

SDN Community Contribution

(This is not an official SAP document.)

Disclaimer & 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.

Applies To:

SAP R/3 Release 4.6C – Search Help enhancement within standard SAP Search help

Summary

This document explains how to enhance a standard SAP search help in SE11 by developing a custom function module using the search help exit interface of function module F4IF_SHLP_EXIT_EXAMPLE

By: Anand Nanjundaiah

Company: Intelligroup Inc

Date: 27 Nov 2005

Table of Contents

Applies To:.....	2
Summary	2
Table of Contents	2
Background	2
Procedure	3
Code Sample	11
Output Result.....	16
Conclusion.....	18
Author Bio.....	18

Background

When a standard SAP search help needs to be enhanced, the steps involved are:

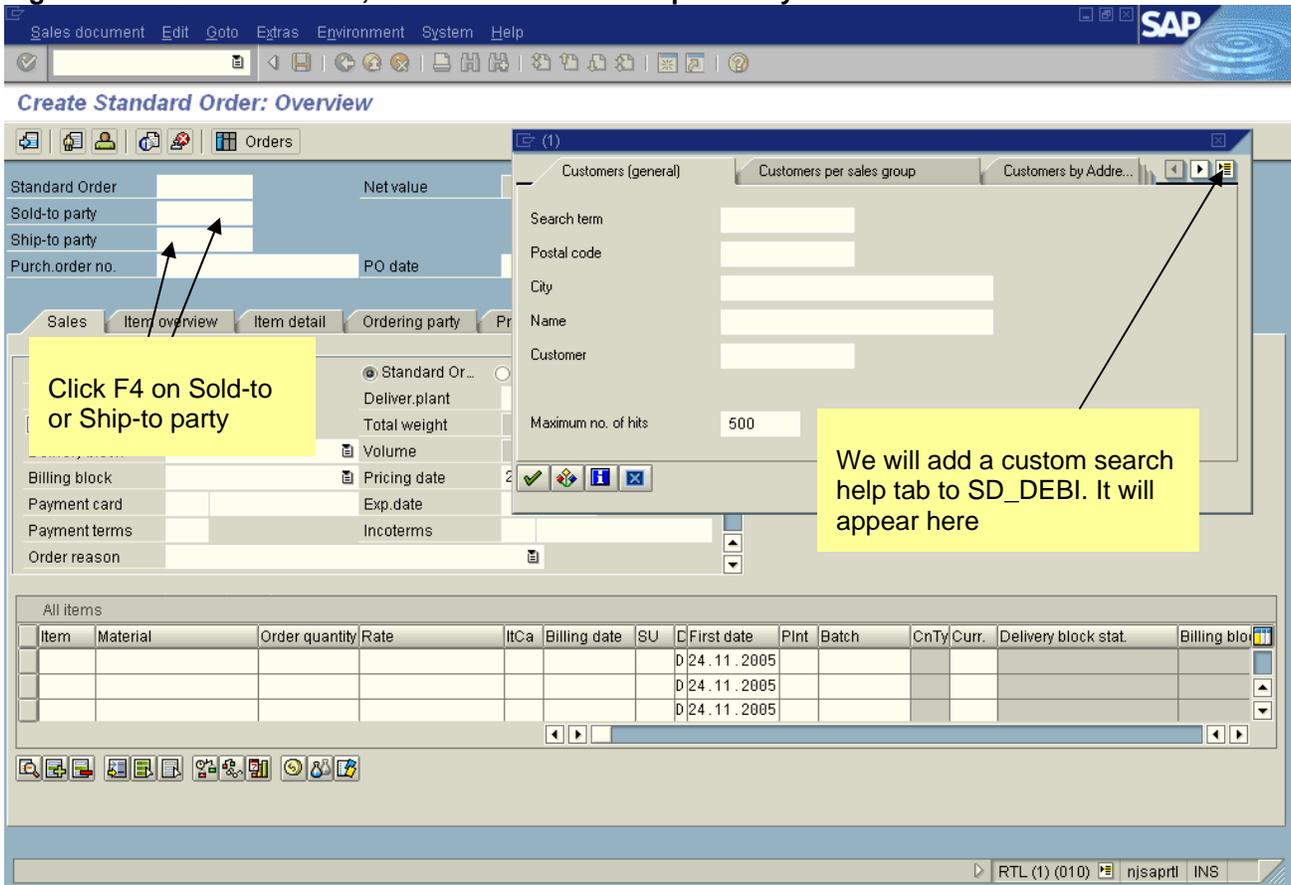
- a) Creating a custom search help and attaching a table or view as a Selection method in the 'Definition' tab
- c) Including the custom search help to the standard search help

However, when the data to be shown in the search help is complex and cannot be achieved by simply creating a view in SE11, we need to develop a custom function module with the data retrieval logic and attach it to our custom search help as Search help exit in the 'Definition' tab. This document explains how to go about doing this.

Procedure

As an example, let's go about developing a custom search help enhancement for standard SAP search help SD_DEBI used in transactions VA01 or VA02. The standard search help SD_DEBI is illustrated in example below, Figure 1. Hit 'F4' help on either the Sold-to or Ship-to party on subsequent screen of VA01, VA02 and a pop-up of SD_DEBI search help appears as shown below:

Figure 1 – Transaction VA01, Hit F4 on Sold-to or Ship-to Party fields



We will include a custom search help enhancement to SD_DEBI. Let's say we want to see the Bill-to party (partner function RE) customers from table KNVP on the hit-list. When we are done, the search help selection screen will appear as another tab on this standard SAP F4 help.

