Documentation of Winshuttle Wrapper RFM’s

1. /WINSHTLQ/BPAR_C_A_VAT_NUMBER

- ✓ **PURPOSE:** Check Additional Data of Business Partner
- ✓ **DESCRIPTION:** This WFM is a wrapper of ‘BPAR_C_A_VAT_NUMBER’ FM which is used to check the VAT number specific to the country entered.

Following is the interface for the function module

✓ **Import Parameters**

1. **COUNTRY:** Enter country of Business partner
2. **VAT_NUMBER:** Enter VAT Registration number

✓ **Export Parameters**

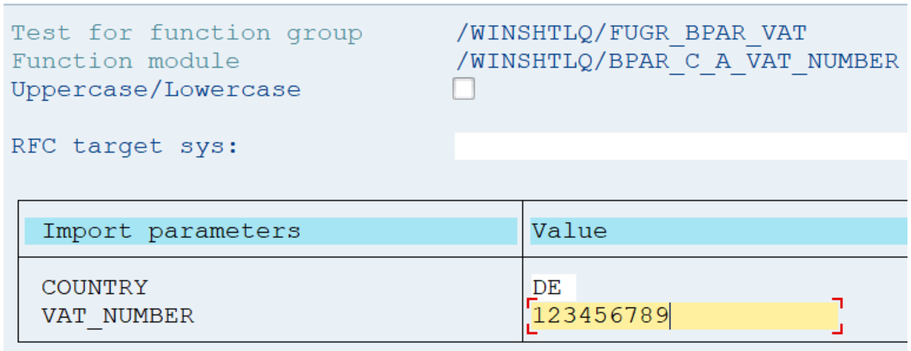
1. **MESSAGE:** Displays error message if any.

✓ **Exceptions**

1. **ERROR:** This exception occurs in case of any error.
2. **NO_VAT_NUMBER_IN_EG:** This exception occurs if entered VAT number is invalid.
3. **NOT_AUTHORIZED_MSG_GETDETAIL:** This exception occurs if the user is unauthorized to run BAPI_MESSAGE_GETDETAIL FM.
4. **NOT_AUTH_BPAR_C_A_VAT_NUMBER:** This exception occurs if the user is unauthorized to run /WINSHTLQ/BPAR_C_A_VAT_NUMBER FM.
5. **NO_RELEASE_FOUND:** This exception occurs if no SAP BASIS release is found in the SAP system.

Example:

Enter Input as below:

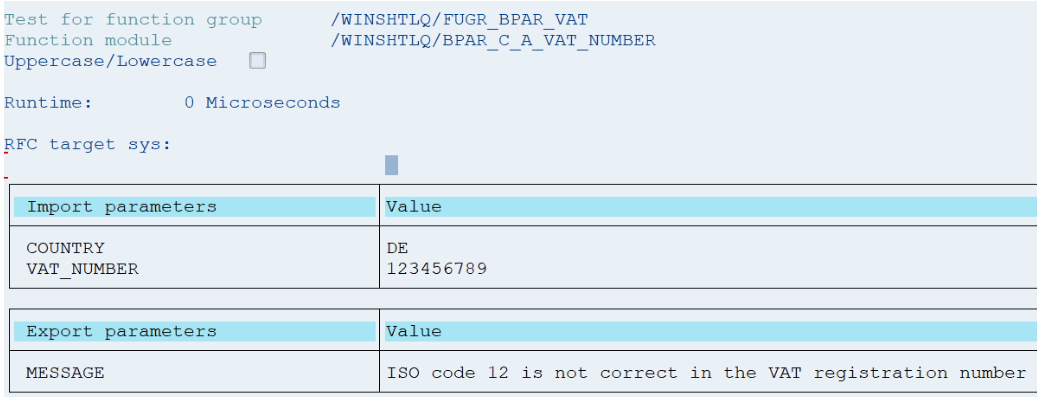


Test for function group /WINSHTLQ/FUGR_BPAR_VAT
Function module /WINSHTLQ/BPAR_C_A_VAT_NUMBER
Uppercase/Lowercase □

RFC target sys:

Import parameters	Value
COUNTRY	DE
VAT_NUMBER	123456789

Since the entered VAT number is invalid, it gives error message as below:

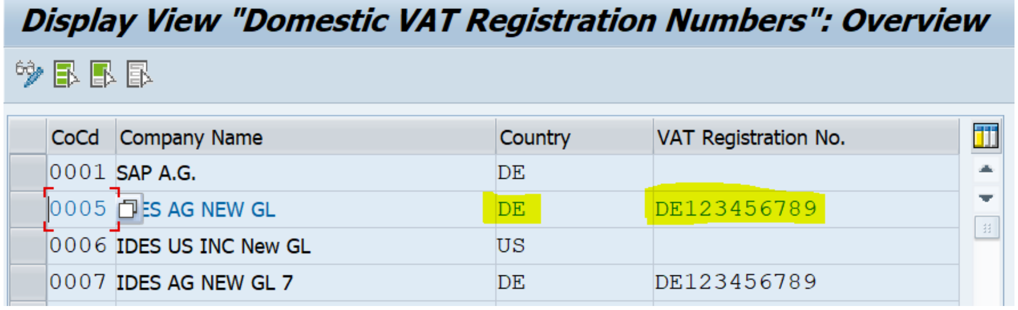


```
Test for function group       /WINSHTLQ/FUGR_BPAR_VAT
Function module               /WINSHTLQ/BPAR_C_A_VAT_NUMBER
Uppercase/Lowercase      □

Runtime:          0 Microseconds

RFC target sys:
```

Import parameters	Value
COUNTRY	DE
VAT_NUMBER	123456789

Export parameters	Value
MESSAGE	ISO code 12 is not correct in the VAT registration number

However, when correct VAT number is entered for the country as below, no error message will be displayed.



Display View "Domestic VAT Registration Numbers": Overview

CoCd	Company Name	Country	VAT Registration No.
0001	SAP A.G.	DE	
0005	ES AG NEW GL	DE	DE123456789
0006	IDES US INC New GL	US	
0007	IDES AG NEW GL 7	DE	DE123456789

2. /WINSHTLQ/BANK_ACCOUNT_CHECK

- ✓ **PURPOSE:** Check of Bank Account Number
- ✓ **DESCRIPTION:** This WFM is a wrapper of 'BANK_ACCOUNT_CHECK' FM which is used to check the entered Bank Account Number along with Bank Control Key, Bank country and Bank number.

Following is the interface for the function module

✓ **Import Parameters**

1. **BANK_ACCOUNT:** Enter Bank account number
2. **BANK_CONTROL_KEY:** Enter Bank Control Key
3. **BANK_COUNTRY:** Enter Bank Country Key
4. **BANK_NUMBER:** Enter Bank Key

✓ **Export Parameters**

1. **MESSAGE:** Displays error message if any.

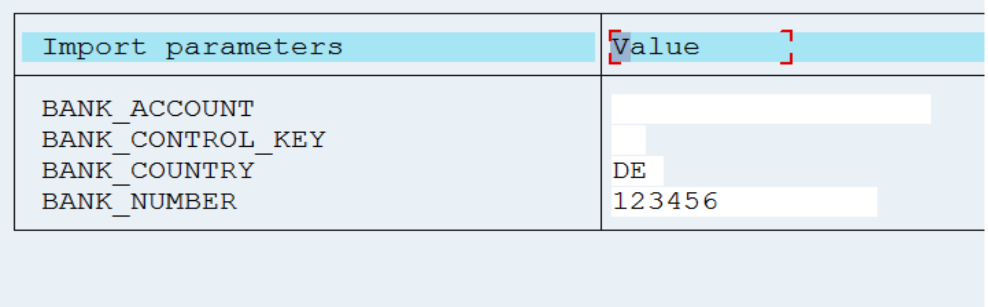
✓ **Exceptions**

1. **NOT_VALID:** This exception occurs if entered Bank account number is invalid.
2. **NOT_AUTHORIZED_MSG_GETDETAIL:** This exception occurs if the user is unauthorized to run BAPI_MESSAGE_GETDETAIL FM.

3. **NOT_AUTH_BANK_ACCOUNT_CHECK:** This exception occurs if the user is unauthorized to run /WINSHTLQ/BANK_ACCOUNT_CHECK FM.
4. **NO_RELEASE_FOUND:** This exception occurs if no SAP BASIS release is found in the SAP system.

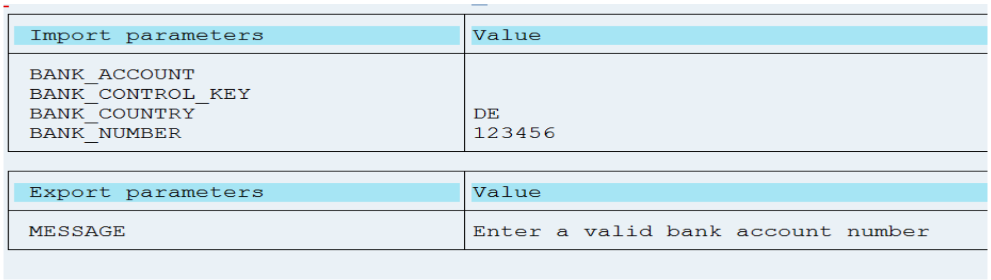
Example:

Enter Input as below:



Import parameters	Value
BANK_ACCOUNT	
BANK_CONTROL_KEY	
BANK_COUNTRY	DE
BANK_NUMBER	123456

If the entered input details are valid, no error message is displayed. However, if the entered Bank details are invalid, below error message will be displayed.



Import parameters	Value
BANK_ACCOUNT	
BANK_CONTROL_KEY	
BANK_COUNTRY	DE
BANK_NUMBER	123456

Export parameters	Value
MESSAGE	Enter a valid bank account number

3. /WINSHTLQ/BUSINESS_NUMBER_CHEK

- ✓ **PURPOSE:** Check of international location number (ILN)
- ✓ **DESCRIPTION:** This function module is a wrapper of ‘BUSINESS_NUMBER_CHECK’ and checks the international location number (ILN) for each country (controlled by the international license plate number).

The BUSINESS_NUMBER_CHECK function module checks the international location number (ILN) for each country (controlled by the international license plate number). The ILN consists of 3 parts: positions 1 to 7 (this part is described as the base number in the case of an ILN of type 2), positions 8 to 12, and the check digit.

If the check has negative results i.e. the number is invalid, then the NOT_VALID exception is triggered. A corresponding message is issued at the same time. If the number is valid then nothing happens.

In the case of South Africa, the system carries out the check according to a special algorithm which differs from the one for all other countries. A check does not take place for Belgium, Italy, Spain and the Netherlands.

Following is the interface for the function module

✓ **Import Parameters**

1. **BUSINESS_NUMBER:** International location number (part 1)
2. **CHECK_NUMBER:** Check digit for international location number
3. **COUNTRY:** Country Key
4. **SUPPLEMENT_NUMBER:** International location number (Part 2)

✓ **Export Parameters**

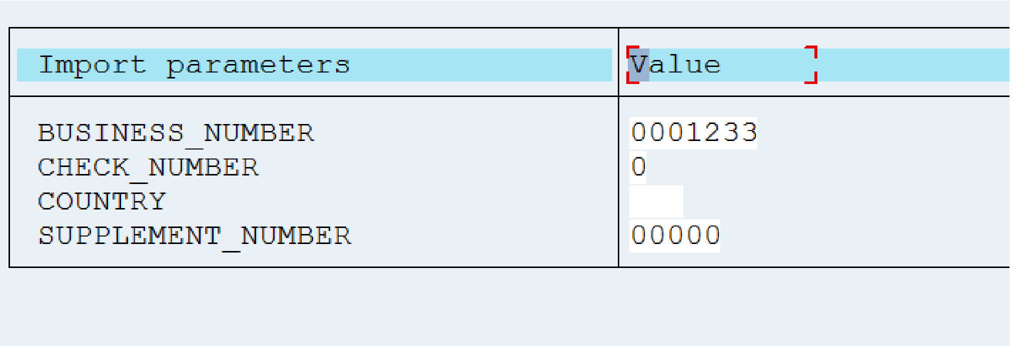
1. **MESSAGE:** Displays error message if any.

✓ **Exceptions**

1. **NOT_VALID:** This exception occurs if entered Business number is invalid.
2. **NOT_AUTHORIZED_MSG_GETDETAIL:** This exception occurs if the user is unauthorized to run BAPI_MESSAGE_GETDETAIL FM.
3. **NOT_AUTH_BUIS_NUMBER_CHEK:** This exception occurs if the user is unauthorized to run /WINSHTLQ/BUSINESS_NUMBER_CHEK FM.
4. **NO_RELEASE_FOUND:** This exception occurs if no SAP BASIS release is found in the SAP system.

