

# Creating Extension IDOC for Outbound Delivery



## Applies to:

Developing Extension IDOC and testing it, in versions SAP 4.6C and above. For more information click on the following link [Extension of IDoc Types and their Processing](#).

## Summary

This document provides the step by step procedure for, creating an extension IDOC, populating the custom segment and triggers the created extension IDOC from transaction.

**Author:** Manoharan Murugesan

**Company:** Intelligroup Asia Pvt Limited

**Created on:** 27 October 2009

## Author Bio



Manoharan Murugesan works with Intelligroup Asia Pvt Ltd since 2008. He has good knowledge in SAP ABAP.

## Table of Contents

Introduction .....	3
Scenario.....	3
Tcode Used .....	3
Steps to Be Followed.....	3
1) Creating a Custom Segment.....	3
2) Create an Extension IDOC .....	5
3) Create a Logical Message Type .....	9
4) Assign Logical Message Type to Extension IDOC Type .....	10
5) Create a Partner Profile and Assign Message Type .....	11
6) Finding the Customer Exit and Write the Logic to Populate the Custom Segment.....	15
7) Configuring the Output Types in VL02N Transaction .....	18
8) Checking the Created IDOC .....	21
9) Standalone Testing.....	23
Related Content.....	24
Disclaimer and Liability Notice.....	25

## Introduction

The extension IDOC concept comes into action when the standard IDOCs provided by SAP does not meet customers requirement. IDOC is the acronym of Intermediate Document; IDOCs are nothing but a Data Format provided by SAP to transfer data in and out of an SAP system. Data are grouped under this (IDOC) data format for each application in order to maintain consistency while transferring data from one system to another system. Hence each application like order, delivery, invoice etc will have a standard IDOC available along with standard processing programs for the same. When standard IDOC does not meet customers requirement, custom extension IDOC are to be developed to meet the requirement and the corresponding processing program will be enhanced using existing customer exits.

## Scenario

In this case, we will discuss the extension of basic IDOC type 'DELVRY03', this IDOC type is used to group delivery detail. The mentioned IDOC type does not have fields VBUK-KOSTK (Overall picking) and VBUK-KOQUK (Status of pick confirmation in its standard format. But customers business needs demand these fields to be added to delivery details. Hence the basic IDOC type 'DELVRY03' will be extended as 'ZDELVRY03' with the two new fields and relevant processing program will be identified and processing logic for the new fields will be included in the relevant customer exits.

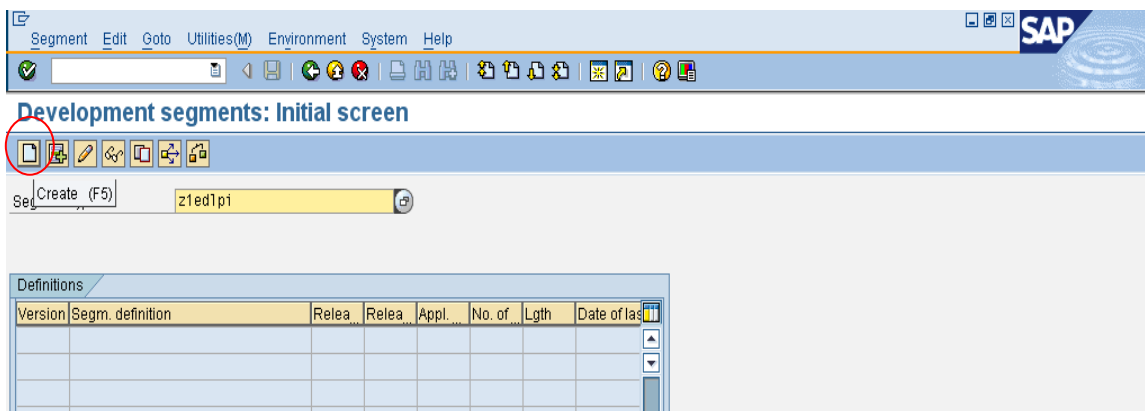
## Tcode Used

- WE31 – Development segments – initial screen
- WE30 - Develop IDOC type – initial screen
- WE81 - EDI:Logical message type
- WE82 - Output type and Assignment to IDOC type
- WE20 - Partner Profile.
- NACE - Conditions for output controls
- WE19 - Test tool for IDOC processing

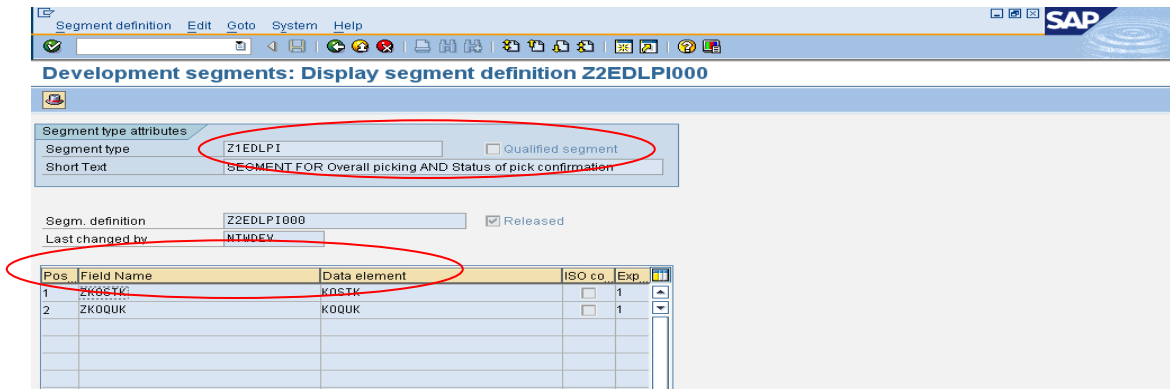
## Steps to Be Followed

### 1) Creating a Custom Segment

It begins with creating a custom segment; custom segment creation is developed in transaction WE31. Custom segment names begin with 'Z1' (Naming Conventions), Now go to T-code: WE31 and type your custom segment name as Z1EDLPI and click on the create button as shown below.

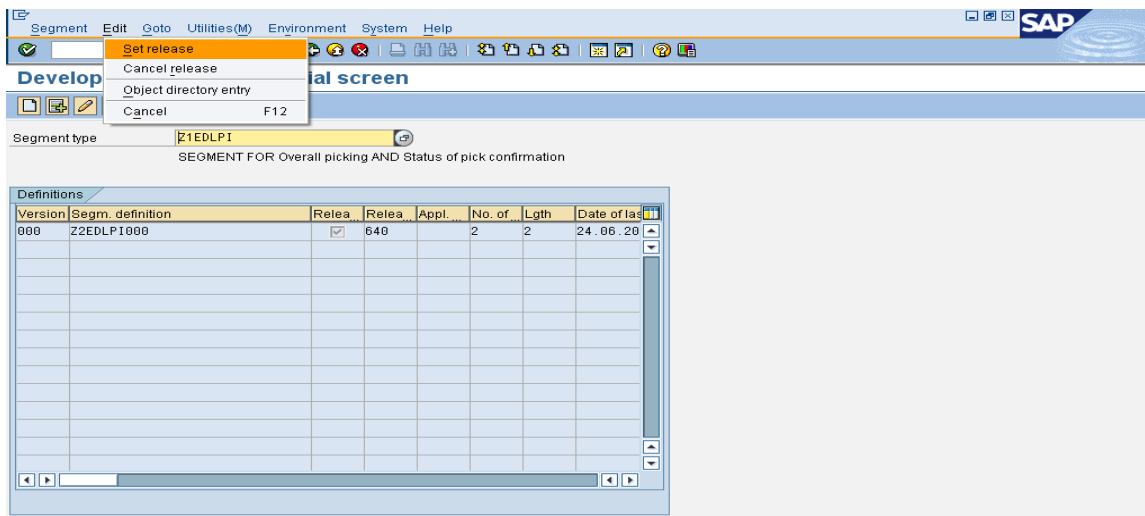


When the create button is clicked, the below shown screen will be displayed. Enter a short description for the custom segment and add the required field name and the corresponding data element. Once the required fields are added, click on the save button.



After clicking on save button, click on back button, you will see the below show screen. Now click on the segment definition you have created and do Set Release from EDIT menu.

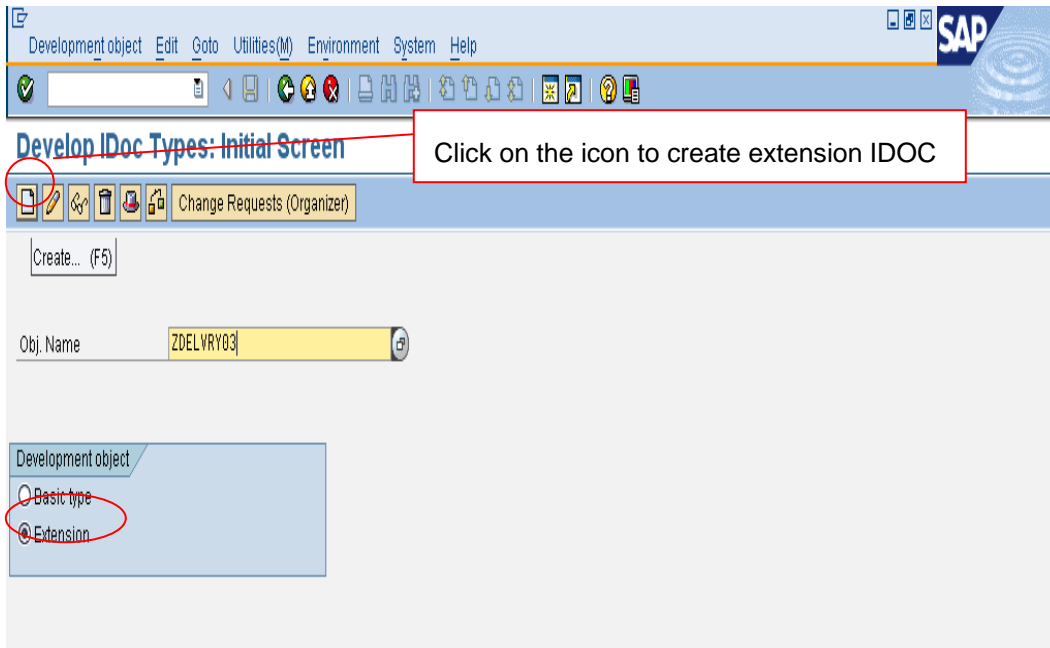
**Note:** Without releasing your segment definition you cannot use the segment in any IDOC type.



