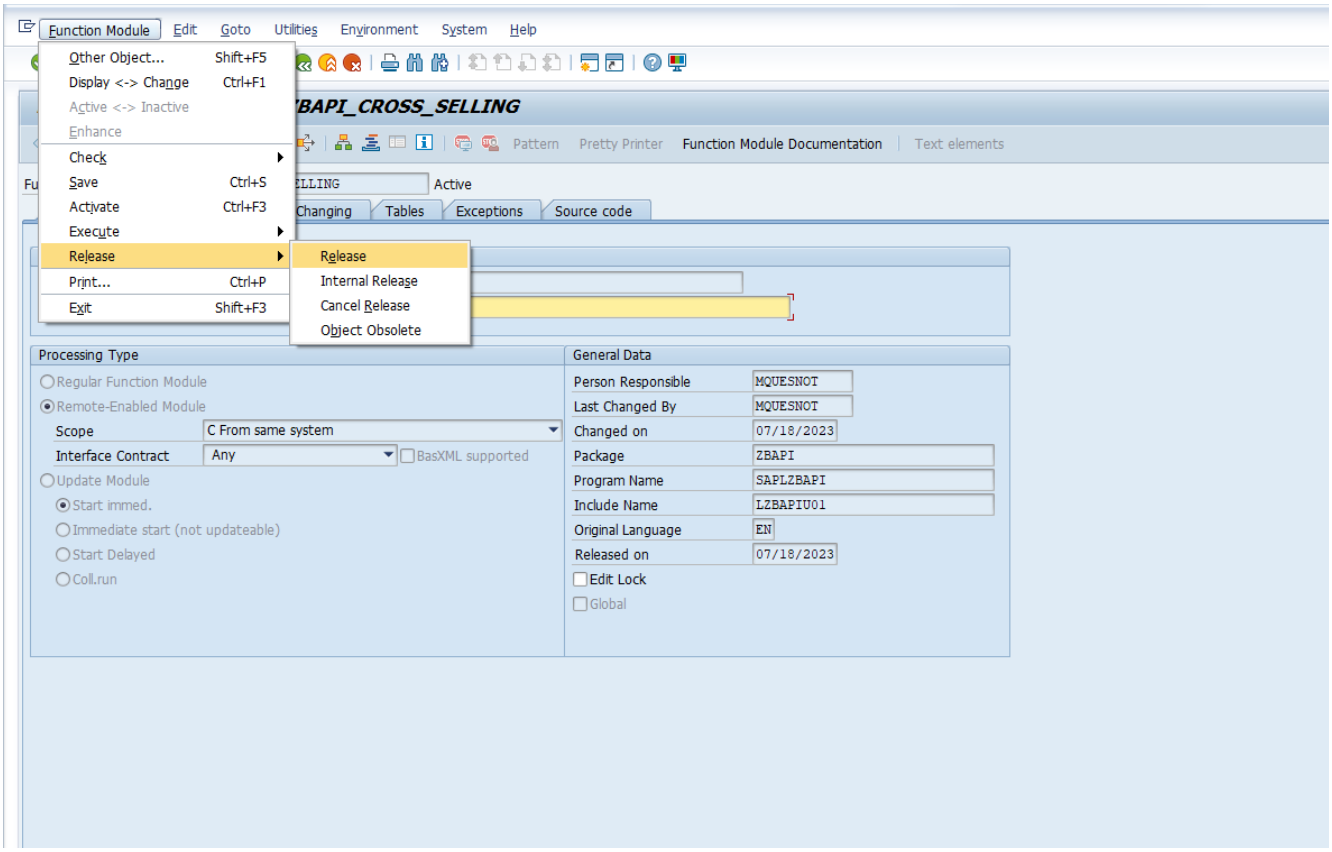

Test Method ZBAPISELLING: Maintain Input Parameters
Debugging
TestObject->ZBAPISELLING()
Case-Sensitive
ZBAPISELLING
Import Parameter
PVKSCHL
PVKOTABNR
PVMATWA
PVSMATN
PVMEINS
PVSUGRD
PVDATAB
PVDATBI
Changing Parameter
ZBAPIVB41
0 Entries