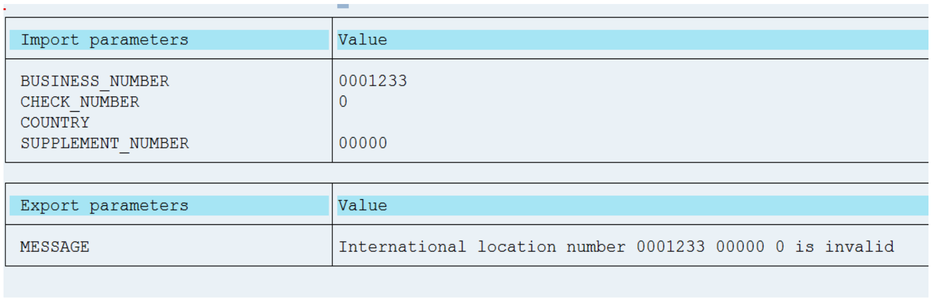
Example:

Enter Input as below:



Import parameters	Value
BUSINESS_NUMBER	0001233
CHECK_NUMBER	0
COUNTRY	
SUPPLEMENT_NUMBER	00000

If the entered details are invalid, below error message is displayed.



Import parameters	Value
BUSINESS_NUMBER	0001233
CHECK_NUMBER	0
COUNTRY	
SUPPLEMENT_NUMBER	00000

Export parameters	Value
MESSAGE	International location number 0001233 00000 0 is invalid

4. /WINSHTLQ/EU_TAX_NUMBER_CHECK

- ✓ **PURPOSE:** Check of VAT registration number (EU tax number)
- ✓ **DESCRIPTION:** This function module is a wrapper of 'EU_TAX_NUMBER_CHECK' and checks the VAT registration number (EU tax number) for each country.

The check rules are stored in table T005 in the field PRUIN. The length of the VAT number, which is important for this check, is stored in field UINLN.

If the check comes across an invalid format, an incorrect country key, or an incorrect check digit, the exception NOT_VALID is triggered, and an error message is sent. If the check runs successfully, nothing will be drawn to the attention of the user.

Following is the interface for the function module

- ✓ **Import Parameters**

1. **COUNTRY:** Country Key
2. **EU_TAX_NUMBER:** VAT registration number (EU tax number)

- ✓ **Export Parameters**

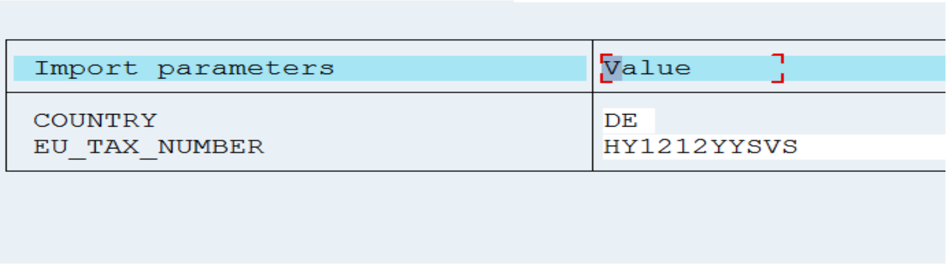
1. **MESSAGE:** Displays error message if any.

- ✓ **Exceptions**

1. **NOT_VALID:** This exception occurs if entered VAT number is invalid.
2. **NOT_AUTHORIZED_MSG_GETDETAIL:** This exception occurs if the user is unauthorized to run BAPI_MESSAGE_GETDETAIL FM.
3. **NOT_AUTHORIZED_EU_TAX_NUM_CHK: This** exception occurs if the user is unauthorized to run /WINSHTLQ/EU_TAX_NUMBER_CHECK FM.
4. **NO_RELEASE_FOUND:** This exception occurs if no SAP BASIS release is found in the SAP system.

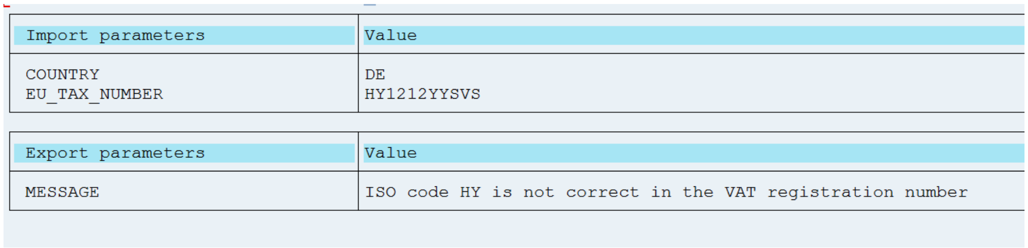
Example:

Enter Input as below:



Import parameters	Value
COUNTRY EU_TAX_NUMBER	DE HY1212YYSVS

No such Tax number exists, hence below error message will be displayed.



Import parameters	Value
COUNTRY EU_TAX_NUMBER	DE HY1212YYSVS

Export parameters	Value
MESSAGE	ISO code HY is not correct in the VAT registration number

5. /WINSHTLQ/TAX_NUMBER_CHECK

- ✓ **PURPOSE:** Check Tax Numbers
- ✓ **DESCRIPTION:** This function module is a wrapper of 'TAX_NUMBER_CHECK' and checks the fields Tax Number 1 and Tax Number 2 using the rules defined in table T005.

The following additional checks are made against the motor vehicle country key:

Country	Code	Tax Number 1	Tax Number 2
Argentina	RA	×	
Belgium	B	×	
Brazil	BRA	×	×
Denmark	DK		×
France	F	×	
Italy	I	×	×
Norway	N		×
Portugal	P	×	
Spain	E	×	
Sweden	S		×

Following is the interface for the function module

- ✓ **Import Parameters**

1. **COUNTRY:** Country Key
2. **NATURAL_PERSON_FLAG:** Natural Person
3. **REGION:** Region
4. **STKZU:** Country-Specific Check
5. **TAX_CODE_1:** Tax Number 1
6. **TAX_CODE_2:** Tax Number 2
7. **TYPE_OF_TAX_CODE_1:** Tax Number Type

8. **TAX_CODE_3:** Tax Number 3
9. **TAX_CODE_4:** Tax Number 4

- ✓ **Export Parameters**

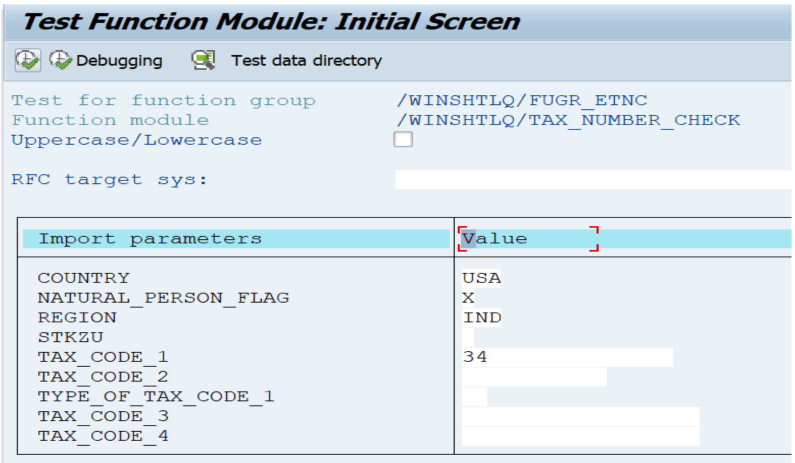
1. **MESSAGE:** Displays error message if any.

- ✓ **Exceptions**

1. **NOT_VALID:** This exception occurs if entered Natural number is invalid.
2. **NOT_AUTHORIZED_MSG_GETDETAIL:** This exception occurs if the user is unauthorized to run BAPI_MESSAGE_GETDETAIL FM.
3. **NOT_AUTHORIZED_TAX_NUM_CHK:** This exception occurs if the user is unauthorized to run /WINSHTLQ/TAX_NUMBER_CHECK FM.
4. **NO_RELEASE_FOUND:** This exception occurs if no SAP BASIS release is found in the SAP system.
5. **DIFFERENT_FPRCD:** This exception occurs if Different Provincial Tax Codes are entered in STCD1 and STCD2 fields.

Example:

Enter Input as below:

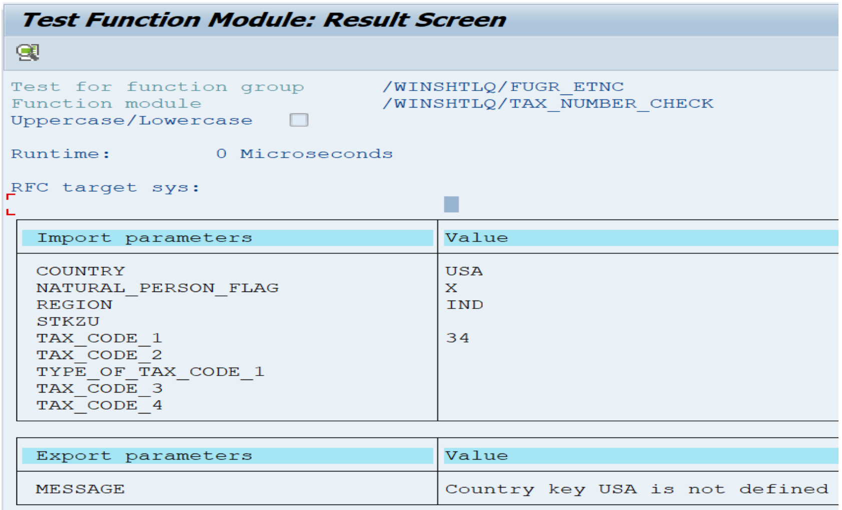


Test Function Module: Initial Screen

Test for function group	/WINSHTLQ/FUGR_ETNC
Function module	/WINSHTLQ/TAX_NUMBER_CHECK
Uppercase/Lowercase	□
RFC target sys:	

Import parameters	Value
COUNTRY	USA
NATURAL_PERSON_FLAG	X
REGION	IND
STKZU	
TAX_CODE_1	34
TAX_CODE_2	
TYPE_OF_TAX_CODE_1	
TAX_CODE_3	
TAX_CODE_4	

If the entered details are valid, no error message will be displayed. Otherwise, error message will be displayed.



Test Function Module: Result Screen

```
Test for function group        /WINSHTLQ/FUGR_ETNC
Function module                /WINSHTLQ/TAX_NUMBER_CHECK
Uppercase/Lowercase          □

Runtime:          0 Microseconds

RFC target sys:
```

Import parameters	Value
COUNTRY	USA
NATURAL_PERSON_FLAG	X
REGION	IND
STKZU	
TAX_CODE_1	34
TAX_CODE_2	
TYPE_OF_TAX_CODE_1	
TAX_CODE_3	
TAX_CODE_4	

Export parameters	Value
MESSAGE	Country key USA is not defined

6. /WINSHTLQ/POSTAL_CODE_CHECK

- ✓ **PURPOSE:** Check of postal code and region
- ✓ **DESCRIPTION:** This function module is a wrapper of ‘POSTAL_CODE_CHECK’ and checks the entered postal code and region.

Following is the interface for the function module

✓ **Import Parameters**

1. **COUNTRY:** Country Key
2. **ONE_TIME_ACCOUNT:** Indicator for one-time transactions (one-time account)
3. **POSTAL_CODE:** Postal code
4. **POSTAL_CODE_PO_BOX:** P.O. Box postal code
5. **PO_BOX:** PO Box
6. **REGION:** Region (federal state, state, province, county)

✓ **Export Parameters**

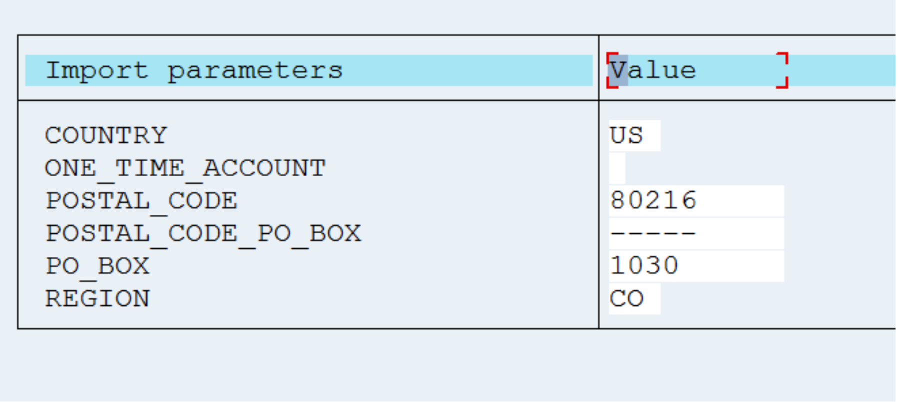
1. **POSTAL_CODE:** Postal code
2. **EV_POSTAL_CODE:** Postal code
3. **EV_POSTAL_CODE_PO_BOX:** P.O. Box Postal Code
4. **MESSAGE:** Displays error message if any.