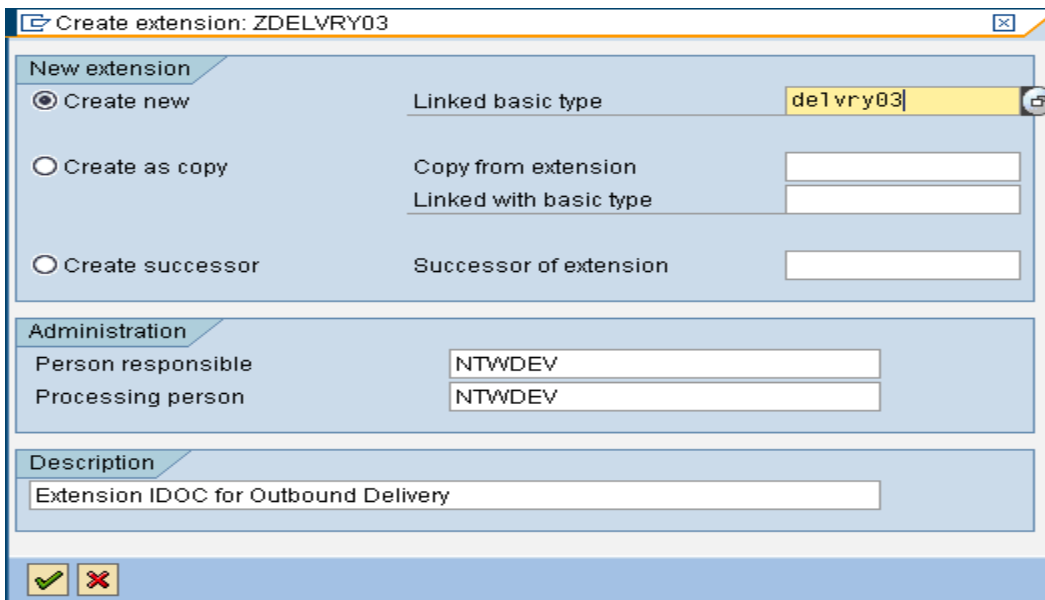
With this, custom segment creation is completed.

## 2) Create an Extension IDOC

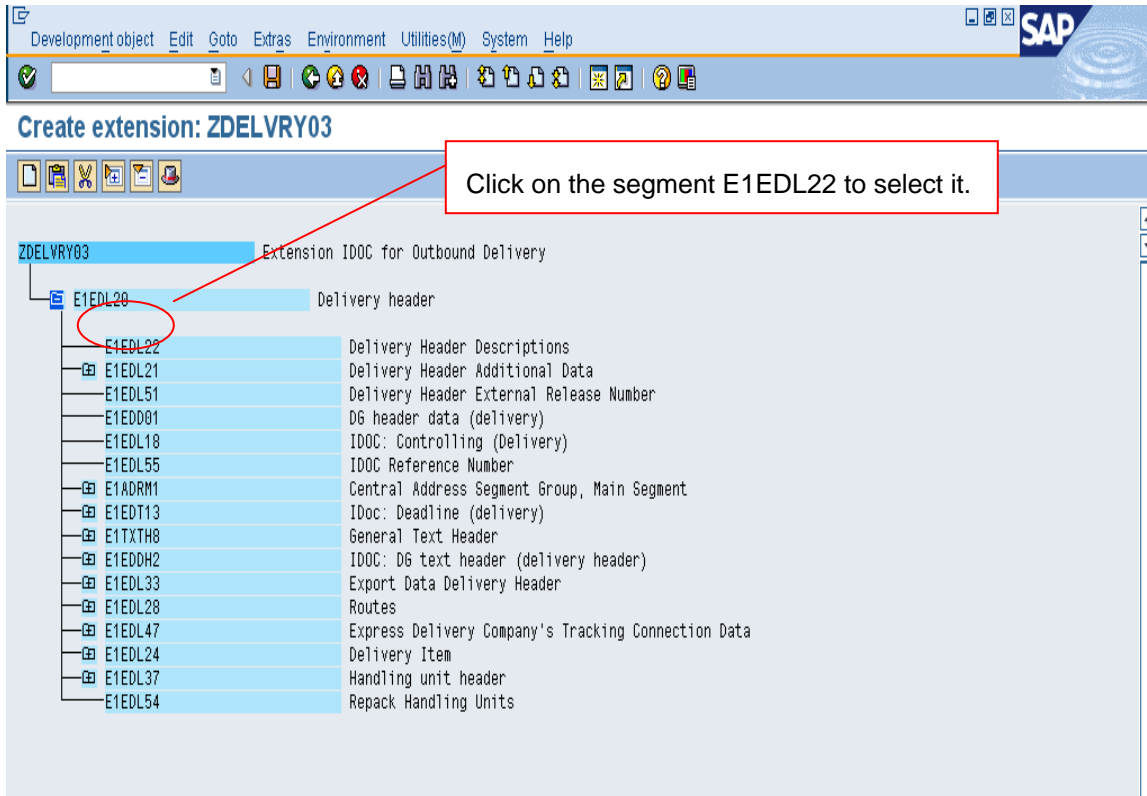
With the above created custom segment, go to transaction WE30. Enter a name 'ZDELVR03' for the extension IDOC and select on the Extension radio button and click create button.



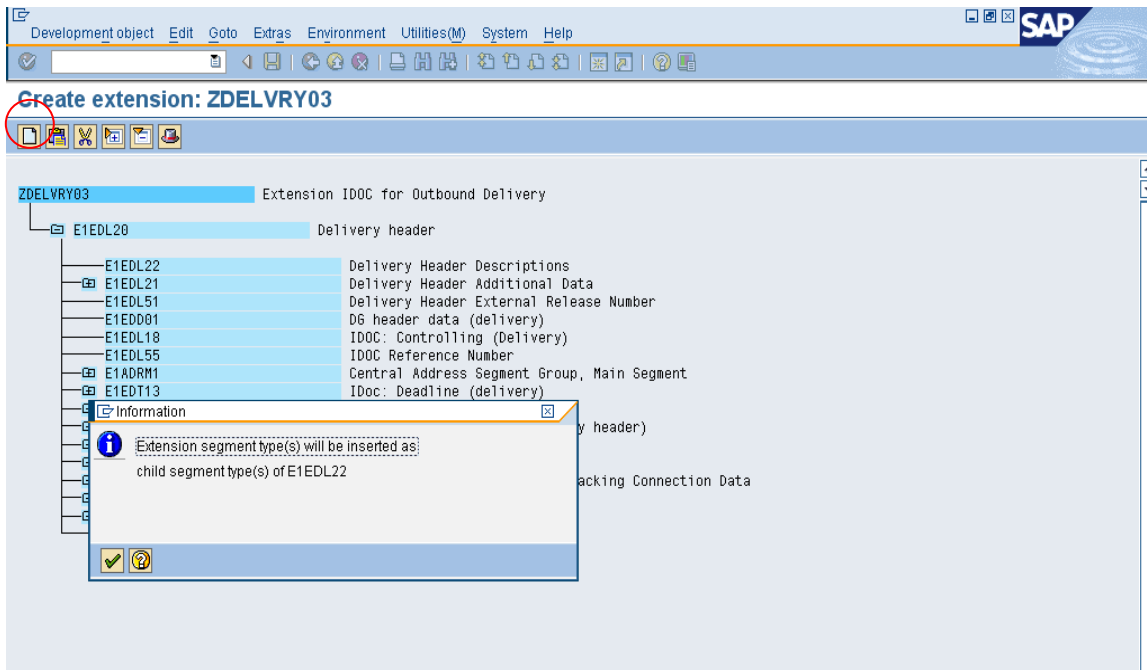
The following popup screen will be displayed. Click on the 'Create New' radio button to create an extension with the help of a Basic IDOC type. Enter the standard basic type name 'delvry03' in the 'Linked Basic type' field and enter a short description and click continue.



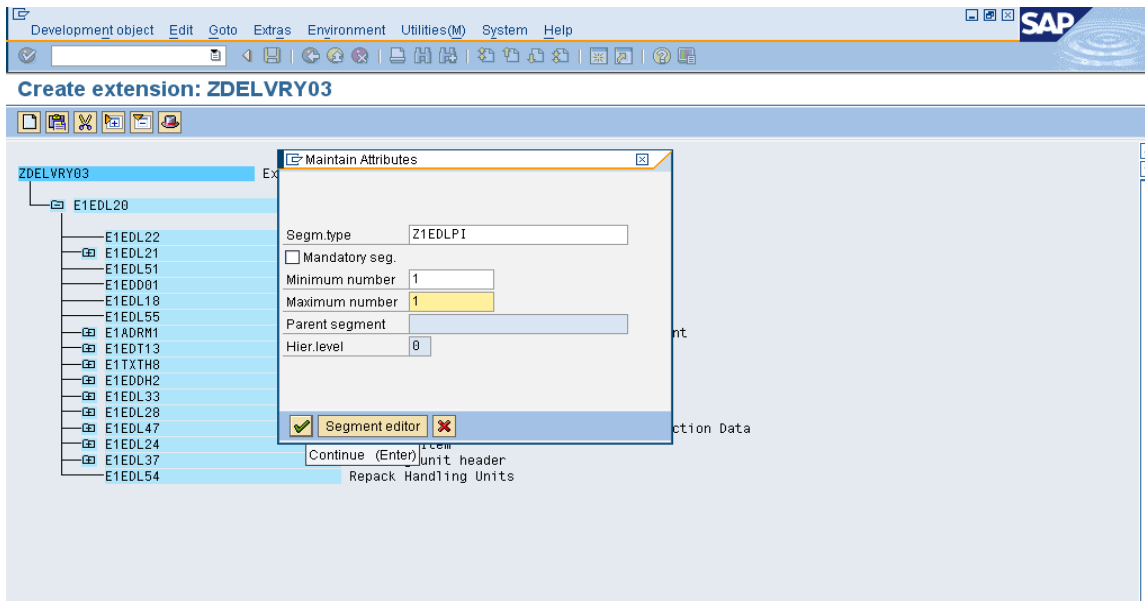
The following screen will be displayed with all the standard segment of basic type 'delvry03'. Now we need to add our custom segment to the required segment, in our example we will take segment E1EDL22. We will have to add the custom segment Z1EDLPI to segment E1EDL22.



Select the segment E1EDL22 and click on the create button, an information message will pop up and say 'Extension segment type(s) will be in child segment type(s) of E1EDL22' as shown below. Click on continue.

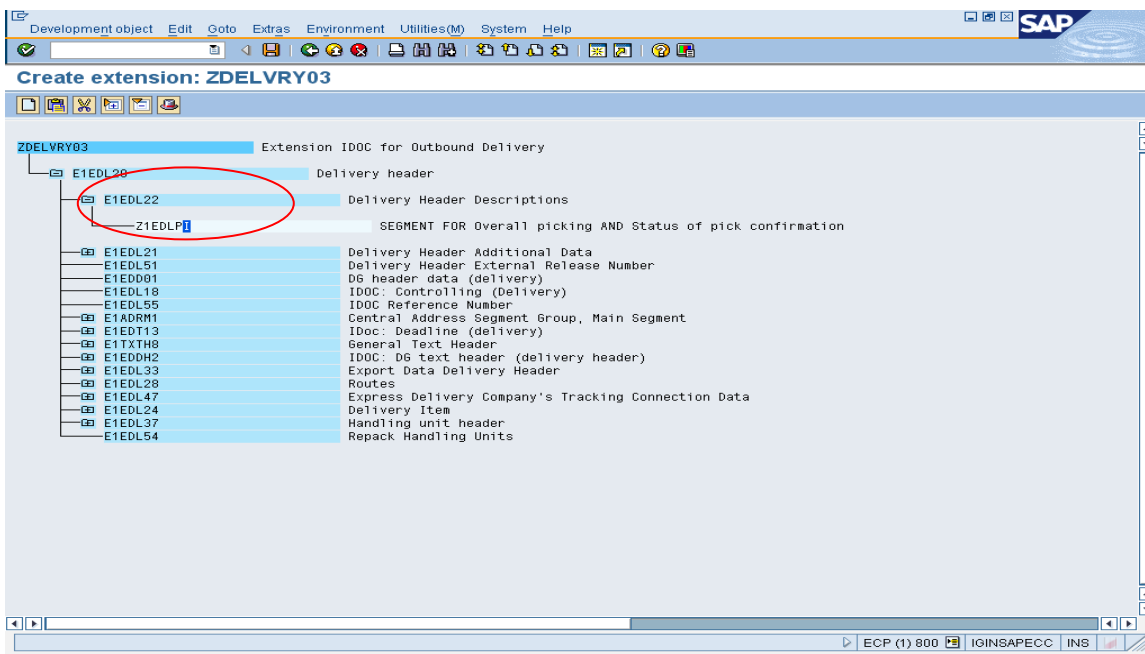


Another pop up will be prompted for entering the custom segment name and attributes. Enter the name of the segment Z1EDLPI, minimum and maximum occurs as shown below.