Test Method ZBAPISELLING: Maintain Input Parameters
Debugging
TestObject->ZBAPISELLING()
Case-Sensitive
ZBAPISELLING
Import Parameter
PVKSCHL
CS01
PVKOTABNR
011
PVMATWA
2
PVSMATN
3
PVMEINS
PVSUGRD
PVDATAB
PVDATBI
Changing Parameter
ZBAPIVB41
0 Entries

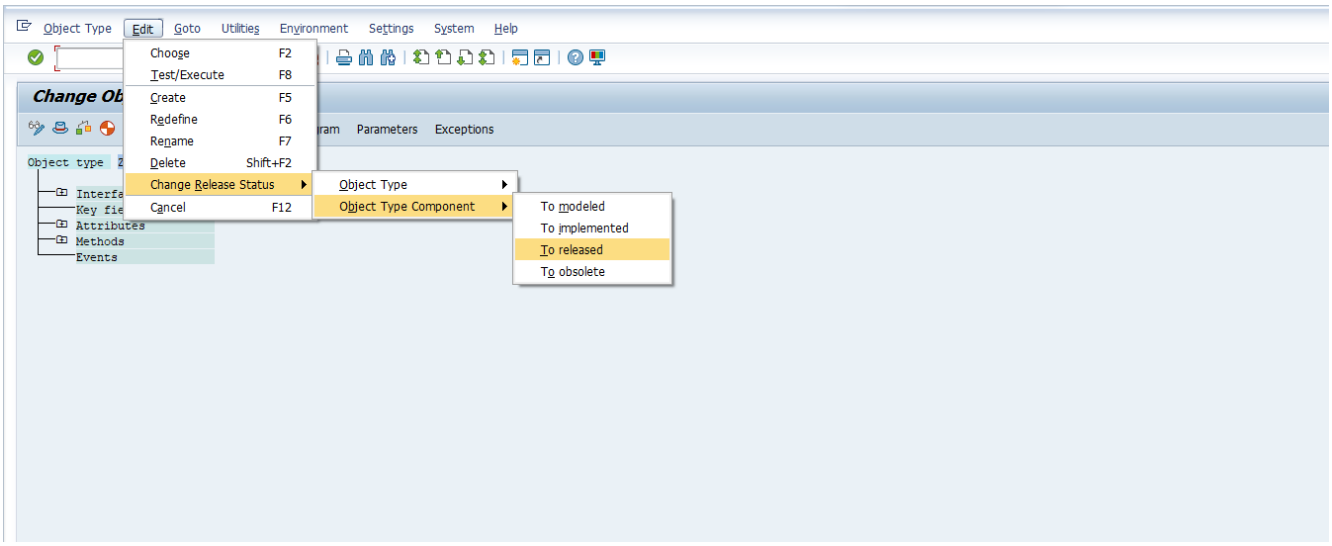

Test Method ZBAPISELLING: Display Results
TestObject->ZBAPISELLING()
Runtime: 28,209,707 Microseconds
ZBAPISELLING
Import Parameter
PVKSCHL
PVKOTABNR
PVMATWA
2
PVSMATN
11
PVMEINS
PVSUGRD
PVDATAB
PVDATBI
Export Parameter
RETURN
S
023Condition record 0000000074 has been created
Changing Parameter
ZBAPIVB41
0 Entries

Releasing and freezing the BAPI

- To release the BAPI , first release the function module (using transaction se37) .



- Set the status of the method to ‘released’ in the Business Object Builder (using transaction SWo1 – Edit-> change status-> released.)



You can also use the BAPI Explorer (Transaction code BAPI) for 360' view on BAPI