✓ **Exceptions**

1. **NOT_VALID:** This exception occurs if entered postal code and region is invalid.
2. **NOT_AUTHORIZED_MSG_GETDETAIL:** This exception occurs if the user is unauthorized to run BAPI_MESSAGE_GETDETAIL FM.
3. **NOT_AUTH_POSTAL_CODE_CHECK:** This exception occurs if the user is unauthorized to run /WINSHTLQ/POSTAL_CODE_CHECK FM.
4. **NO_RELEASE_FOUND:** This exception occurs if no SAP BASIS release is found in the SAP system.

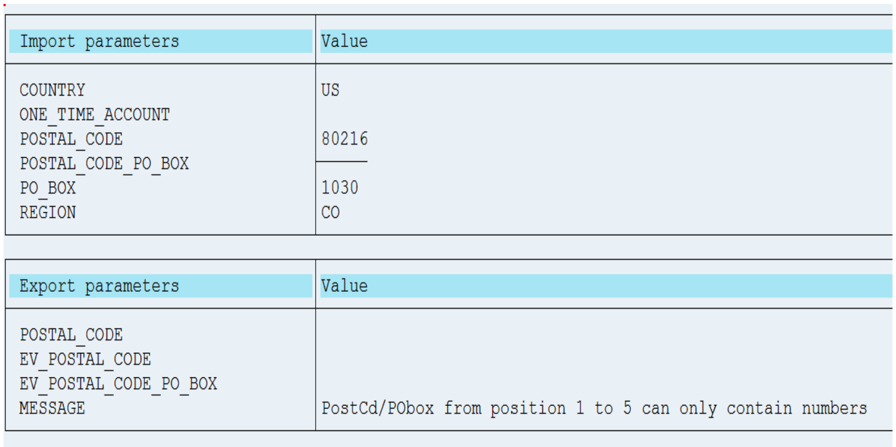
Example:

Enter Input as below:



Import parameters	Value
COUNTRY	US
ONE_TIME_ACCOUNT	
POSTAL_CODE	80216
POSTAL_CODE_PO_BOX	-----
PO_BOX	1030
REGION	CO

If the entered details are valid, no error message will be displayed. Otherwise, error message will be displayed.



Import parameters	Value
COUNTRY	US
ONE_TIME_ACCOUNT	
POSTAL_CODE	80216
POSTAL_CODE_PO_BOX	_____
PO_BOX	1030
REGION	CO

Export parameters	Value
POSTAL_CODE	
EV_POSTAL_CODE	
EV_POSTAL_CODE_PO_BOX	
MESSAGE	PostCd/POBox from position 1 to 5 can only contain numbers

7. /WINSHTLQ/RPC_CALCULATE_PRICES

- ✓ **PURPOSE:** WFM to Request Calculations and Adjust Requested Calculation
- ✓ **DESCRIPTION:** This BAPI (wrapper of 'BAPI_RPC_CALCULATE_PRICES') or the CalculatePrices method in the RetailPricing business object, facilitates communication between the sales price calculation function of the SAP System and one or more external systems.

You can use this BAPI to request and adjust calculations in external systems:

- **Requesting a Pricing Item**

Every pricing item required must be requested precisely once using this BAPI. Requesting a pricing item causes the system to suggest all the relevant pricing data (such as markup, margins and sales prices, for example).

- **Adjusting a Requested Pricing Item**

If the pricing items requested do not meet your expectations - for example, if the sales price suggested by the system is either too high or too low, the pricing item can be adjusted using this BAPI.

Depending on whether a calculation is to be requested or whether a requested calculation is to be adjusted, the interface tables CALCITEMIN and CALCITEMINX can be filled in different ways. You will find further information on this in the documentation for the individual tables.

- **Results of the Calculation**

Each time the BAPI is called up, the result of the calculation is entered in the CALCITEMOUT export table. Any errors that may have occurred are displayed in the RETURN export table. You will find further information in the interface documentation for the relevant tables.

- **Customer-Specific Enhancements**

Further fields that are not contained in the CALCITEMIN, CALCITEMINX or CALCITEMOUT interface tables can be entered using the enhancement tables EXTENSIONIN and EXTENSIONOUT and the BAPI. You can find further information on this in the interface documentation for the individual tables.

Following is the interface for the function module

✓ **Import Parameters**

1. **RPCCONTROLIN:** Control Sales Price Calculation for BAPI

✓ **Tables**

1. **CALCITEMIN:** BAPI Structure for Pricing Item (Import)
2. **CALCITEMINX:** Change BAPI Reference Structure for Structure BAPICALCITEMIN
3. **EXTENSIONIN:** Ref. structure for BAPI parameter ExtensionIn/ExtensionOut
4. **CALCITEMOUT:** BAPI Structure for Pricing Item (Export)
5. **EXTENSIONOUT:** Ref. structure for BAPI parameter ExtensionIn/ExtensionOut
6. **RETURN:** Return Parameter

✓ **Exceptions**

1. **NOT_AUTHORIZED_CALC_PRICE:** This exception occurs if the user is unauthorized to run /WINSHTLQ/RPC_CALCULATE_PRICES FM.
2. **NOT_AUTHORIZED_BAPI_CALC_PRICE:** This exception occurs if the user is unauthorized to run BAPI_RPC_CALCULATE_PRICES FM.
3. **NO_RELEASE_FOUND:** This exception occurs if no SAP BASIS release is found in the SAP system.

Example:

Enter Input as below:

- **Table: CALCITEMIN**

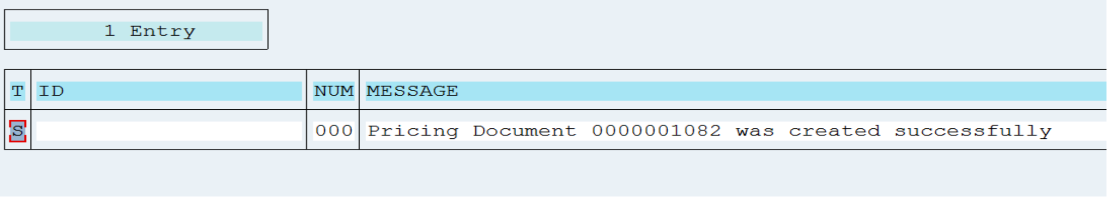
SALES_ORG = ‘0001’.
DISTR_CHAN = ‘01’.
MATERIAL = ‘100-100’.
SP_VALID_FROM = ’29.01.2018 ‘.
SP_VALID_TO = ’29.01.2018 ‘.
SALES_CURRENCY = ‘AUD’.
FINAL_PRICE = ’10’.
SP_MARKED_FOR_ACTIVATION = ‘X’.

- **Table: CALCITEMINX**

SALES_ORG = ‘0001’.
DISTR_CHAN = ‘01’.
MATERIAL = ‘100-100’.
FINAL_PRICE = ’X’.
SP_MARKED_FOR_ACTIVATION = ‘X’.

Output:

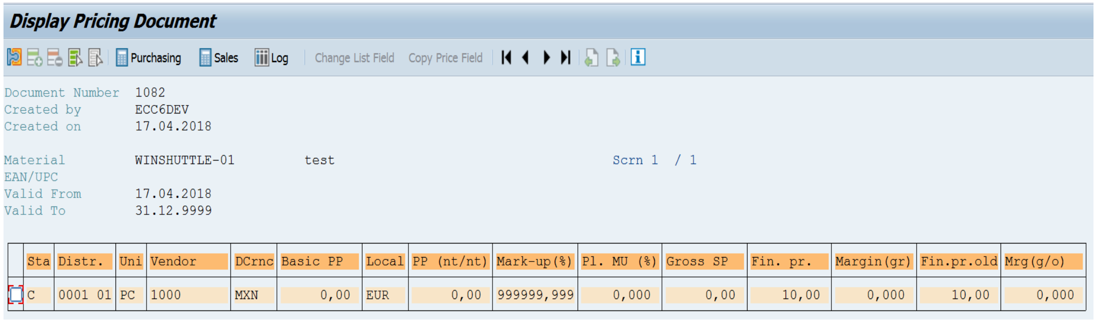
Return Table:



1 Entry

T	ID	NUM	MESSAGE
S		000	Pricing Document 0000001082 was created successfully

Created Document can be seen in tcode – VKP7



Display Pricing Document

```
Document Number  1082
Created by       ECC6DEV
Created on       17.04.2018

Material         WINSHUTTLE-01        test                        Scrn 1  / 1
EAN/UPC
Valid From       17.04.2018
Valid To         31.12.9999
```

	Sta	Distr.	Uni	Vendor	DCrnc	Basic PP	Local	PP (nt/nt)	Mark-up(%)	Pl. MU (%)	Gross SP	Fin. pr.	Margin(gr)	Fin.pr.old	Mrg(g/o)
□	C	0001 01	PC	1000	MXN	0,00	EUR	0,00	999999,999	0,000	0,00	10,00	0,000	10,00	0,000

8. /WINSHTLQ/TRN_TAX_TXJCD_TABLE

- ✓ **PURPOSE:** WFM for Tax Jurisdiction Code Determination
- ✓ **DESCRIPTION:** This function is intended for all entities (including most R/3 master records such as plant, customer, vendor, cost center, asset, etc.) that use the Central Address Management (CAM) to store their addresses.

This function has two features:

- If a jurisdiction code is not specified, this function determines a unique jurisdiction code from the given address: Country, state, Zipcode, and possibly city and/or county.

- If a jurisdiction code is specified, the function checks that this jurisdiction code is valid. To check if a jurisdiction code is valid, the country tax procedure needs to be determined to establish whether an external tax system is being used. If an external system is in use, the specified jurisdiction code will be checked against the specified address (state, zipcode, etc). Alternatively, the specified jurisdiction code will be checked against table TTXJ and the corresponding jurisdiction code text will also be returned.

Tax Jurisdiction Code: Location identification for tax purposes. The combination of country, state, county, city and zipcode produces a unique jurisdiction code. The jurisdiction code is used by most US tax packages to simplify the US tax calculation process in which the place of origin and destination are of major significance. For this reason, a jurisdiction code usually belongs to an entity.

Following is the interface for the function module

✓ **Import Parameters**

1. **IM_COUNTRY:** Country Key
2. **IM_REGION:** Region (State, Province, County)
3. **IM_ZIPCODE: City** postal code
4. **IM_CITY:** City
5. **IM_COUNTY:** District

✓ **Tables**

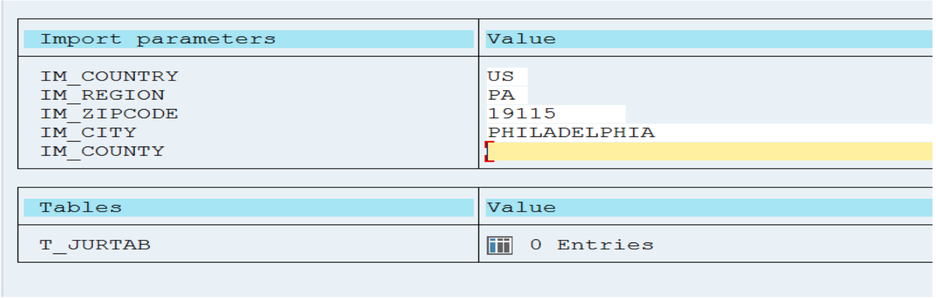
1. **T_JURTAB:** US tax interface: Communication structure for TXJCD determ.

✓ **Exceptions**

1. **NOT_AUTHORIZED_TRN_TAX_TXJCD:** This exception occurs if the user is unauthorized to run /WINSHTLQ/TRN_TAX_TXJCD_TABLE FM.
2. **NO_RELEASE_FOUND:** This exception occurs if no SAP BASIS release is found in the SAP system.