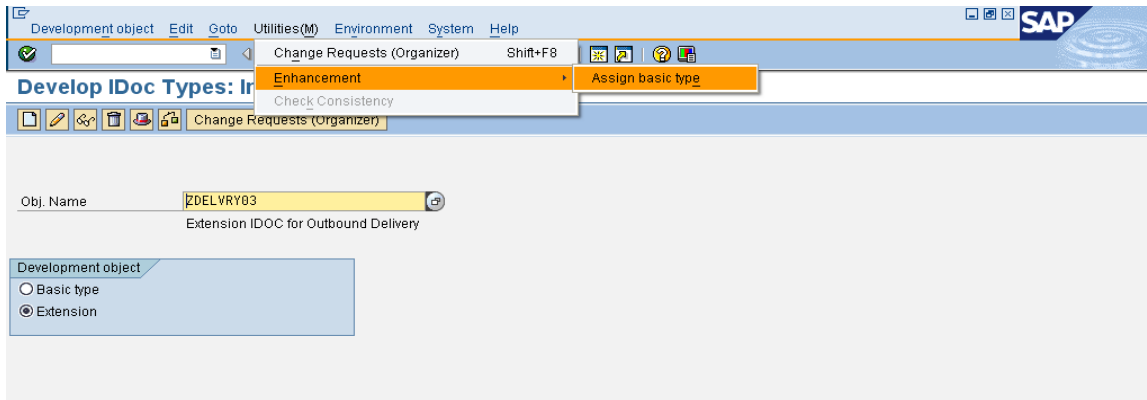


**Note:** Attribute entries may vary depending on the requirement, the above shown attribute entries are particular to this example only. When a custom segment is assigned at item level the minimum and maximum number may vary as per the requirement.

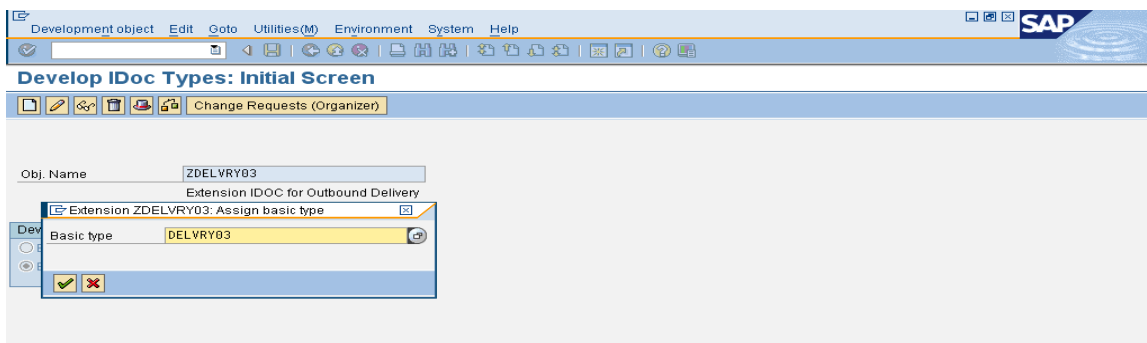
Newly added segment is seen as below.



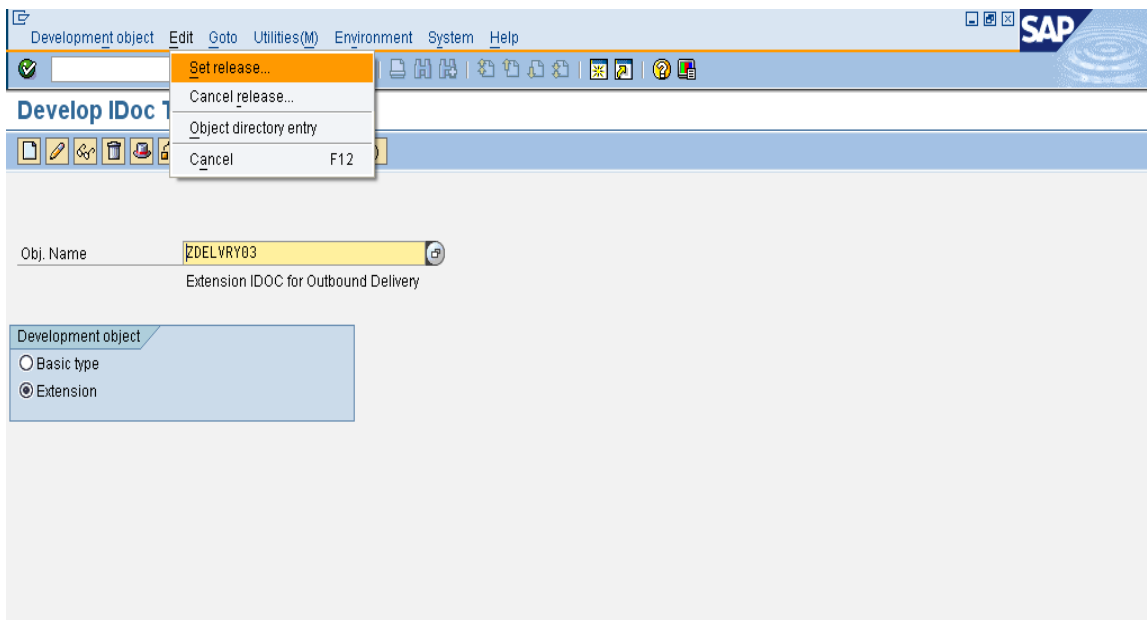
Save the extension IDOC and click on the back button and go to Utilities -> Enhancement -> Assign basic type as shown below.



The below pop up will be prompted with the basic type name, which we have used to create our extension type. In our case it is 'delvry03'. Click on the continue button.

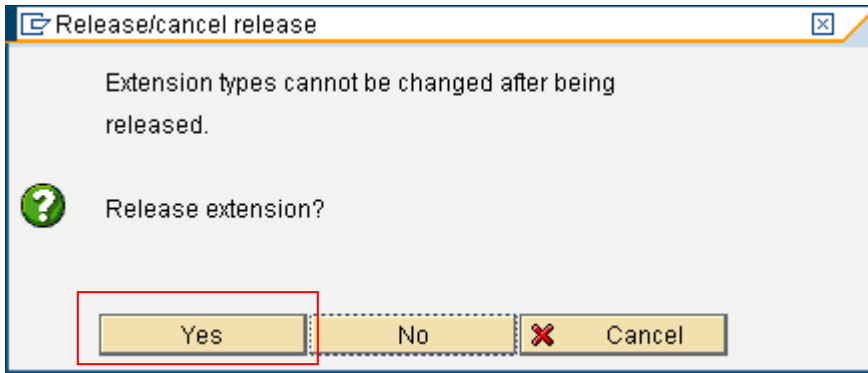


Now click on the Set Release menu item in EDIT menu to release the extension IDOC.





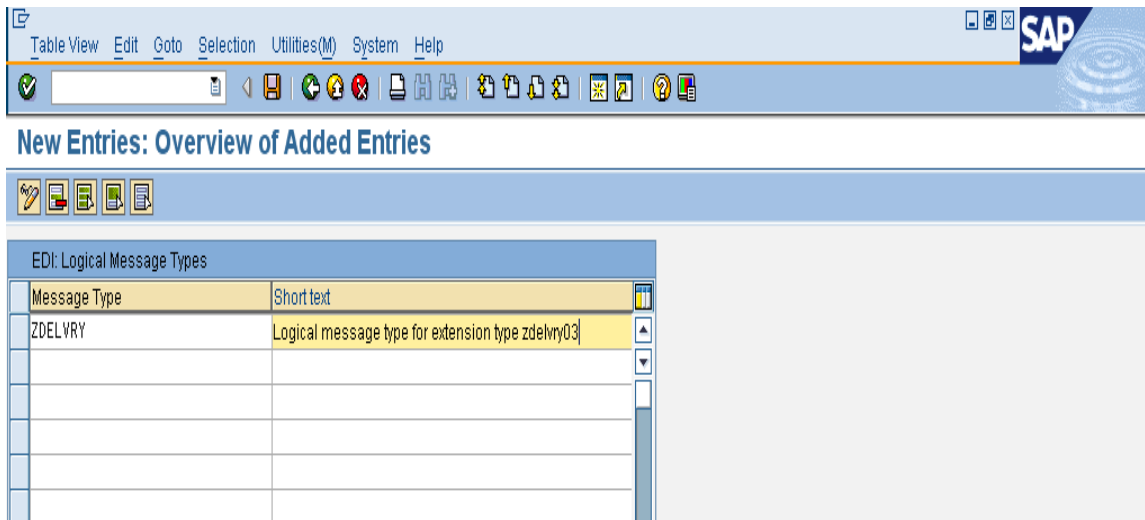
A pop up will be prompted as shown below, click on YES to release the Extension IDOC.



We have successfully created our extension IDOC 'ZDELVR03'.

### 3) Create a Logical Message Type

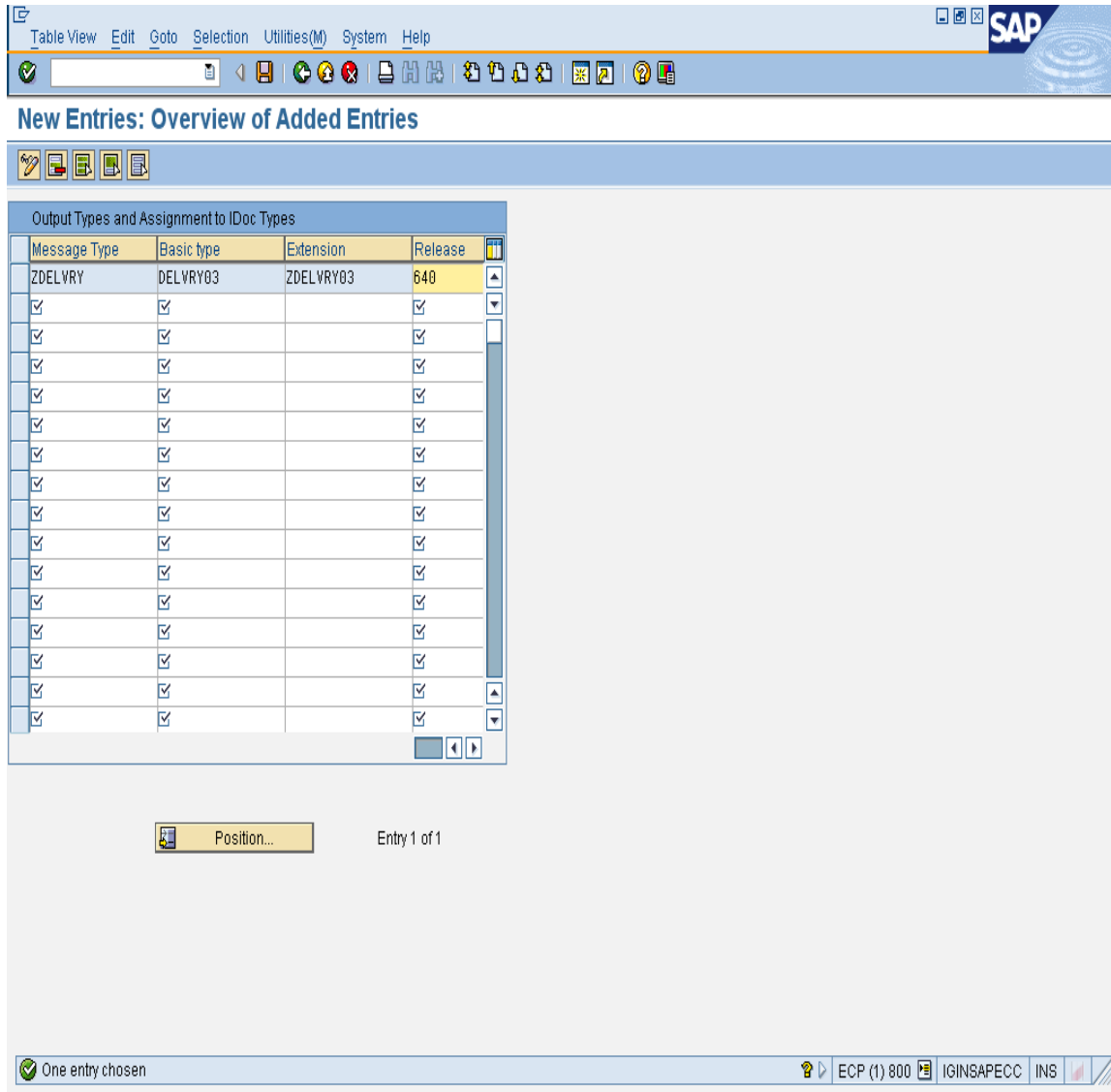
Go to transaction WE81. By default the screen will be in display mode, go to change mode to enter a new entry. Give some name to the logical message type and a short description. Save the entry.



#### 4) Assign Logical Message Type to Extension IDOC Type

Assign the created extension IDOC type with the above created logical message type. This is carried out in transaction WE82.

Here you will assign the message type with the extension IDOC type along with the basic type and release version combination as shown below.



The screenshot displays the SAP WE82 transaction interface. The title bar reads "New Entries: Overview of Added Entries". Below the title bar is a menu bar with options: "Table View", "Edit", "Goto", "Selection", "Utilities(M)", "System", and "Help". A toolbar with various icons is visible below the menu bar. The main area contains a table titled "Output Types and Assignment to IDOC Types".

Message Type	Basic type	Extension	Release
ZDELVRY	DELVRY03	ZDELVRY03	640
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		<input checked="" type="checkbox"/>
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		<input checked="" type="checkbox"/>
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		<input checked="" type="checkbox"/>
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		<input checked="" type="checkbox"/>
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		<input checked="" type="checkbox"/>
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		<input checked="" type="checkbox"/>
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		<input checked="" type="checkbox"/>
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		<input checked="" type="checkbox"/>
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		<input checked="" type="checkbox"/>
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		<input checked="" type="checkbox"/>
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		<input checked="" type="checkbox"/>
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		<input checked="" type="checkbox"/>
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		<input checked="" type="checkbox"/>
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		<input checked="" type="checkbox"/>
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		<input checked="" type="checkbox"/>
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		<input checked="" type="checkbox"/>

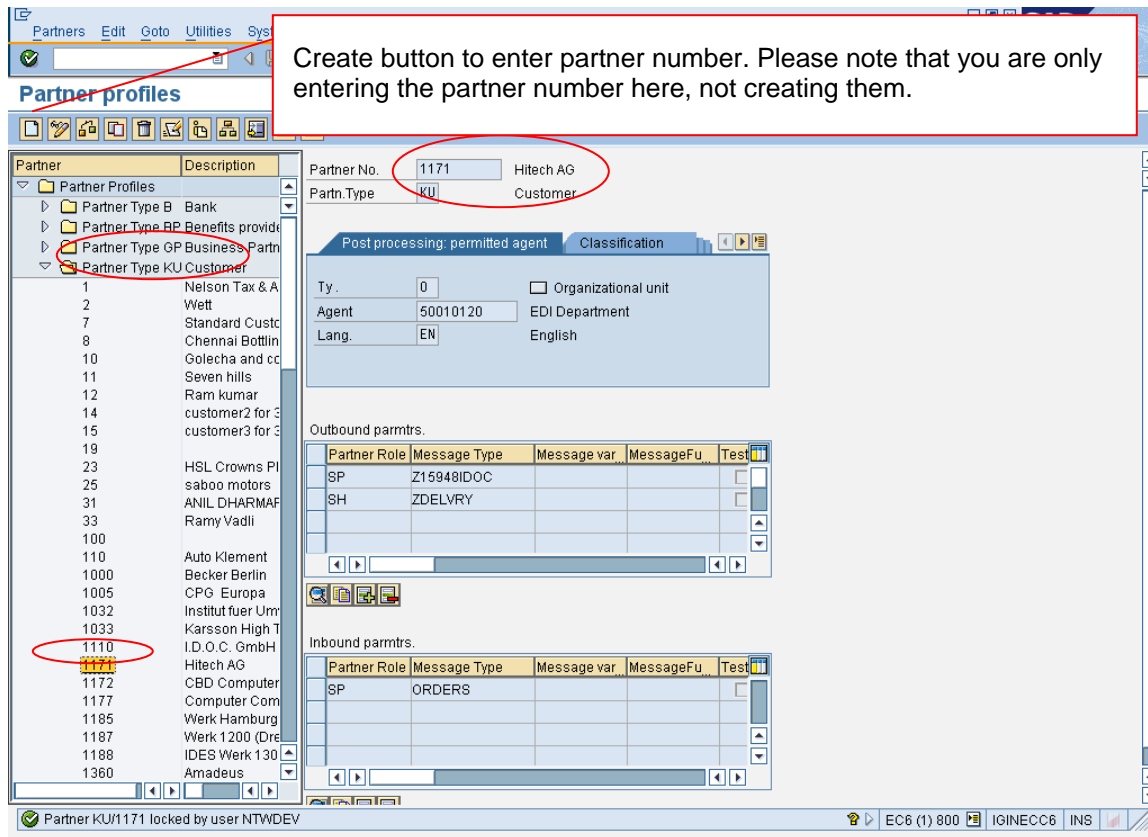
Below the table, there is a "Position..." button and the text "Entry 1 of 1". At the bottom of the window, a status bar shows "One entry chosen" and system information: "ECP (1) 800 IGINSAPECC INS".

Save the entry. Now the message type 'ZDELVRY' will have the same structure as extension IDOC 'ZDELVRY03' in runtime.

**Note:** Message type carries the actual data at the runtime. Message type uses the structure of the assigned IDOC type (In our case the extension IDOC type 'ZDELVRY03').

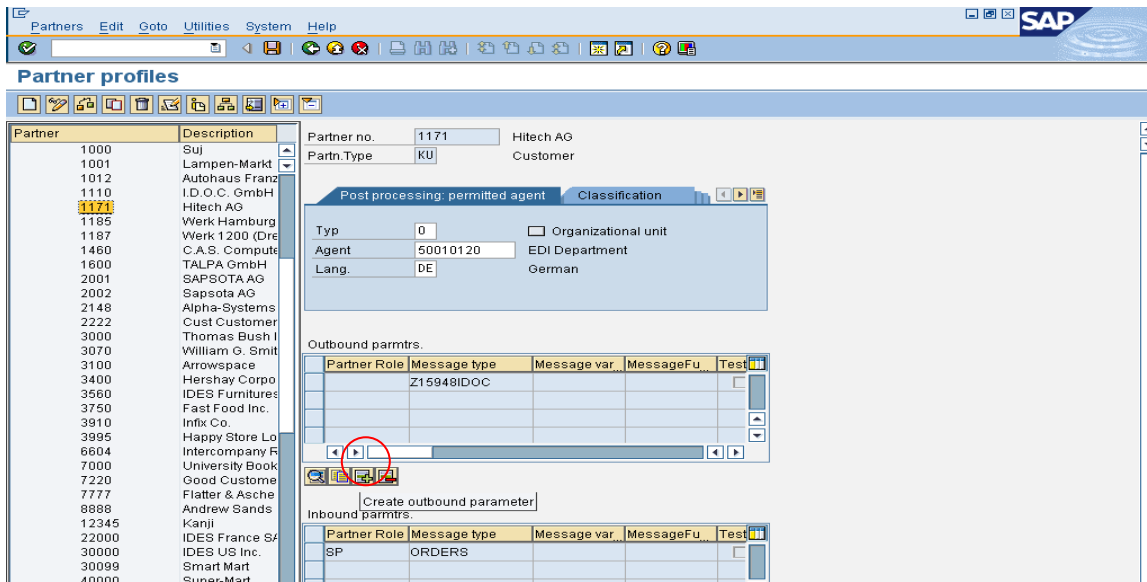
### 5) Create a Partner Profile and Assign Message Type

Create a partner profile and assign our message type to it. Also provide details like output types, process code etc in message control tab. To create a partner profiles go to transaction WE20.