Example:

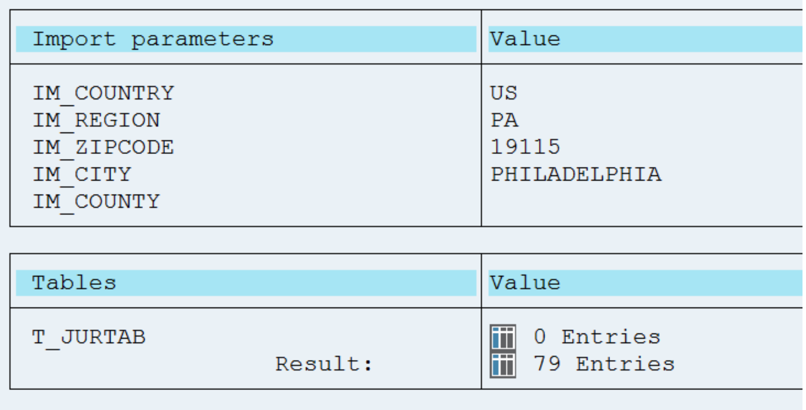
Enter Input as below:



Import parameters	Value
IM_COUNTRY	US
IM_REGION	PA
IM_ZIPCODE	19115
IM_CITY	PHILADELPHIA
IM_COUNTY	

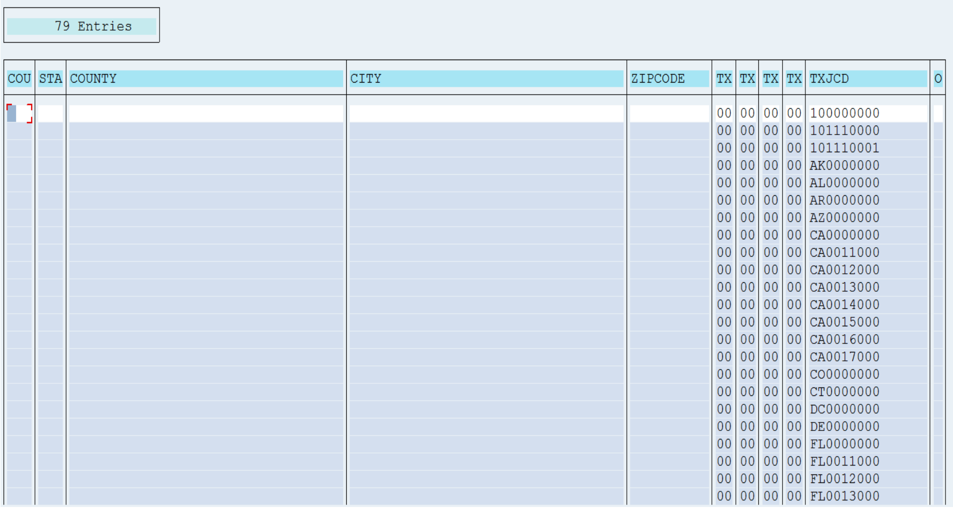
Tables	Value
T_JURTAB	0 Entries

Output:



Import parameters	Value
IM_COUNTRY	US
IM_REGION	PA
IM_ZIPCODE	19115
IM_CITY	PHILADELPHIA
IM_COUNTY	

Tables	Value
T_JURTAB Result:	0 Entries 79 Entries



79 Entries

COU	STA	COUNTY	CITY	ZIPCODE	TX	TX	TX	TX	TXJCD	O
					00	00	00	00	100000000	
					00	00	00	00	101110000	
					00	00	00	00	101110001	
					00	00	00	00	AK0000000	
					00	00	00	00	AL0000000	
					00	00	00	00	AR0000000	
					00	00	00	00	AZ0000000	
					00	00	00	00	CA0000000	
					00	00	00	00	CA0011000	
					00	00	00	00	CA0012000	
					00	00	00	00	CA0013000	
					00	00	00	00	CA0014000	
					00	00	00	00	CA0015000	
					00	00	00	00	CA0016000	
					00	00	00	00	CA0017000	
					00	00	00	00	CO0000000	
					00	00	00	00	CT0000000	
					00	00	00	00	DC0000000	
					00	00	00	00	DE0000000	
					00	00	00	00	FL0000000	
					00	00	00	00	FL0011000	
					00	00	00	00	FL0012000	
					00	00	00	00	FL0013000	

9. /WINSHTLQ/SD_CUST_MAINTAIN_ALL

- ✓ **PURPOSE:** Maintain Customer Master data
- ✓ **DESCRIPTION:** This FM is a wrapper of ‘SD_CUSTOMER_MAINTAIN_ALL’ and is used to maintain General, Sales and Company code data for Customers.

Following is the interface for the function module

- ✓ **Import Parameters**

1. **I_KNA1:** Maintain General data in Customer Master
2. **I_KNB1:** Maintain Customer Master (Company Code) data
3. **I_KNVV:** Maintain Customer Master Sales Data
4. **I_BAPIADDR1:** BAPI Reference Structure for Addresses (Org./Company)
5. **I_BAPIADDR2:** BAPI reference structure for addresses (nat. persons)
6. **I_MAINTAIN_ADDRESS_BY_KNA1:** This flag should be 'X' if address is maintained in I_KNA1 table
7. **I_KNB1_REFERENCE**
8. **I_FORCE_EXTERNAL_NUMBER_RANGE:**
9. **I_NO_BANK_MASTER_UPDATE**
10. **I_CUSTOMER_IS_CONSUMER**
11. **I_RAISE_NO_BTE**
12. **PI_POSTFLAG**
13. **PI_CAM_CHANGED**
14. **PI_ADD_ON_DATA:** Add-On Data Changed in Customer Master
15. **I_FROM_CUSTOMERMASTER**

✓ **Tables**

1. **T_JURTAB:** US tax interface: Communication structure for TXJCD determination.

✓ **Exceptions:**

1. **NOT_AUTHORIZED_TRN_TAX_TXJCD:** This exception occurs if the user is unauthorized to run /WINSHTLQ/TRN_TAX_TXJCD_TABLE FM.
2. **NO_RELEASE_FOUND:** This exception occurs if no SAP BASIS release is found in the SAP system.

Example:

Case 1: To Maintain General data in Customer Master, Enter input as below

Exporting Parameters:

I_MAINTAIN_ADDRESS_BY_KNA1	X
PI_POSTFLAG	X

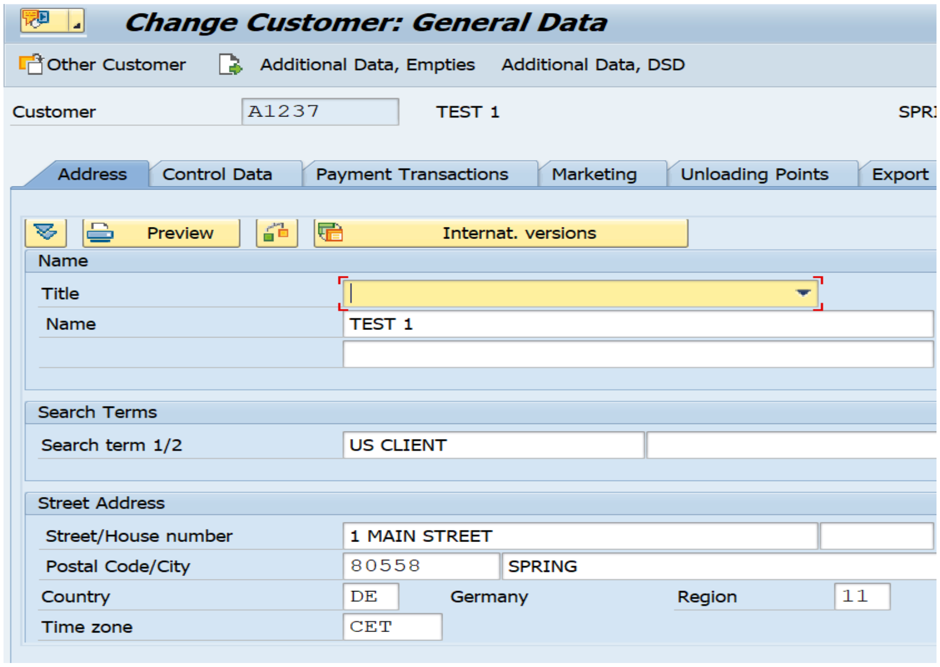
Structure: I_KNA1	**Values**
I_KNA1-KUNNR	A1237
I_KNA1-MANDT	800
I_KNA1-LAND1	DE
I_KNA1-NAME1	TEST 1
I_KNA1-ORT01	SPRING
I_KNA1-PSTLZ	80558
I_KNA1-REGIO	11
I_KNA1-SORTL	US CLIENT
I_KNA1-STRAS	1 MAIN STREET
I_KNA1-MCOD3	GRAZ

I_KNA1-KTOKD	0160
I_KNA1-SPRAS	EN
I_KNA1-LZONE	0000000001
I_KNA1-NAME2	Kumar
I_KNA1-NAME3	Verma
I_KNA1-NAME4	Sharma
I_KNA1-MCOD1	GRAZ
I_KNA1-MCOD2	GRAZ
I_KNA1-PFACH	12
I_KNA1-PSTL2	SW1A 2SW
I_KNA1-BEGRU	0004
I_KNA1-STCD1	3456-56-789
I_KNA1-STCEG	DE-12345679

Structure: I_BAPIADDR1	**Values**
I_BAPIADDR1 -POSTL_COD1	80558
I_BAPIADDR1- STR_SUPPL1	STREET 1
I_BAPIADDR1- STR_SUPPL2	STREET 2
I_BAPIADDR1- STR_SUPPL3	STREET 3
I_BAPIADDR1-LOCATION	STREET 4
I_BAPIADDR1-TEL1_NUMBR	(215) 665 45
I_BAPIADDR1- FAX_NUMBER	2012020202
I_BAPIADDR1- EMAIL	Ramkumar4221@gmail.com
I_BAPIADDR1- COMM_TYPE	RML
I_BAPIADDR1- NAME	RUMI
I_BAPIADDR1- NAME_2	
I_BAPIADDR1- NAME_3	
I_BAPIADDR1- NAME_4	
I_BAPIADDR1- SORT1	Search term 1
I_BAPIADDR1- SORT2	Search term 2
I_BAPIADDR1- CITY	SPRING
I_BAPIADDR1- COUNTRY	DE
I_BAPIADDR1- LANGU	EN

Data saved successfully.

Output:



Customer A1237 TEST 1 SPRI

Name

Title

Name TEST 1

Search Terms

Search term 1/2 US CLIENT

Street Address

Street/House number 1 MAIN STREET

Postal Code/City 80558 SPRING

Country DE Germany Region 11

Time zone CET

Case 2: To Maintain data of Customer Company Code, Enter input as below

Input values as below:

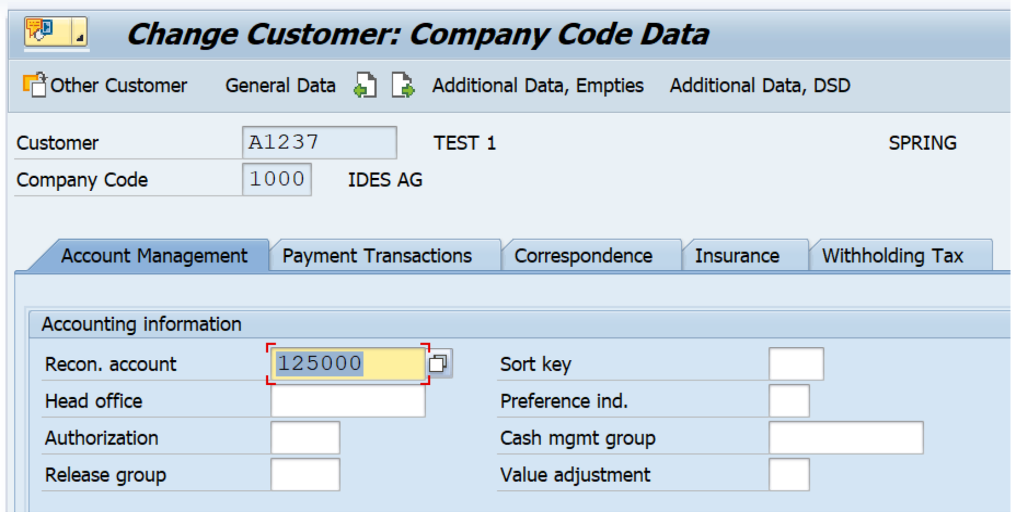
Exporting Parameters:

I_MAINTAIN_ADDRESS_BY_KNA1	X
PI_POSTFLAG	X

Structure: I_KNB1	**Values**
I_KNB1-BUKRS	1000
I_KNB1-AKONT	125000
I_KNB1-ZTERM	0001
I_KNB1-TOGRU	1000
I_KNB1-XZVER	X
I_KNB1-LOCKB	LCKBX
I_KNB1-BUSAB	D1
I_KNB1-KUNNR	A1237

Data saved successfully.

Output:



Change Customer: Company Code Data

Customer	A1237	TEST 1	SPRING
Company Code	1000	IDES AG	

Accounting information

Recon. account	125000	Sort key	
Head office		Preference ind.	
Authorization		Cash mgmt group	
Release group		Value adjustment	

Case 3: To Maintain data in Customer Sales Data, Enter input as below

Input values as below:

Exporting Parameters:

I_MAINTAIN_ADDRESS_BY_KNA1	X
PI_POSTFLAG	X

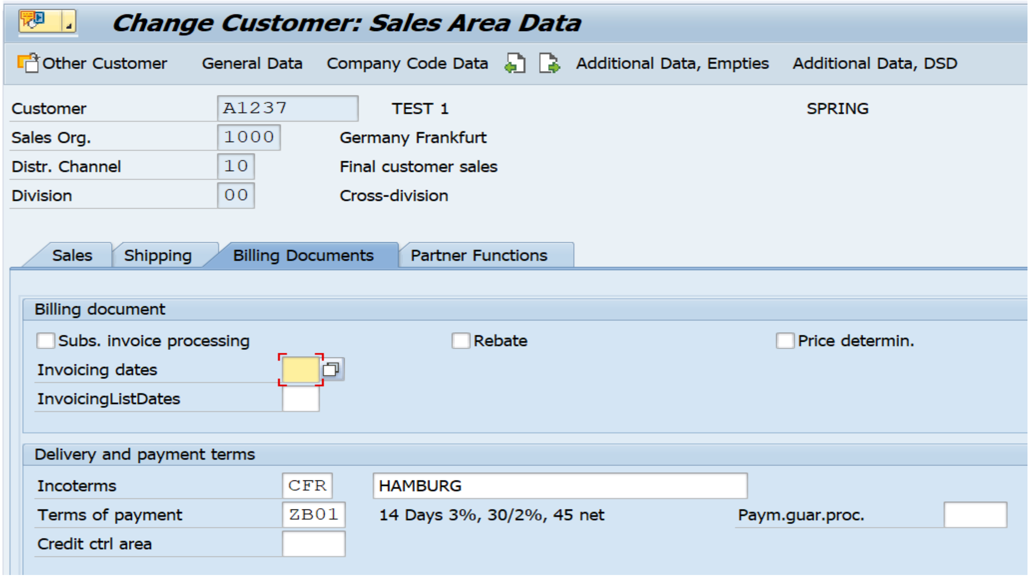
Structure: I_KNVV	**Values**
I_KNVV-VKORG	1000
I_KNVV-VTWEG	10
I_KNVV-SPART	00
I_KNVV-VKBUR	1000
I_KNVV-KLABC	
I_KNVV-PLTYP	
I_KNVV-VERSG	
I_KNVV-LPRIO	01
I_KNVV-VSBED	01
I_KNVV-KZTLF	
I_KNVV-ANTLF	9
I_KNVV-BOKRE	
I_KNVV-PERFK	
I_KNVV-PERRL	
I_KNVV-INCO1	CFR
I_KNVV-INCO2	HAMBURG
I_KNVV-ZTERM	ZB01
I_KNVV-KTGRD	

I_KNVV-ERNAM	ECC6DEV
I_KNVV-ERDAT	10.10.2018
I_KNVV-KUNNR	A1237

Table: T_XKNVI	**Values**
T_XKNVI-KUNNR	A1237
T_XKNVI-TAXKD	1
T_XKNVI-ALAND	DE
T_XKNVI-TATYP	MWST

Data saved successfully.

Output:



Change Customer: Sales Area Data

Customer	A1237	TEST 1	SPRING
Sales Org.	1000	Germany Frankfurt	
Distr. Channel	10	Final customer sales	
Division	00	Cross-division	

Billing document

□ Subs. invoice processing　□ Rebate　□ Price determin.

Invoicing dates

InvoicingListDates

Delivery and payment terms

Incoterms	CFR	HAMBURG		
Terms of payment	ZB01	14 Days 3%, 30/2%, 45 net	Paym.guar.proc.	
Credit ctrl area				

10. /WINSHTLQ/TRN_CREATE_TEXT

- ✓ **PURPOSE:** To maintain SO10 Long text
- ✓ **DESCRIPTION:** This function is used to maintain SO10 long texts.

Following is the interface for the function module

- ✓ **Import Parameters**

1. **FID:** TextID
2. **FLANGUAGE:** Language Key
3. **FNAME:** Name
4. **FOBJECT:** Texts: Application Object
5. **SAVE_DIRECT:** Default value is ‘X’. This is set to ‘X’ for saving the long text.
6. **TEXT:** Long text

7. **FFORMAT:** Default value is '*'. The tag column contains format keys which define the output formatting of the text or initiate control commands.

- A format key defines the beginning of a new paragraph and its formatting. All text lines directly after it which have blanks in this field belong to this paragraph. These lines are handled as continuous text lines and are always formatted by the SAPscript editor so that as many words as possible fit into an editor line. Blanks between words are considered. The editor formatting is not the same as the output formatting.

- The output formatting of a text is not visible until the text is output to a printer or displayed on the screen. Here, the format definitions of the paragraph format are evaluated.

- The format keys possible and their respective meanings are defined in styles or forms. If a style or form is assigned to a text module you can use the paragraph formats defined there to format your text. Format keys which can be defined by the user can consist of one or two characters. Only the letters A to Z and the numbers 0 to 9 are allowed. The paragraph format must always begin with a letter. If a format key is not contained in the assigned style or form, the default paragraph defined there is used instead.

  Some format keys are defined by SAPscript. They have a predefined meaning and can be used in all texts:

  *** Default paragraph**

  The format definitions which are specified for the paragraph defined in the assigned style or form as the default paragraph are used for the output formatting of the paragraph involved.

  / New line

  The subsequent text is written to a new line during output formatting. The formatting attributes of the last paragraph format are retained.

  /: SAPscript control command

  The characters contained in the actual text line are not output as text but are regarded as a control command. They are not interpreted or executed until output formatting of the text. Control commands must always fit into a line fully. Subsequent lines are not allowed.

  /* Comment line

  This line is not considered during output formatting of the text.

  = Extended line

  This line is not subject to the line formatting in the SAPscript editor. The text contained in this line is appended directly to the character of the preceding text line which was output last. If this is not required, there must be at least one blank at the beginning of the extended line.

  /= Like extended line

  But the subsequent text is placed in a new line during output formatting.

(Raw line

The subsequent editor line is not interpreted by the SAPscript composer during output formatting. This means that character formats, symbols, tab characters, masking characters, or hypertext links which may be contained in this line are not evaluated and are therefore passed unchanged to the output device. The text contained in this line is also appended directly to the character of the preceding text line which was output last. If this is not required, there must be at least one blank at the beginning of the extended line.

/(Like raw line

But the subsequent text is placed in a new line during output formatting.

>x Fixed line

The line is not ready for input in the SAPscript editor. It can also not be deleted or separated. You can only create fixed lines with a program. You can therefore give a text a fixed structure, for example, which cannot be changed by the user.
You can use any number or letter for the 'x'. You can therefore separate different sub-headings, for example. If several fixed lines occur consecutively with the same indicator, they are regarded as a unit by the SAPscript editor. It is not possible to insert anything between these lines in the editor.
In the case of fixed lines, SAPscript print formatting interprets the first two characters of the line as a paragraph format for formatting. You therefore need to enter the required paragraph format or blank here.

- ✓ **Tables**

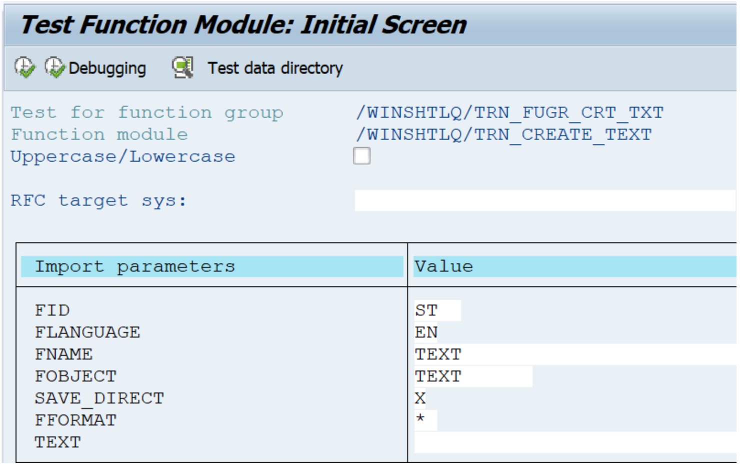
1. **FLINES:** Text lines which needs to be updated in Long text.
2. **MESSAGES:** Get return messages
3. **WINOPTIONS:** RFC Table Read: Select Options / WHERE Clause(Not in use as of now).

- ✓ **Exceptions:**

1. **NOT_AUTHORIZED_RFC_CREATE_TEXT:** This exception occurs if the user is unauthorized to run /WINSHTLQ/TRN_CREATE_TEXT FM.
2. **DATA_IS_IN_FLINES_N_TEXT_BOTH:** Either 'TEXT' or 'FLINES' can have input once, not both. This exception occurs if Long text is entered on parameter ‘TEXT or in Table ‘FLINES’.

Example:

Enter Input as below:

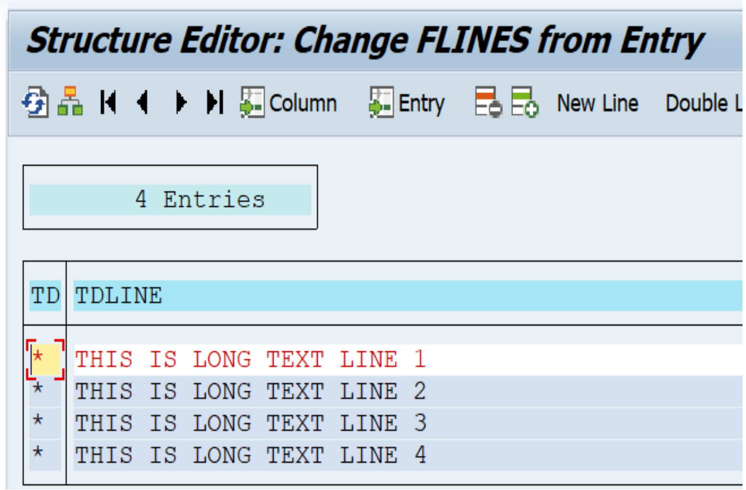


Test Function Module: Initial Screen

Test for function group	/WINSHTLQ/TRN_FUGR_CRT_TXT
Function module	/WINSHTLQ/TRN_CREATE_TEXT
Uppercase/Lowercase	□
RFC target sys:	

Import parameters	Value
FID	ST
FLANGUAGE	EN
FNAME	TEXT
FOBJECT	TEXT
SAVE_DIRECT	X
FFORMAT	*
TEXT	

Long Text Table:

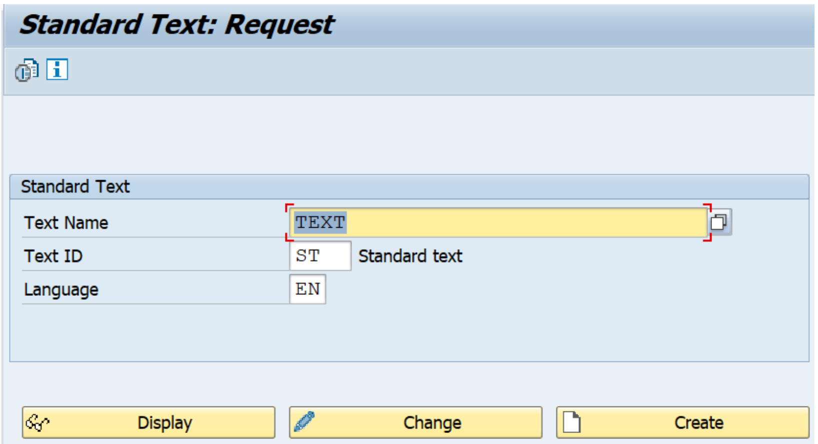


Structure Editor: Change FLINES from Entry

4 Entries

TD	TDLINE
*	THIS IS LONG TEXT LINE 1
*	THIS IS LONG TEXT LINE 2
*	THIS IS LONG TEXT LINE 3
*	THIS IS LONG TEXT LINE 4

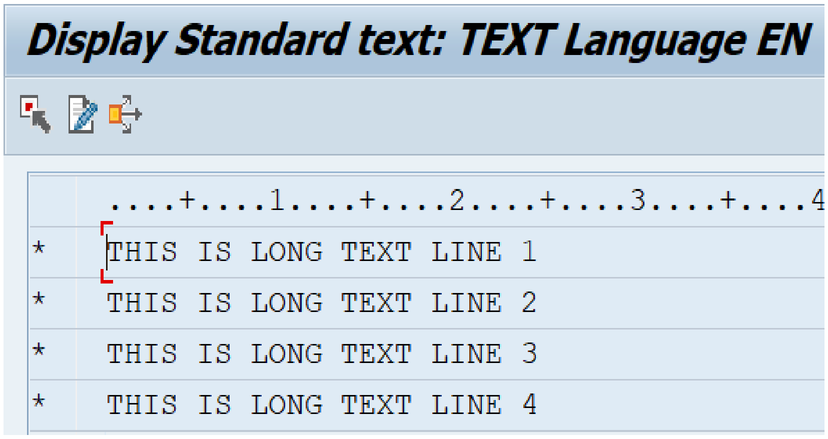
This Long text will be saved in SO10 Text.