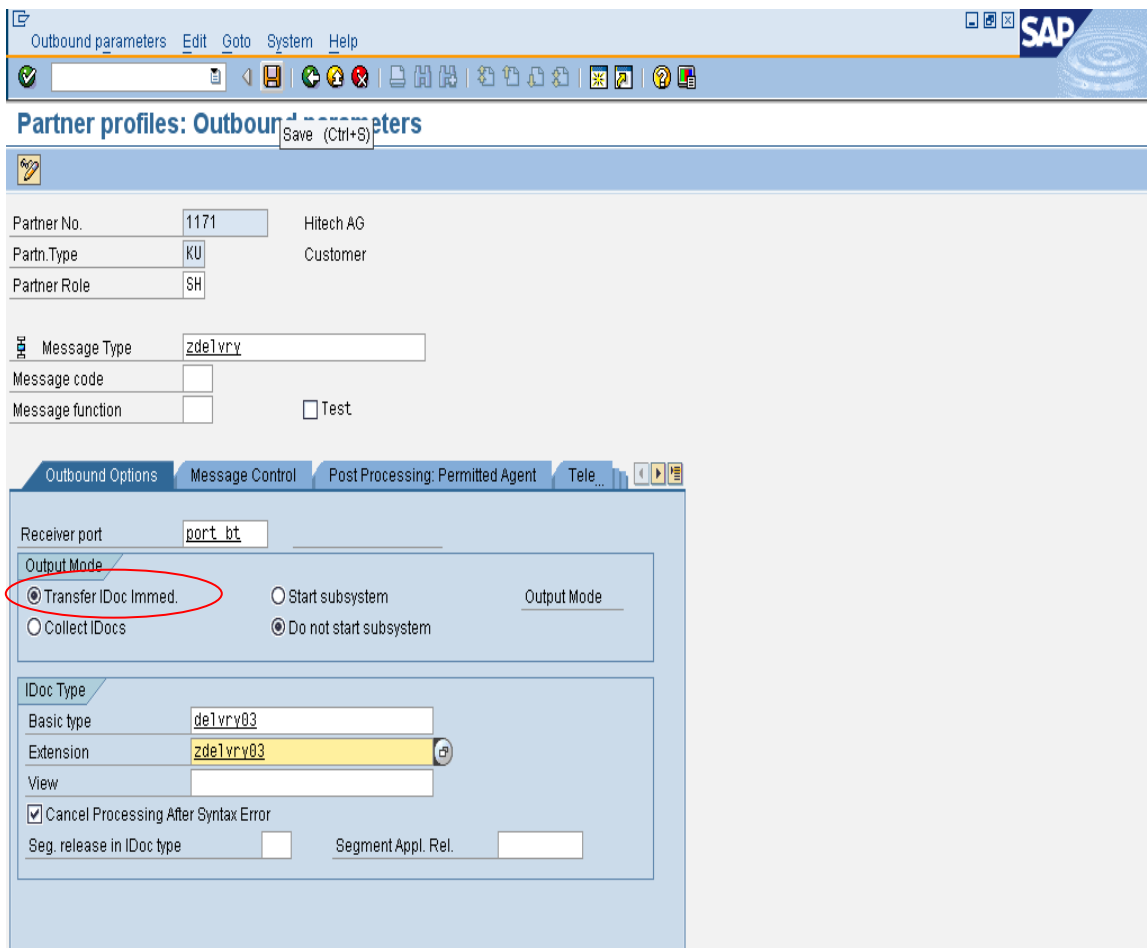


**Note:** The partner number in partner profile and ship to party number in delivery (VL02N) should be same. In the below shown example we are using Delivery number '80016347' and hence the ship to party '1171' will be our partner number. In WE20 transaction, expand the partner type 'Partner type KU Customer' and find the partner number '1171'. If it doesn't exist then click on the create button and type the partner number '1171' and save the entry.

Now click on the Create Outbound Parameter button (An icon with green plus symbol).



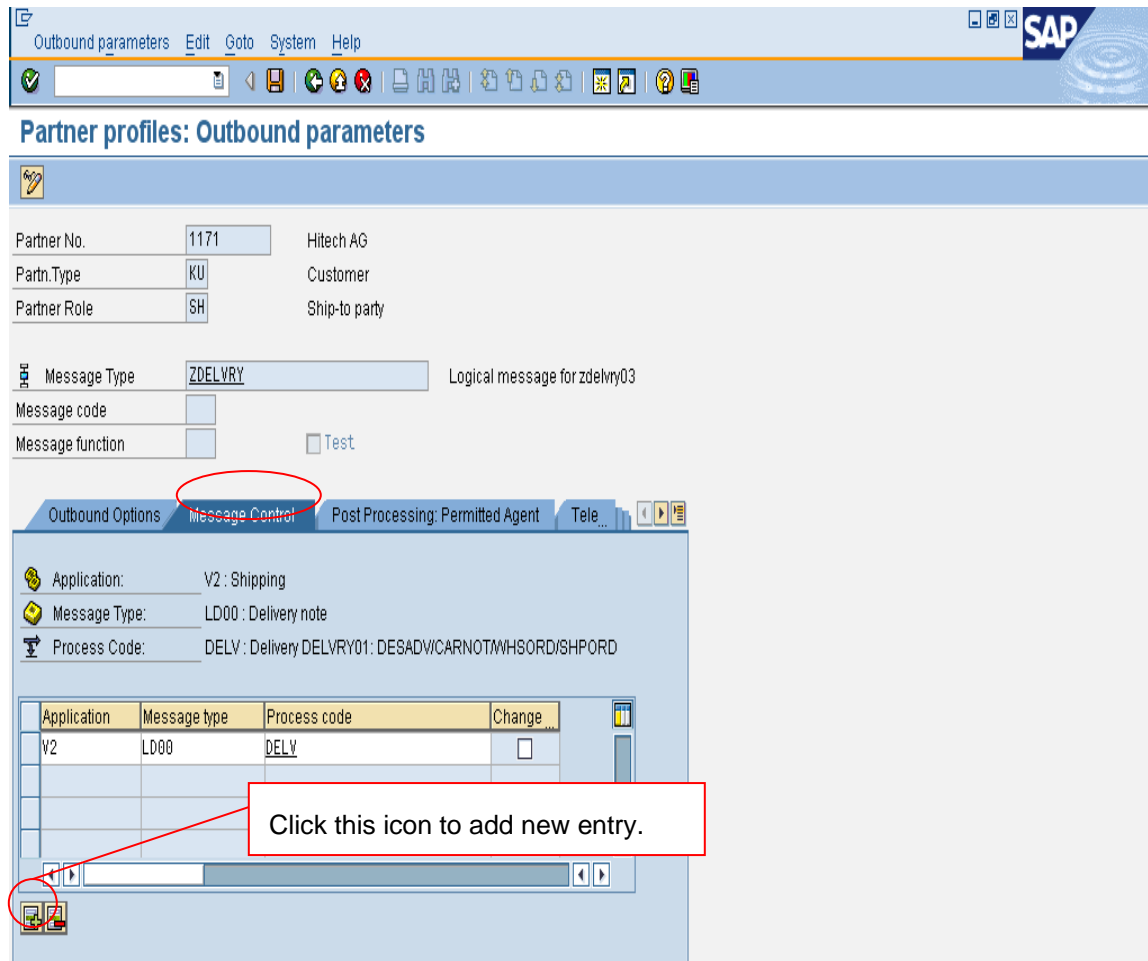
The following screen will be displayed. Enter the partner role (optional), message type, receiver port, basic type and extension type names respectively. Also select the 'Transfer IDOC immed.' Radio button. And save the entries.



Transfer IDOC immediately option is selected to send the IDOC immediately when a delivery document is saved (i.e. each time the delivery document is saved successfully the IDOC will get created and transferred to the port).

**Note:** Collect IDOCS option will collect the IDOCS and process it when scheduled (i.e. each time the delivery document gets saved, the IDOC gets created but it will get collected and sent to port only on a scheduled basis.)

Now click on the message control tab as shown below and click on the insert row icon (green plus symbol) to enter the necessary details.



Partner profiles: Outbound parameters

Partner No. 1171 Hitech AG  
 Partn. Type KU Customer  
 Partner Role SH Ship-to party

Message Type ZDELVRY Logical message for zdelvry03  
 Message code  
 Message function  Test

Outbound Options **Message Control** Post Processing: Permitted Agent Tele...

Application: V2 : Shipping  
 Message Type: LD00 : Delivery note  
 Process Code: DELV : Delivery DELVRY01: DESADV/CARNOT/WH5ORD/SHPOED

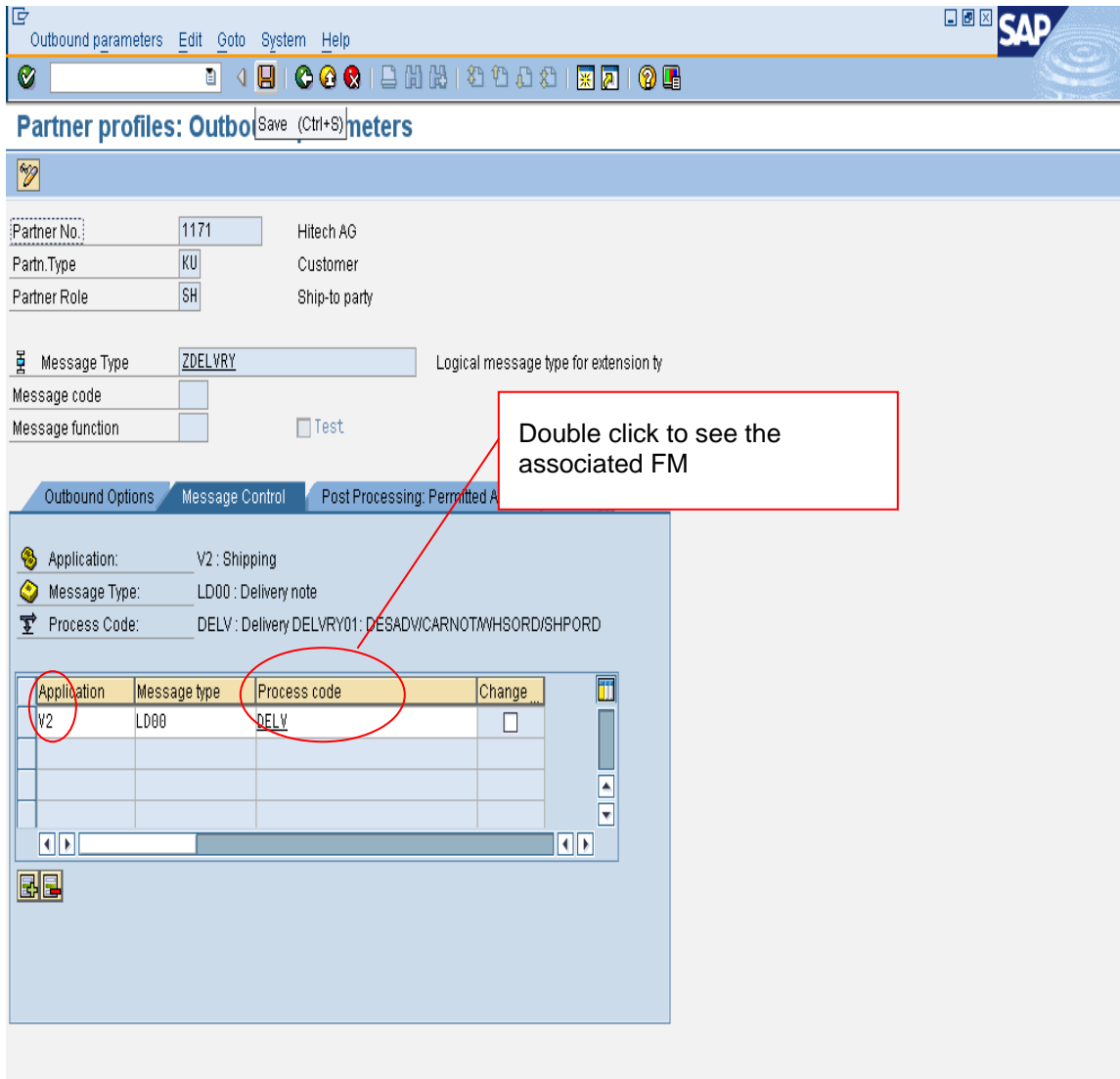
Application	Message type	Process code	Change
V2	LD00	DELV	<input type="checkbox"/>

Click this icon to add new entry.