Standard Text: Request

Standard Text

Text Name	TEXT	
Text ID	ST	Standard text
Language	EN	



Display Standard text: TEXT Language EN

```
....+....1....+....2....+....3....+....4
* THIS IS LONG TEXT LINE 1
* THIS IS LONG TEXT LINE 2
* THIS IS LONG TEXT LINE 3
* THIS IS LONG TEXT LINE 4
```

11. /WINSHTLQ/MATERIAL_GET_NEXT

- ✓ **PURPOSE**: To get next Material number available for the entered Material Type.
- ✓ **DESCRIPTION:** This function is used to get next Material number available for the entered Material Type.

Following is the interface for the function module

✓ **Import Parameters**

1. **MATERIALART:** Material Type

✓ **Export Parameters**

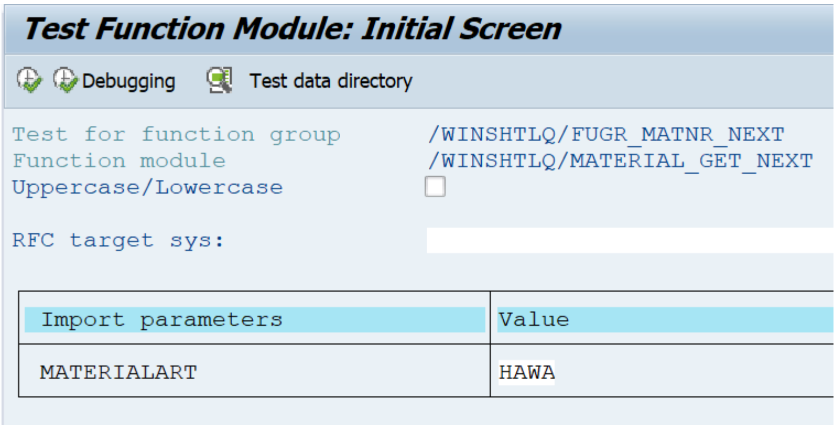
1. **MATERIALNR:** Get Material Number
2. **RETURNCODE:** Get Error messages
3. **E_RETURN:** Get Success message in this field if the FM has run successfully.

✓ **Exceptions:**

1. **NO_AUTHORIZED_TO_CREATE:** This exception occurs if the user is unauthorized to generate next Material Number.
2. **NO_RELEASE_FOUND:** This exception occurs if no SAP BASIS release is found in the SAP system.
3. **NO_INTERNAL_INTERVAL:** This exception occurs if no Internal interval is defined in the SAP for the entered Material Type.
4. **TYPE_NOT_FOUND:** This exception occurs if the entered Material type is not found in the SAP.

Example:

Enter Input as below:



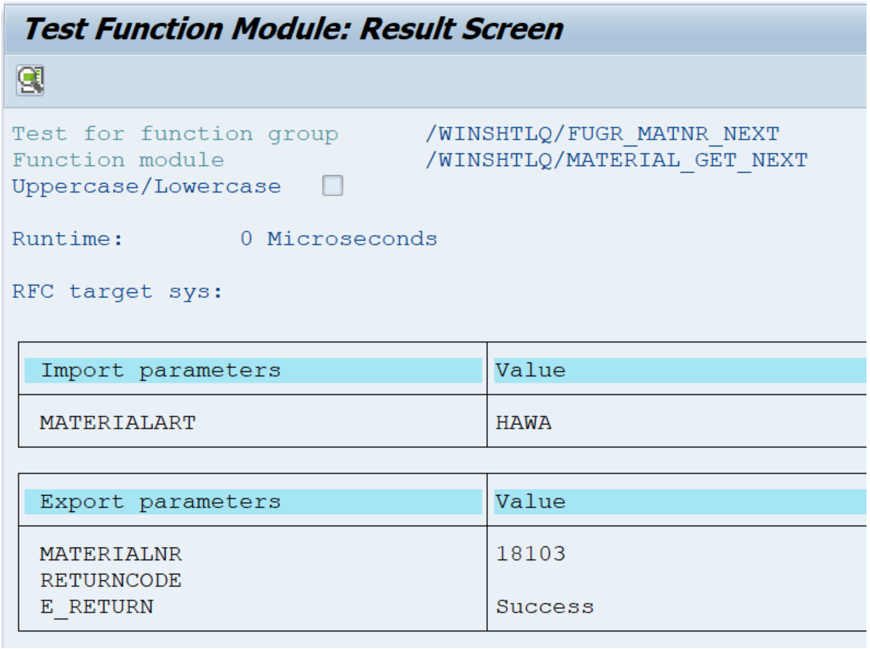
Test Function Module: Initial Screen

Test for function group /WINSHTLQ/FUGR_MATNR_NEXT
Function module /WINSHTLQ/MATERIAL_GET_NEXT
Uppercase/Lowercase □

RFC target sys:

Import parameters	Value
MATERIALART	HAWA

You will get next material number available for entered Material Type.



Test Function Module: Result Screen

Test for function group /WINSHTLQ/FUGR_MATNR_NEXT
Function module /WINSHTLQ/MATERIAL_GET_NEXT
Uppercase/Lowercase □

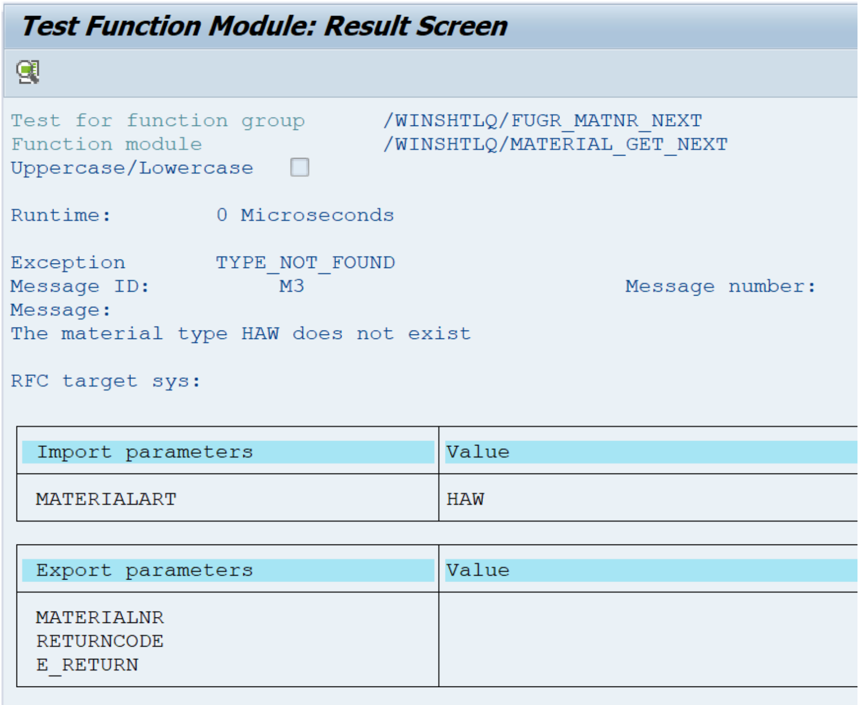
Runtime: 0 Microseconds

RFC target sys:

Import parameters	Value
MATERIALART	HAWA

Export parameters	Value
MATERIALNR	18103
RETURNCODE	
E_RETURN	Success

If the entered Material Type is invalid, it will display message as below:



Test Function Module: Result Screen

```
Test for function group       /WINSHTLQ/FUGR_MATNR_NEXT
Function module               /WINSHTLQ/MATERIAL_GET_NEXT
Uppercase/Lowercase     □

Runtime:         0 Microseconds

Exception        TYPE_NOT_FOUND
Message ID:           M3                        Message number:
Message:
The material type HAW does not exist

RFC target sys:
```

Import parameters	Value
MATERIALART	HAW

Export parameters	Value
MATERIALNR RETURNCODE E_RETURN	

12. /WINSHTLQ/ACC_DOC_COPA_POST

- ✓ **PURPOSE:** Accounting: Posting
- ✓ **DESCRIPTION:** This function is used for accounting posting.

Following is the interface for the function module

✓ **Import Parameters**

1. DOCUMENTHEADER: Enter Header details like company code, posting dates, Fiscal year etc.

2. CUSTOMERCPD: Enter One-time Customer details like Name, Address, Bank account number etc.

3. CONTRACTHEADER: Enter Additional Contract Accounts Receivable and Payable Header Line details like Doc no., Doc. Type, Reconciliation Key for General Ledger Accounting.

✓ **Export Parameters**

1. OBJ_TYPE: This parameter will return the Reference Transaction number

2. OBJ_KEY: This parameter will return the Reference Key

3. OBJ_SYS: This parameter will return the Logical system of source document

✓ **Tables**

1. **ACCOUNTGL:** This Table is used for transferring G/L account line items

The following fields must be filled out:

1. **GL_ACCOUNT**: The currency fields are not contained in this parameter. They must be transferred separately in the parameter CurrencyAmount for the various currency types and currencies. The system creates a link to the relevant line item via the line item number ITEMNO_ACC, which for this reason must be unique for each line item.

   DEFAULT: If this method can be used to post G/L account line items across several company codes, the following fields are also contained in the G/L account line items:

COMP_CODE	Company code
PSTNG_DATE	Posting date
DOC_TYPE	Document type
AC_DOC_NO	Document number
FISC_YEAR	Fiscal year
FIS_PERIOD	Fiscal period

   If the fields are blank, the information is copied from the document header (parameter **Document header**).

2. **ACCOUNTRECEIVABLE:** This table is used for transferring customer line items.

   If the CUSTOMER number is not given, only the general ledger is updated. In this case, an entry must be made in the field GL_ACCOUNT (G/L account for general ledger accounting).

   The currency fields are not contained in this parameter. They must be transferred separately in the parameter CurrencyAmount for the various currency types and currencies. The system creates a link to the relevant line item via the line item number ITEMNO_ACC, which for this reason must be unique for each line item.

3. **ACCOUNTPAYABLE:** This table is used for transferring vendor line items.

   If the vendor account number VENDOR_NO is not given, only the general ledger is updated. In this case, an entry must be made in the field GL_ACCOUNT (G/L account for general ledger accounting).

   The currency fields are not contained in this parameter. They must be transferred separately in the parameter CurrencyAmount for the various currency types and currencies. The system creates a link to the relevant line item via the line item number ITEMNO_ACC, which for this reason must be unique for each line item.

4. **ACCOUNTTAX:** This table is used for transferring tax lines.

   The following fields must be filled:

ITEMNO_ACC	Item number

TAX_CODE	Tax code
COND_KEY	Condition type

If the Jurisdiction Code is active, these fields must also be filled.
The currency fields are not contained in this parameter. They must be transferred in the parameter Currency Amount for the different currency types or currencies. A connection is created to the corresponding document line item via the item number ITEMNO_ACC which must be unique for all line items.

5. **CURRENCYAMOUNT:** This parameter describes the currency fields for a particular line item. The parameter is a required field for the method POST. If the parameter is not maintained for the method CHECK, only a general account assignment check is carried out. Individual line items or several documents can be checked at once in this way.

The following fields must be filled out:

CURRENCY	Currency key
AMT_DOCCUR	Currency amount

Value range

The field CURR_TYPE (currency type) describes the role played by a currency in the SAP System.
The currency type can take on the following values, for example:

00	Transaction currency
10	Company code currency
20	Controlling area currency
30	Group currency
40	Hard currency
50	Index-based currency
60	Global company currency

If an entry is not made in the field CURR_TYPE, the default value "00" is retained.

If only one data record is transferred in transaction currency for each line item, the field CURR_TYPE does not need to be filled out. If several data records are transferred for each line item, the currency type must be entered for each record.

6. **CRITERIA:** CO-PA Account Assignment Characteristics

You can use this parameter to transfer the Profitability Analysis (CO-PA) characteristics to an account assignment. The system creates a link to the relevant line item via the line item number ITEMNO_ACC, which for this reason must be unique for each line item.

Note: The contents of the parameter must be sorted before being transferred. If not, errors may occur when assigning error messages to parameter lines.

7. **VALUEFIELD:** CO-PA Account Assignment Value Fields

This parameter can be used to transfer the value fields for the profitability analysis (CO-PA) for an accounting assignment.

The system creates a link to the relevant line item via the line item number ITEMNO_ACC, which therefore must be unique for each line item.

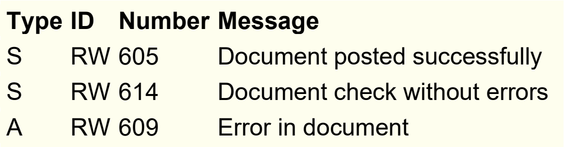
Note: The parameter contents must be sorted when transferred, otherwise errors may occur when assigning error messages to the parameter lines.

8. **EXTENSION1:** Container for 'Customer Exit' Parameter

This exit can be used to edit the accounting document before it is transferred to the accounting components for the update.

9. **RETURN:** This table contains the return messages from the method call.

Value range: The first message contains the general status of processing:



Type	ID	Number	Message
S	RW	605	Document posted successfully
S	RW	614	Document check without errors
A	RW	609	Error in document

The fields MESSAGE_V1, MESSAGE_V2 and MESSAGE_V3 contain the following information:

- Reference transaction (OBJ_TYPE)
- Reference key (OBJ_KEY)
- Logical system of source document (OBJ_SYS)

Additional messages are transferred with the following information:

- Parameter (PARAMETER)
- Row (ROW)
- Field (FIELD) in which the error occurred.

10. **PAYMENTCARD**: In this parameter data for credit card payment can be transferred.

Link to debitor line:
The item is also transferred to the debitor line (ACCOUNTRECEIVABLE) and must therefore contain the same item number (ITEMNO_ACC).

11. **CONTRACTITEM:** Additional Contract Accounts Receivable and Payable Document Line Item.

In this parameter fields are transferred that are concerned with an accounting document not including the contract accounts receivable and payable (FI-CA) at the item level.

Link to debitor line:

The item is also transferred to the debitor line (ACCOUNTRECEIVABLE) and must therefore contain the same item number (ITEMNO_ACC).

12. **EXTENSION2:** Reference Structure for BAPI Parameters EXTENSIONIN/EXTENSIONOUT

This enhancement structure can be used to transfer additional fields.

There are four fields available for each data record (VALUEPART1 - 4), which each contain 240 characters. A defined ABAP dictionary structure can be transferred in the field STRUCTURE, based on which the field contents are read.

The accounting document can be supplemented and changed in an implemented BADI (ACC_DOCUMENT) before the active accounting component is called.

Default:

The class CL_EXM_IM_ACC_DOCUMENT is available as example implementation. Here, for each data record of the parameter that is transferred the ABAP dictionary structure is checked and the content of the fields VALUEPART1 to -4 is set in the structure. If the structure that is transferred contains a field POSNR, the associated lines of the accounting document are updated using the contents of the enhancement.

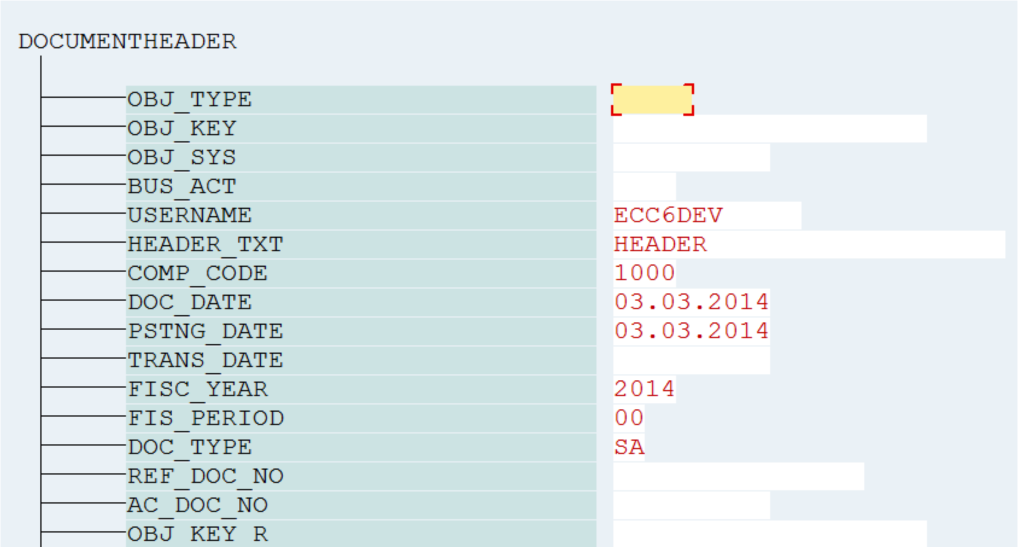
13. **REALESTATE:** Real Estate Account Assignment Data

This parameter can be used to transfer account assignments for Real Estate Management. The current line represents an addition to the accounting document line items (ACCOUNTGL, ACCOUNTRECEIVABLE, ACCOUNTPAYABLE). The line in parameter REALESTATE must be connected to the document line item (ITEMNO_ACC) using the line number.

14. **ACCOUNTWT:** This table is used for Withholding Tax Information like ITEMNO_ACC, Withholding Tax Base Amount (Local Currency), Withholding Tax Base Amount (Document Currency), **Withholding tax** code etc.

Example:

Enter Input as below:



DOCUMENTHEADER

OBJ_TYPE	
OBJ_KEY	
OBJ_SYS	
BUS_ACT	
USERNAME	ECC6DEV
HEADER_TXT	HEADER
COMP_CODE	1000
DOC_DATE	03.03.2014
PSTNG_DATE	03.03.2014
TRANS_DATE	
FISC_YEAR	2014
FIS_PERIOD	00
DOC_TYPE	SA
REF_DOC_NO	
AC_DOC_NO	
OBJ_KEY_R	

- **AccountGl table:**

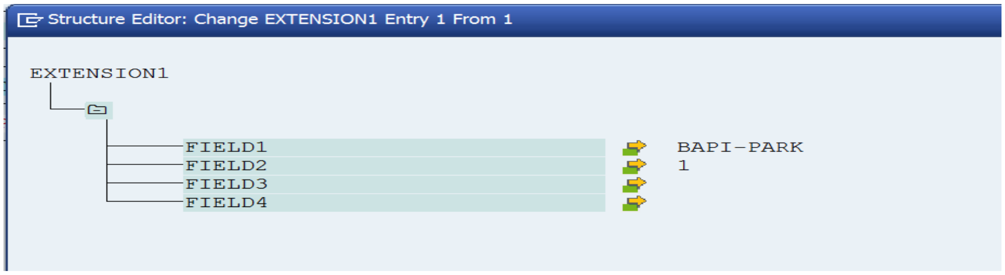
  ITEMNO_ACC = ‘0000000001’.
  GL_ACCOUNT = ‘473000 ‘.
  ITEM_TEXT = ‘LINE 1’.
  COSTCENTER = ‘3010’.
  ITEMNO_ACC = ‘0000000002’.
  GL_ACCOUNT = ‘405200 ‘.
  ITEM_TEXT = ‘LINE 2’.
  COSTCENTER = ‘3020’.

- **CURRENCYAMOUNT Table:**

  ITEMNO_ACC = ‘0000000001’.
  CURR_TYPE = ‘00’.
  CUURENCY = ‘EUR’.
  AMT_DOCCUR = ‘100’.

  ITEMNO_ACC = ‘0000000002’.
  CURR_TYPE = ‘00’.
  CUURENCY = ‘EUR’.
  AMT_DOCCUR = ‘-100’.

- **EXTENSION1 Structure:**

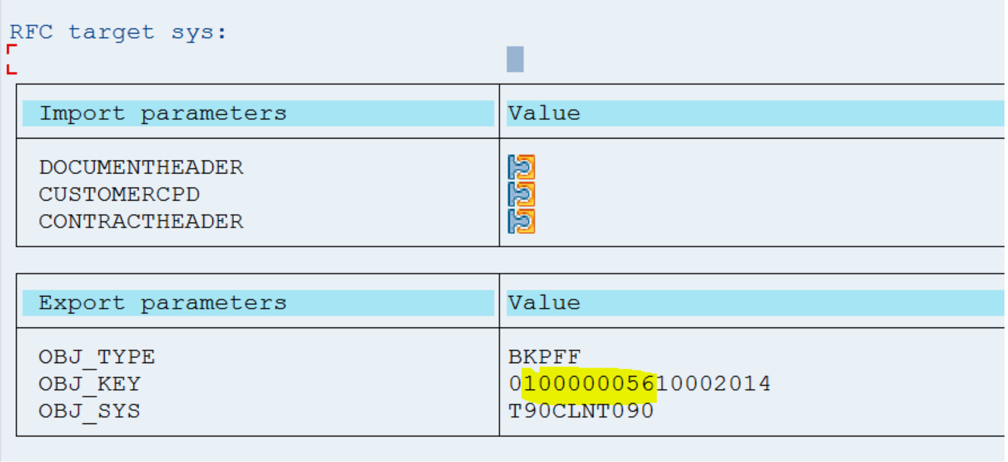


Structure Editor: Change EXTENSION1 Entry 1 From 1

EXTENSION1

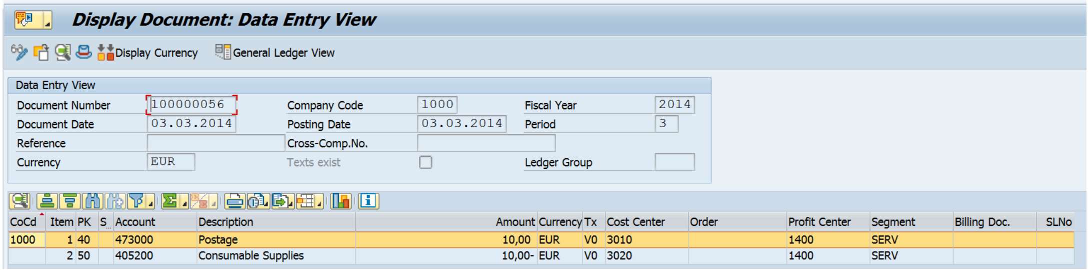
FIELD1	BAPI-PARK
FIELD2	1
FIELD3	
FIELD4	

Output:



```
RFC target sys:

Import parameters                Value

DOCUMENTHEADER
CUSTOMERCPD
CONTRACTHEADER

Export parameters                Value

OBJ_TYPE                         BKPFF
OBJ_KEY                          010000005610002014
OBJ_SYS                          T90CLNT090
```

Document 0100000056 is parked and can be seen at Tcode FBV0



Display Document: Data Entry View

Data Entry View

Document Number	100000056	Company Code	1000	Fiscal Year	2014
Document Date	03.03.2014	Posting Date	03.03.2014	Period	3
Reference		Cross-Comp.No.			
Currency	EUR	Texts exist	□	Ledger Group	

CoCd	Item	PK	S	Account	Description	Amount	Currency	Tx	Cost Center	Order	Profit Center	Segment	Billing Doc.	SLNo
1000	1	40		473000	Postage	10,00	EUR	V0	3010		1400	SERV		
	2	50		405200	Consumable Supplies	10,00-	EUR	V0	3020		1400	SERV		

13. /WINSHTLQ/POST_ACDOC_MULT_CURR

- ✓ **PURPOSE:** Accounting: Posting
- ✓ **DESCRIPTION:** This function is used for Posting Acc Doc in Multiple Currencies

Following is the interface for the function module

✓ **Import Parameters**

1. DOCUMENTHEADER: Enter Header details like company code, posting dates, Fiscal year etc.

2. CUSTOMERCPD: Enter Partner billing doc (load receivable) details like Name, Address, Bank account number etc.

3. CONTRACTHEADER: Enter Additional Contract Accounts Receivable and Payable Header Line details like Doc no., Doc. Type, Reconciliation Key for General Ledger Accounting.

4. **CHECK_POST:** For posting this Flag should be set to 'P' and for checking data, it should be set to 'C'.

✓ **Export Parameters**

1. **OBJ_TYPE:** This parameter will return the Reference Transaction number
2. **OBJ_KEY:** This parameter will return the Reference Key

3. OBJ_SYS: This parameter will return the Logical system of source document

✓ Tables

1. ACCOUNTGL: This Table is used for transferring G/L account line items

The following field must be filled out:

- ✓ **GL_ACCOUNT** G/L account: **Default:** If this method can be used to post G/L account line items across several company codes, the following fields are also contained in the G/L account line items.
- ✓ **COMP_CODE** Company code
- ✓ **PSTNG_DATE** Posting date
- ✓ **DOC_TYPE** Document type
- ✓ **AC_DOC_NO** Document number
- ✓ **FISC_YEAR** Fiscal year
- ✓ **FIS_PERIOD** Fiscal period

If the fields are blank, the information is copied from the document header (parameter Document header).

2. ACCOUNTRECEIVABLE: This table is used for transferring customer line items.

If the CUSTOMER number is not given, only the general ledger is updated. In this case, an entry must be made in the field GL_ACCOUNT (G/L account for general ledger accounting).