Use the F4 help to select the appropriate application, in our case it is 'V2' (For Delivery). The next entry is choosing the message type (note that this message type is different from the one which we have created earlier, this message type is used to set the output types in VL02N transaction). In our case the message type is 'LD00'.

**Note:** Appropriate message types are selected with the help of NACE transaction. In NACE transaction, select the appropriate Application (V2) and click the output type PUSH button in the application tool bar. List of output types will be displayed, see which output type is configured for your required output medium (ALE, EDI, PTINTER, FAX etc.) you can use only those output types as message types in partner profile.

Next entry is to choose the process code. For each applications (V1 for sales order, V2 for delivery, V3 for invoice) there are set of process codes maintained in transaction WE41 (For outbound process code). Go to WE41 transaction and select the appropriate process code. In our case it is 'DELV'.

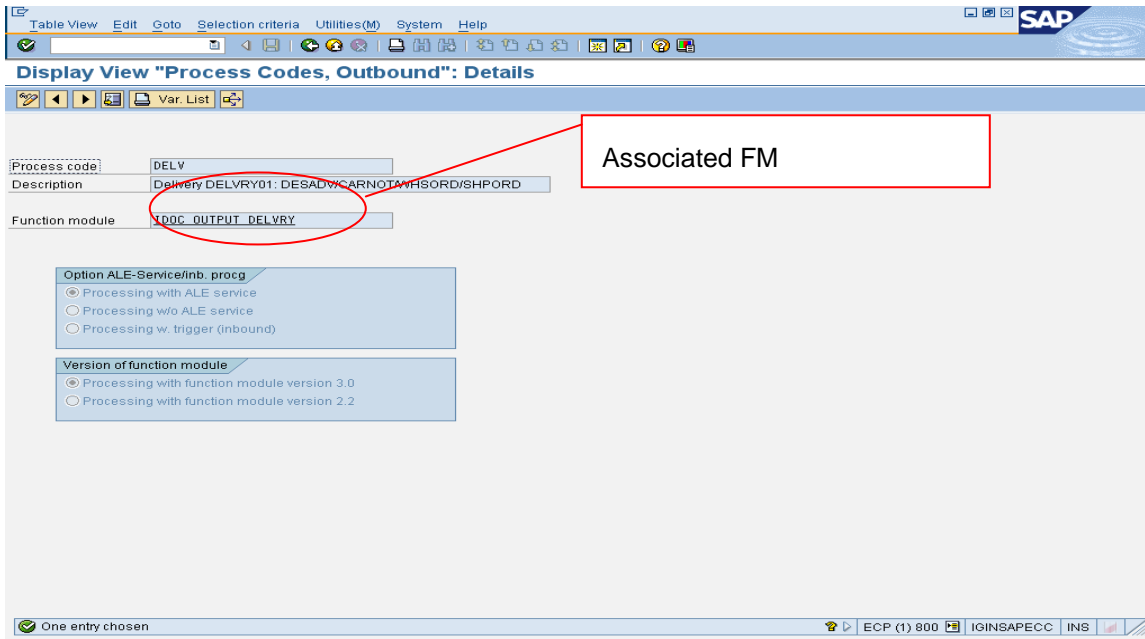


**Note:** The process codes are assigned to function modules which has the code to process (Populate the IDOC segments, save the segment data in EDI specific database table etc) the IDOC segments at runtime.

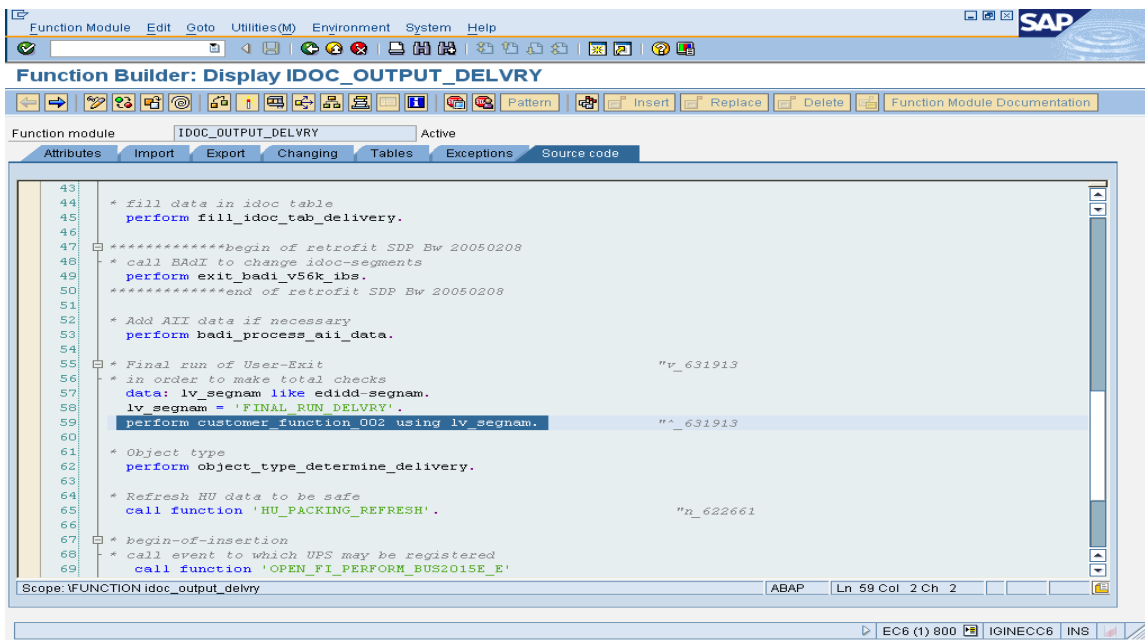
Click on save button. This ends your partner profile creation.

## 6) Finding the Customer Exit and Write the Logic to Populate the Custom Segment

Time for a little coding, as stated earlier the process codes are associated with function modules to process our extension IDOC. To see the function module double click on the process code 'DELV'. It will take you to the following screen.



Next step is to find the customer exit in the function module to populate the custom segment in the extension IDOC. Check for the PERFORM that fill the Data records internal table and proceed to find the relevant customer exit or directly you can search customer exit for Data record. Usually the customer exits for Data records will be "customer-function '002'".



**Note:** FM: IDOC\_OUTPUT\_DELVRY has all the standard logic to populate the standard segments of the extension IDOC 'ZDELVRY03'. We use the customer exit to imply our logic on custom segments and to populate custom segments. In our case we are using it for latter case.

ABAP Editor: Display Include LV56KF01

```

9  form customer_function_002
10     using value(segment_name) like edidd-segnam.
11
12     call customer-function '002'
13     exporting
14         control_record_out = control_record_out
15         message_type       = control_record_in-mestyp
16         segment_name       = segment_name
17         data                = db_data
18         tab_idoc_reduction = tab_idoc_reduction
19     tables
20         idoc_data          = int_edidd
21     exceptions
22         error_message_received = 1
23         data_not_relevant_for_sending = 2.
24     case sy-subrc.
25     when 1.
26         message id      sy-msgid
27         type            sy-msgty
28         number          sy-msgno
29         with            sy-msgv1 sy-msgv2 sy-msgv3 sy-msgv4
30         raising         error_message_received.
31     when 2.
32         message id      sy-msgid
33         type            sy-msgty
34         number          sy-msgno
35         with            sy-msgv1 sy-msgv2 sy-msgv3 sy-msgv4
36         raising         data_not_relevant_for_sending.
37     endcase.
38
39

```

Scope: \FORM customer\_function\_002

ABAP Ln 20 Col 13 Ch 13

EC6 (1) 800 IGINECC6 INS

This is the internal table that need to be populated to add the new segment 'Z1EDLPI' and the corresponding fields.

Function Builder: Display EXIT\_SAPLV56K\_002

```

1  function exit_saplv56k_002.
2  ****
3  ****"Lokale Schnittstelle:
4  ****
5  ****      IMPORTING
6  ****          VALUE (CONTROL_RECORD_OUT) LIKE EDIDC STRUCTURE EDIDC
7  ****          VALUE (MESSAGE_TYPE) LIKE EDMMSG-MSGTYP
8  ****          VALUE (SEGMENT_NAME) LIKE EDIDD-SEGNAM
9  ****          REFERENCE (DATA) TYPE TR_IDOC_SHEMNT_INDATA
10 ****          VALUE (TAB_IDOC_REDUCTION) TYPE TR_TAB_IDOC_REDUCTION
11 ****
12 ****      TABLES
13 ****          IDOC_DATA STRUCTURE EDIDD
14 ****
15 ****      EXCEPTIONS
16 ****          ERROR_MESSAGE_RECEIVED
17 ****          DATA_NOT_RELEVANT_FOR_SENDING
18 ****
19 ****
20 ****
21 ****
22 ****

```

Scope: \FUNCTION exit\_saplv56k\_002

ABAP Ln 11 Col 17 Ch 17

Double click on include name, it is this include which has the custom code for the Data Records.

The below code has the logic for adding the custom segment 'Z1EDLPI' as child of 'E1EDL22' in the data record internal table 'IDOC\_DATA' and what data needs to be populated in the fields of the custom segment.

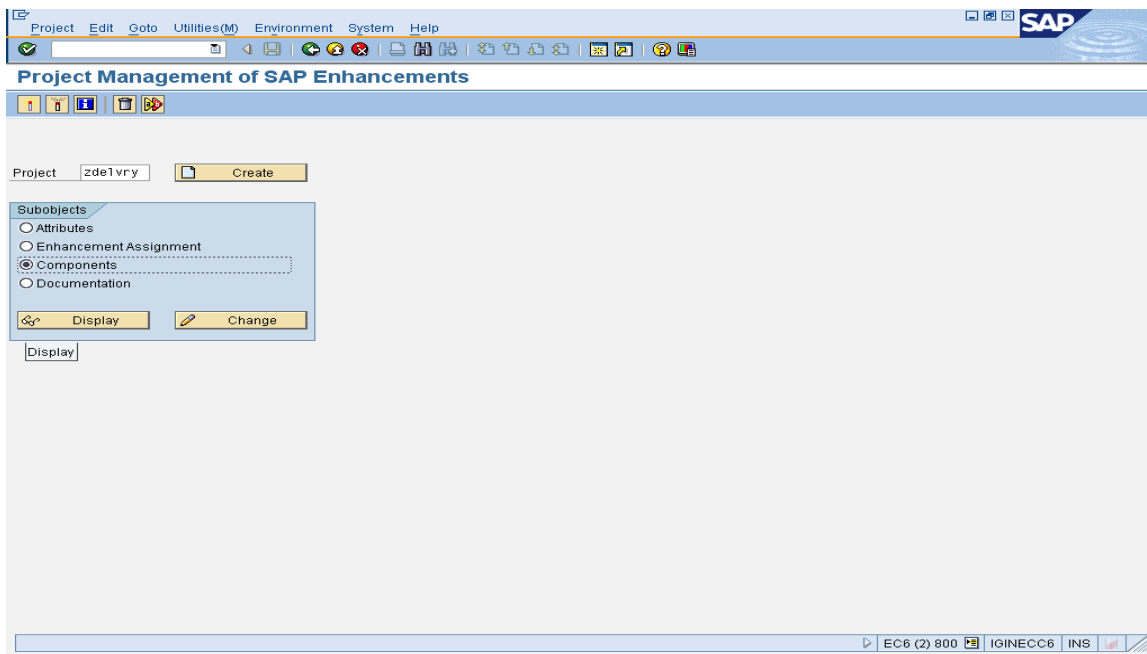


Save and activate include ZXTRKU02.

```

5  *types declarations.
6
7  TYPES: BEGIN OF TY_VBUK,
8         KOSTK TYPE VBUK-KOSTK,
9         KOQUK TYPE VBUK-KOQUK,
10        END OF TY_VBUK.
11
12  DATA: WA_EDIDD TYPE EDIDD,
13         LS_DATA TYPE EDI_SDATA,
14         WA_ZIEDLPI TYPE ZIEDLPI,
15         WA_EIEDL20 TYPE EIEDL20,
16         WA_VBUK TYPE TY_VBUK.
17
18  READ TABLE IDOC_DATA INTO WA_EDIDD WITH KEY SEGNAM = 'ZIEDLPI'.
19  CHECK SY-SUBRC NE 0.
20  READ TABLE IDOC_DATA INTO WA_EDIDD WITH KEY SEGNAM = 'EIEDL22'.
21  CHECK SY-SUBRC = 0.
22  READ TABLE IDOC_DATA INTO WA_EDIDD WITH KEY SEGNAM = 'EIEDL20'.
23  WA_EIEDL20 = WA_EDIDD-SDATA.
24  SELECT KOSTK KOQUK FROM VBUK INTO WA_VBUK
25         WHERE VBELN = WA_EIEDL20-VBELN.
26  ENDSELECT.
27  IF WA_VBUK IS NOT INITIAL.
28  WA_ZIEDLPI-ZKOSTK = WA_VBUK-KOSTK.
29  WA_ZIEDLPI-ZKOQUK = WA_VBUK-KOQUK.
30  IDOC_DATA-SEGNAM = 'ZIEDLPI'.
31  MOVE WA_ZIEDLPI TO IDOC_DATA-SDATA.
32  APPEND IDOC_DATA.
33  CLEAR IDOC_DATA.
34  ENDIF.
    
```

**Note:** The code written above will be executed only if this enhancement component's Project is activated in SMOD.



The marked icons are used to activate and deactivate the exits. If the project is in deactivate mode here then the code in the customer exit will not be executed.

Project			ZDELVRY project for delvry0
Enhancement	Impl	Exp	Y56K0001 Outbound Transporta
Function exit	✓	✓	EXIT_SAPLV56K_001 EXIT_SAPLV56K_002

With this all the necessary configurations on the EDI IDOC front is over.

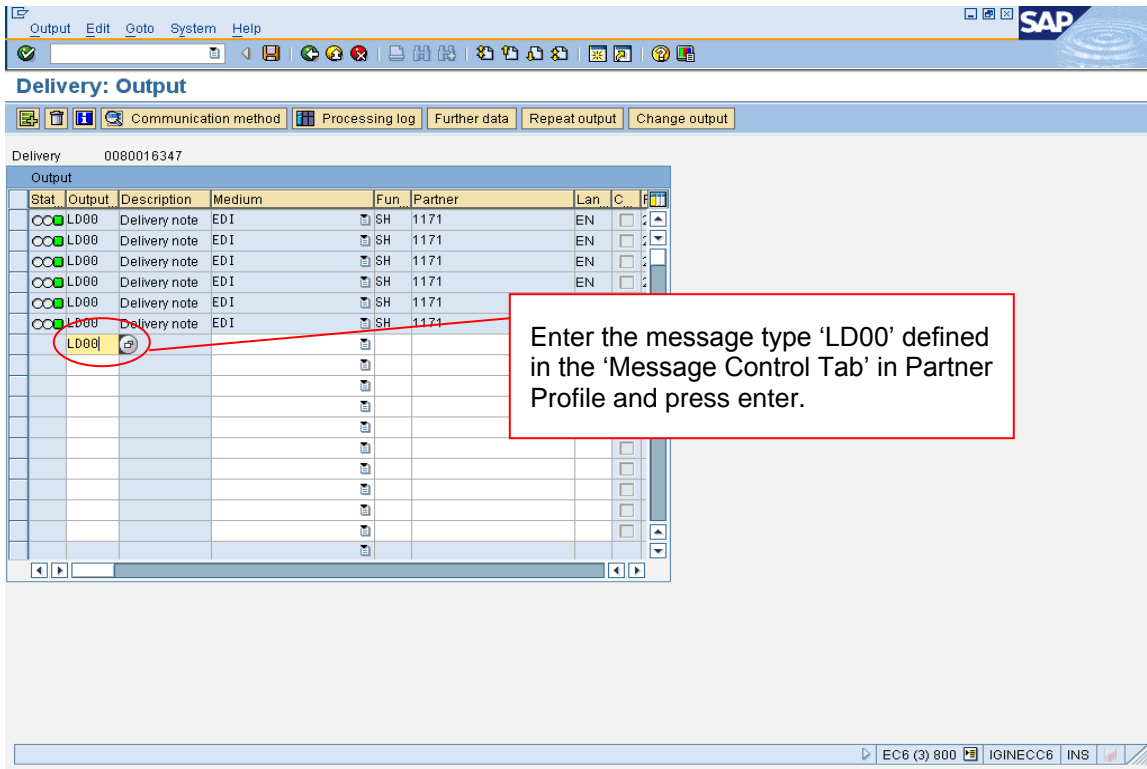
### 7) Configuring the Output Types in VL02N Transaction

Configure the output types in VL02N to trigger the Extension IDOC. To do this, go to Transaction VL02N.

Outbound Delivery: 80016347  
Ship-to party: 1171

Planned GI: 14.11.2008 00:00 Total Weight: 0.02 KG  
Actual GI date: No. of packages:

Item	Material	Deliv. Qty	Un	Description	B	ITyp	P	W	Batch	Val. Type	Open Qty
10	R-1120	1	PC	Kabel mit Schutzkontaktstecker		TAN		C			1

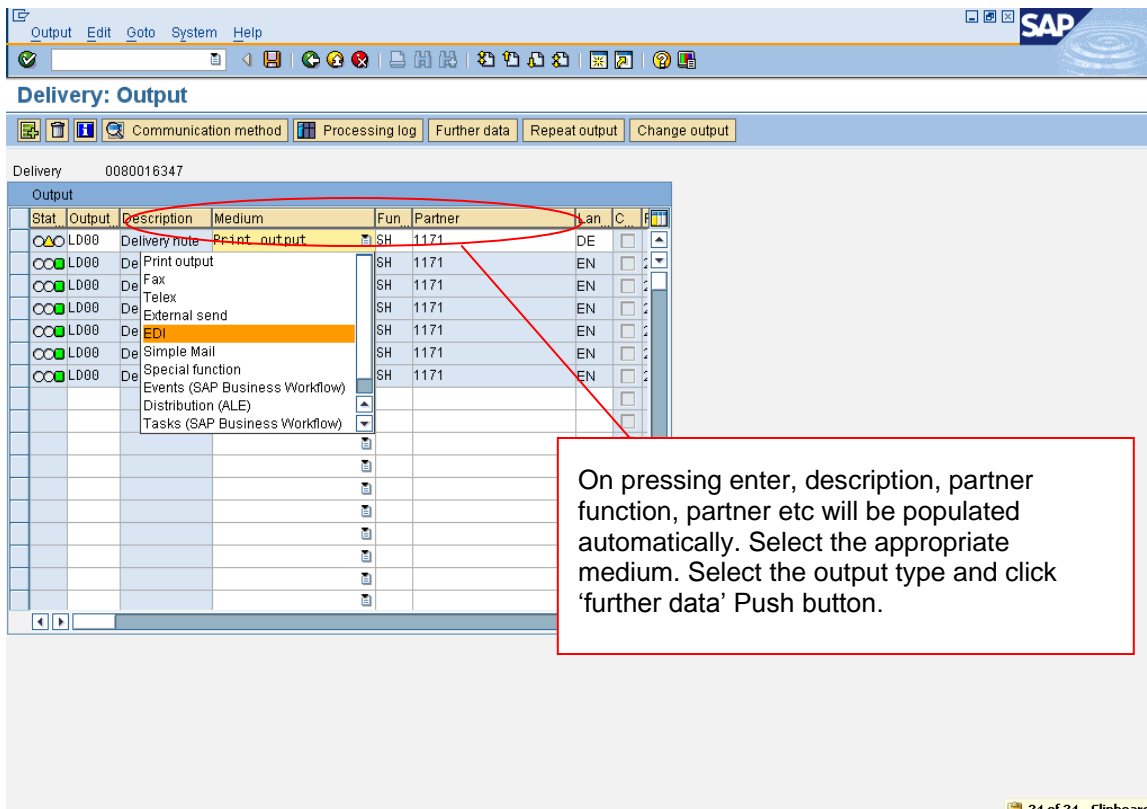


Delivery: Output

Delivery 0080016347

Stat	Output	Description	Medium	Fun	Partner	Lan	C
CO	LD00	Delivery note	EDI	SH	1171	EN	
CO	LD00	Delivery note	EDI	SH	1171	EN	
CO	LD00	Delivery note	EDI	SH	1171	EN	
CO	LD00	Delivery note	EDI	SH	1171	EN	
CO	LD00	Delivery note	EDI	SH	1171	EN	
CO	LD00	Delivery note	EDI	SH	1171	EN	
CO	LD00	Delivery note	EDI	SH	1171	EN	

EC6 (3) 800 | IGINECC6 | INS



Delivery: Output

Delivery 0080016347

Stat	Output	Description	Medium	Fun	Partner	Lan	C
CO	LD00	Delivery note	EDI	SH	1171	DE	
CO	LD00	Print output		SH	1171	EN	
CO	LD00	Fax		SH	1171	EN	
CO	LD00	Telex		SH	1171	EN	
CO	LD00	External send		SH	1171	EN	
CO	LD00	EDI		SH	1171	EN	
CO	LD00	Simple Mail		SH	1171	EN	
CO	LD00	Special function		SH	1171	EN	
CO	LD00	Events (SAP Business Workflow)					
CO	LD00	Distribution (ALE)					
CO	LD00	Tasks (SAP Business Workflow)					

24 of 24 - Clipboard

Select the 'Send Immediately option' and click on back button and save the Delivery.