The currency fields are not contained in this parameter. They must be transferred separately in the parameter CurrencyAmount for the various currency types and currencies. The system creates a link to the relevant line item via the line item number ITEMNO_ACC, which for this reason must be unique for each line item.

3. ACCOUNTPAYABLE: This table is used for transferring vendor line items.

If the vendor account number VENDOR_NO is not given, only the general ledger is updated. In this case, an entry must be made in the field GL_ACCOUNT (G/L account for general ledger accounting).

The currency fields are not contained in this parameter. They must be transferred separately in the parameter CurrencyAmount for the various currency types and currencies. The system creates a link to the relevant line item via the line item number ITEMNO_ACC, which for this reason must be unique for each line item.

4. ACCOUNTTAX: This table is used for transferring tax lines.

The following fields must be filled:

- ITEMNO_ACC Item number

- TAX_CODE Tax code
- COND_KEY Condition type

If the Jurisdiction Code is active, these fields must also be filled.
The currency fields are not contained in this parameter. They must be transferred in the parameter Currency Amount for the different currency types or currencies. A connection is created to the corresponding document line item via the item number ITEMNO_ACC which must be unique for all line items.

5. **CURRENCYAMOUNT:** This parameter describes the currency fields for a line item. The parameter is a required field for the method POST. If the parameter is not maintained for the method CHECK, only a general account assignment check is carried out. Individual line items or several documents can be checked at once in this way.

   The following fields must be filled out:

   ✓ **CURRENCY** Currency key
   ✓ **AMT_DOCCUR** Currency amount

   Value range

   The field CURR_TYPE (currency type) describes the role played by a currency in the SAP System. The currency type can take on the following values, for example:

   ✓ 00 Transaction currency
   ✓ 10 Company code currency
   ✓ 20 Controlling area currency
   ✓ 30 Group currency
   ✓ 40 Hard currency
   ✓ 50 Index-based currency
   ✓ 60 Global company currency

   If an entry is not made in the field CURR_TYPE, the default value "00" is retained.

   If only one data record is transferred in transaction currency for each line item, the field CURR_TYPE does not need to be filled out. If several data records are transferred for each line item, the currency type must be entered for each record.

6. **CRITERIA:** CO-PA Account Assignment Characteristics

   You can use this parameter to transfer the Profitability Analysis (CO-PA) characteristics to an account assignment.

   The system creates a link to the relevant line item via the line item number ITEMNO_ACC, which for this reason must be unique for each line item.

Note: The contents of the parameter must be sorted before being transferred. If not, errors may occur when assigning error messages to parameter lines.

7. **VALUEFIELD:** CO-PA Account Assignment Value Fields

   This parameter can be used to transfer the value fields for the profitability analysis (CO-PA) for an accounting assignment.

   The system creates a link to the relevant line item via the line item number ITEMNO_ACC, which therefore must be unique for each line item.

   Note: The parameter contents must be sorted when transferred, otherwise errors may occur when assigning error messages to the parameter lines.

8. **EXTENSION1:** Container for 'Customer Exit' Parameter

   This exit can be used to edit the accounting document before it is transferred to the accounting components for the update.

9. **RETURN:** This table contains the return messages from the method call.

Value range

The first message contains the general status of processing:

Type	ID	Number	Message
S	RW	605	Document posted successfully
S	RW	614	Document check without errors
A	RW	609	Error in document

The fields MESSAGE_V1, MESSAGE_V2 and MESSAGE_V3 contain the following information:

✓ Reference transaction (OBJ_TYPE)
✓ Reference key (OBJ_KEY)
✓ Logical system of source document (OBJ_SYS)

Additional messages are transferred with the following information:

✓ Parameter (PARAMETER)
✓ Row (ROW)
✓ Field (FIELD) in which the error occurred.

10. **PAYMENTCARD:** In this parameter data for credit card payment can be transferred.

Link to debitor line:
The item is also transferred to the debitor line (ACCOUNTRECEIVABLE) and must therefore contain the same item number (ITEMNO_ACC).

11. **CONTRACTITEM:** Additional Contract Accounts Receivable and Payable Document Line Item

In this parameter fields are transferred that are concerned with an accounting document not including the contract accounts receivable and payable (FI-CA) at the item level.

Link to debitor line:
The item is also transferred to the debitor line (ACCOUNTRECEIVABLE) and must therefore contain the same item number (ITEMNO_ACC).

12. **EXTENSION2:** Reference Structure for BAPI Parameters EXTENSIONIN/EXTENSIONOUT

This enhancement structure can be used to transfer additional fields.

There are four fields available for each data record (VALUEPART1 - 4), which each contain 240 characters. A defined ABAP dictionary structure can be transferred in the field STRUCTURE, based on which the field contents are read.

The accounting document can be supplemented and changed in an implemented BAdI (ACC_DOCUMENT) before the active accounting component is called.

Default: The class CL_EXM_IM_ACC_DOCUMENT is available as example implementation. Here, for each data record of the parameter that is transferred the ABAP dictionary structure is checked and the content of the fields VALUEPART1 to -4 is set in the structure. If the structure that is transferred contains a field POSNR, the associated lines of the accounting document are updated using the contents of the enhancement.

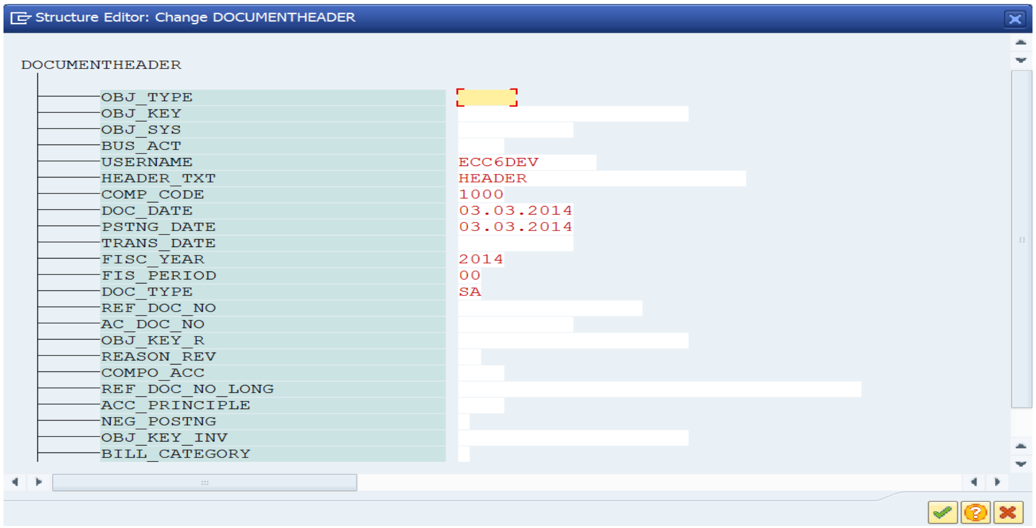
13. **REALESTATE:** Real Estate Account Assignment Data

This parameter can be used to transfer account assignments for Real Estate Management.
The current line represents an addition to the accounting document line items (ACCOUNTGL, ACCOUNTRECEIVABLE, ACCOUNTPAYABLE). The line in parameter REALESTATE must be connected to the document line item (ITEMNO_ACC) using the line number.

14. **ACCOUNTWT:** This table is used for Withholding Tax Information like ITEMNO_ACC, Withholding Tax Base Amount (Local Currency), Withholding Tax Base Amount (Document Currency), Withholding tax code etc.

Example:

Enter Input as below:



Structure Editor: Change DOCUMENTHEADER

DOCUMENTHEADER

OBJ_TYPE	
OBJ_KEY	
OBJ_SYS	
BUS_ACT	
USERNAME	ECC6DEV
HEADER_TXT	HEADER
COMP_CODE	1000
DOC_DATE	03.03.2014
PSTNG_DATE	03.03.2014
TRANS_DATE	
FISC_YEAR	2014
FIS_PERIOD	00
DOC_TYPE	SA
REF_DOC_NO	
AC_DOC_NO	
OBJ_KEY_R	
REASON_REV	
COMPO_ACC	
REF_DOC_NO_LONG	
ACC_PRINCIPLE	
NEG_POSTNG	
OBJ_KEY_INV	
BILL_CATEGORY	

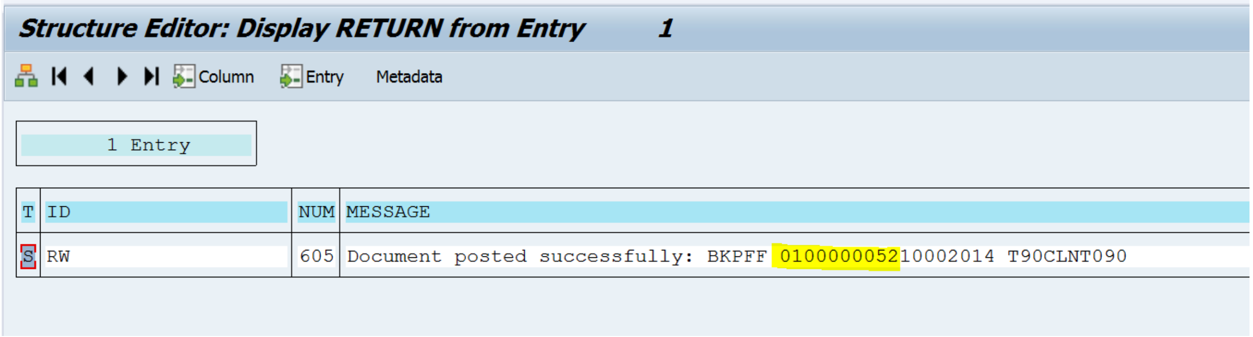
- **AccountGl table:**

  ITEMNO_ACC = ‘0000000001’.
  GL_ACCOUNT = ‘473000 ‘.
  ITEM_TEXT = ‘LINE 1’.
  COSTCENTER = ‘3010’.
  TRANS_CURR_TYPE = ‘00’.
  TRANS_CURRENCY = ‘EUR’.
  TRANS_AMT_DOCCUR = ‘100’.

  ITEMNO_ACC = ‘0000000002’.
  GL_ACCOUNT = ‘405200 ‘.
  ITEM_TEXT = ‘LINE 2’.
  COSTCENTER = ‘3020’.
  TRANS_CURR_TYPE = ‘00’.
  TRANS_CURRENCY = ‘EUR’.
  TRANS_AMT_DOCCUR = ‘-100’.

- CHECK_POST = ‘P’. (If document needs to be posted, else ‘C’ to check the data before posting document)

Output: Return Table shows Document Number (Concatenated with Company code and fiscal year) posted.

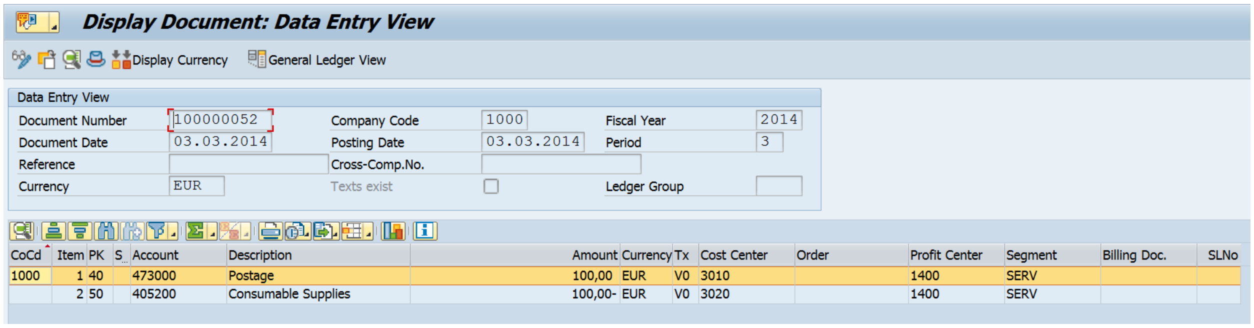


Structure Editor: Display RETURN from Entry 1

1 Entry

T	ID	NUM	MESSAGE
S	RW	605	Document posted successfully: BKPFF 010000005210002014 T90CLNT090

Posted Document can be seen on Tcode: FB03



Display Document: Data Entry View

Data Entry View

Document Number	100000052	Company Code	1000	Fiscal Year	2014
Document Date	03.03.2014	Posting Date	03.03.2014	Period	3
Reference		Cross-Comp.No.			
Currency	EUR	Texts exist	□	Ledger Group	

CoCd	Item	PK	S	Account	Description	Amount	Currency	Tx	Cost Center	Order	Profit Center	Segment	Billing Doc.	SLNo
1000	1	40		473000	Postage	100,00	EUR	V0	3010		1400	SERV		
	2	50		405200	Consumable Supplies	100,00-	EUR	V0	3020		1400	SERV		