**Delivery: Output**

Ship-to party: 1171 Hitech AG  
 Output Type: Delivery note EDI

**Creation**  
 29.06.2009 14:35:06

**Requested processing**  
 Dispatch time: **Send immediately (when saving the application)**  
 Time to: 00:00:00

**Actual processing**  
 00:00:00 Not processed

**General data**  
 Manually changed  
 Locked  
 Created by: NTWDEV

**Change Outbound Delivery**

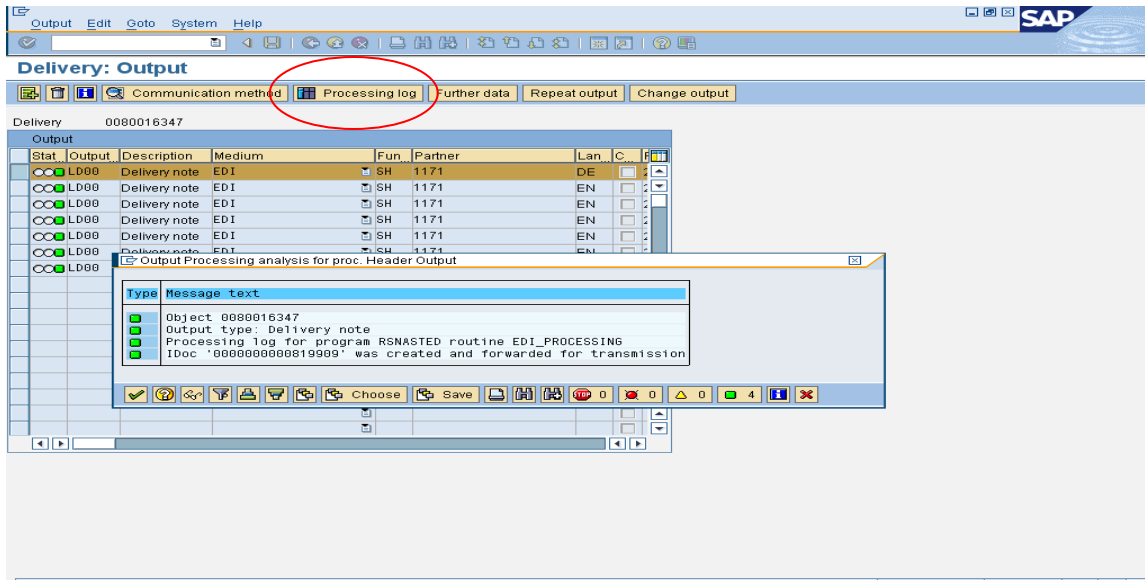
Outbound Delivery: 80016347

Delivery Document has been successfully saved.

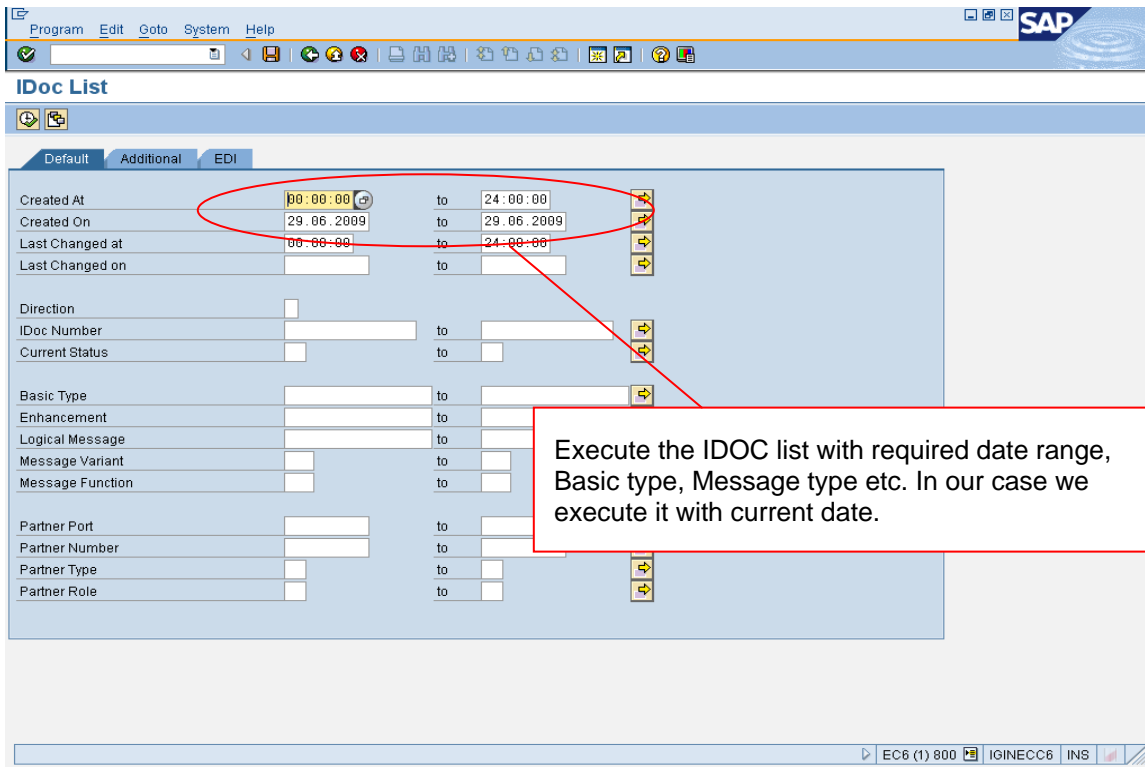
Delivery 80016347 has been saved

### 8) Checking the Created IDOC

When the delivery is saved successfully we can see the IDOC number by selecting the output type and clicking on processing log push button as shown below.



Now to see whether the custom segment has been populated as desired and to know the status of the IDOC, go to transaction WE02.



**Selected IDocs**

IDoc Number	Segm...	Stat	Stat	Partner	BasicType	Date created	Time	Messg...	Direction	Port
000000000819907	4	29	00	LS/ /ZJDE	ZCUST_BAS	29.06.2009	13:58:08	ZCUST_M	Outbox	
000000000819908	4	37	00	LS/ /ZJDE	ZCUST_BAS	29.06.2009	13:59:49	ZCUST_M	Outbox	ZJDE_PORT
000000000819909	26	03	00	KUWE/00000011	DELVRV03	29.06.2009	14:42:59	ZDELVRV	Outbox	PORT_PT

Status Message for Selected IDoc  
 Status Text: IDoc: 000000000819909 Status: Data passed to port OK  
 T100 Text: IDoc sent to SAP system or external program

**Technical short info**

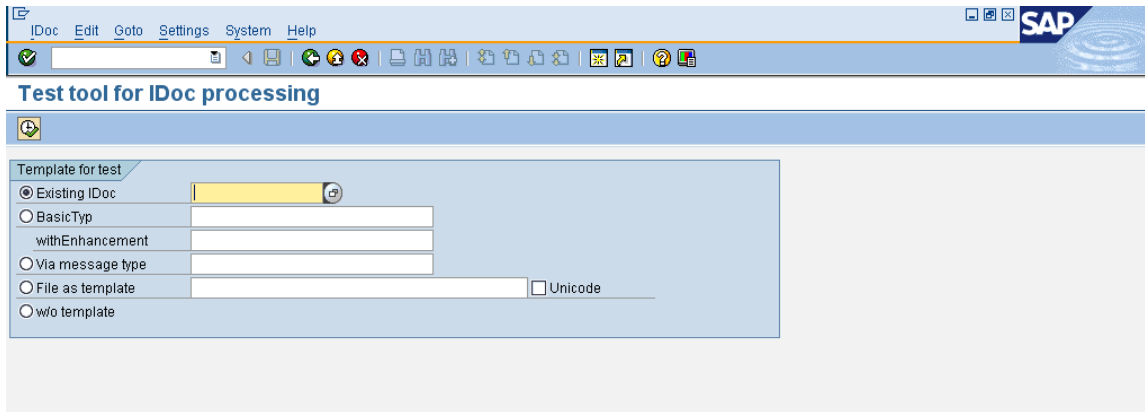
Direction: 1 Outbox  
 Current status: 03  
 Basic type: DELVRV03  
 Extension: ZDELVRV03  
 Message type: ZDELVRV  
 Partner No.: 1171  
 Parth.Type: KU  
 Port: PORT\_PT

**Content of selected segment**

Fid name	Fid cont.
ZKOSTK	C
ZK00UK	C

## 9) Standalone Testing

Use transaction WE19 for testing the IDOC without configuring it to a transaction. Go to transaction WE19.

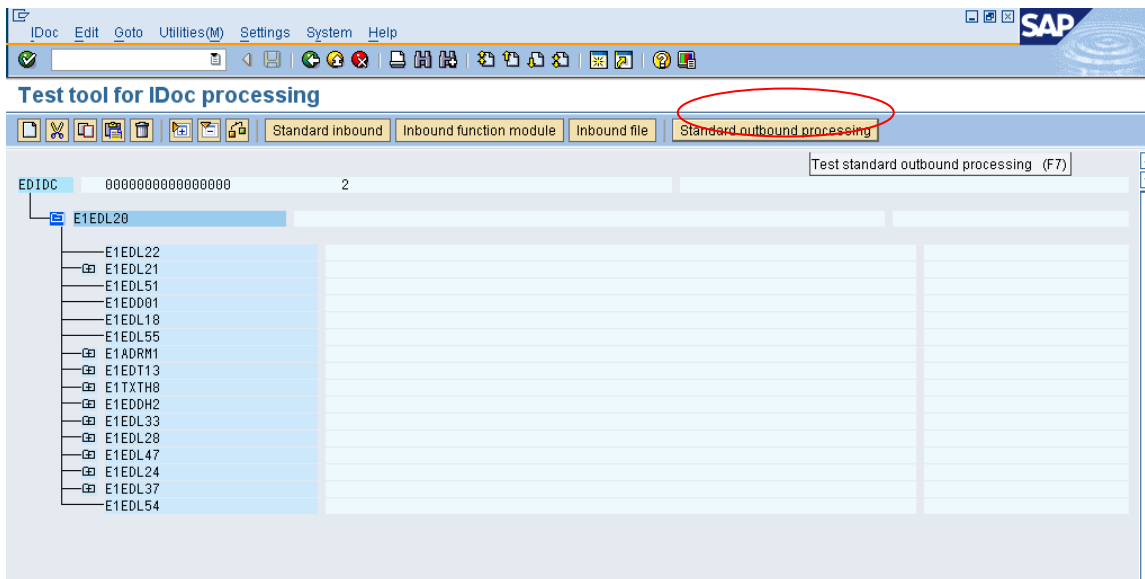


Existing IDOC options - Enter an already created IDOC number and execute.

Bacis type option - Enter any basic type name and execute.

Via message type - Enter a message type name and execute.

Enter the necessary details in the below screen and click on the standard output processing Pushbutton to test the IDOC.



These are the steps involved in creating and testing the custom IDOC.

## Related Content

[Creation of new IDOC types](#)

[Extending an IDoc Type](#)

[Modification Concepts: IDoc Outbound Processing](#)

[Testing IDoc Processing](#)

For more information, visit the [ABAP homepage](#).



## Disclaimer and Liability Notice

This document may discuss sample coding or other information that does not include SAP official interfaces and therefore is not supported by SAP. Changes made based on this information are not supported and can be overwritten during an upgrade.

SAP will not be held liable for any damages caused by using or misusing the information, code or methods suggested in this document, and anyone using these methods does so at his/her own risk.

SAP offers no guarantees and assumes no responsibility or liability of any type with respect to the content of this technical article or code sample, including any liability resulting from incompatibility between the content within this document and the materials and services offered by SAP. You agree that you will not hold, or seek to hold, SAP responsible or liable with respect to the content of this document.