To create our enhancement, following are the steps:

Step 1 - Create a custom Function Group in SE37 called ZSHLP

Step 2 -

1) In order to create a search help enhancement function, we need the interface parameters of function

module F4IF_SHLP_EXIT_EXAMPLE (of Function Group SDHI). Figures 2 & 3 show the 'Changing' and 'Tables' parameters of F4IF_SHLP_EXIT_EXAMPLE.

Figure 2 – SE37 - Function Module – F4IF_SHLP_EXIT_EXAMPLE – Changing parameters

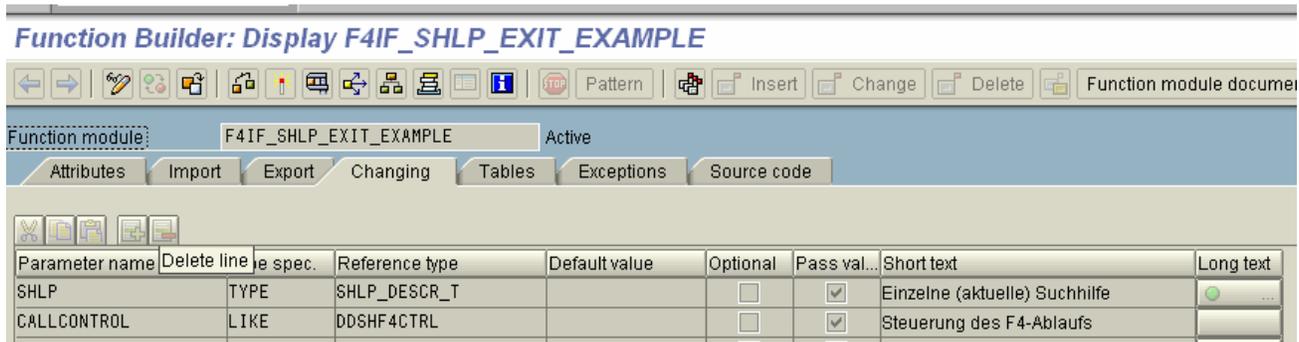
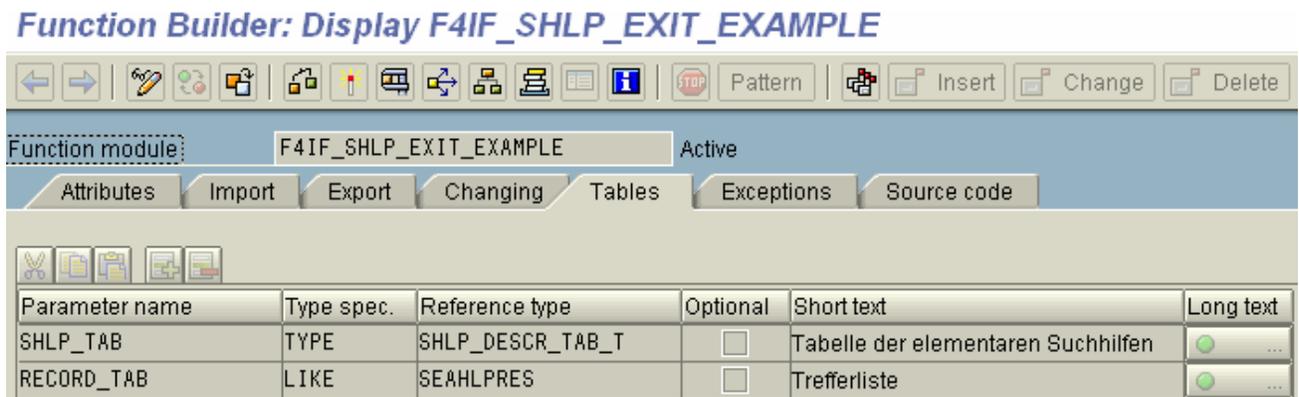
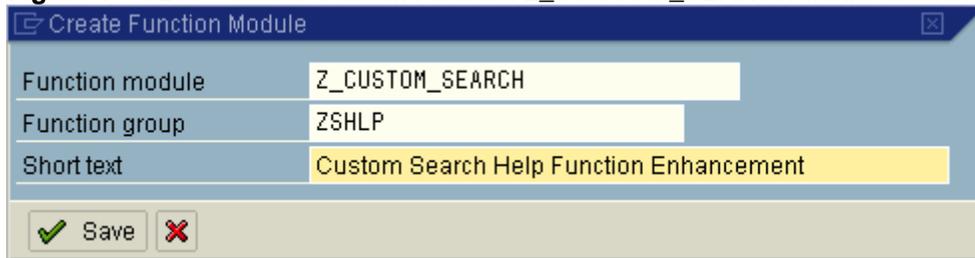


Figure 3 – SE37 - Function Module – F4IF_SHLP_EXIT_EXAMPLE – Tables parameters



2) Create a function module Z_CUSTOM_SEARCH within our custom function group ZSHLP (Figure 4).

Figure 4 – SE37 - Create Custom Function Z_CUSTOM_SEARCH within function group ZSHLP



3) We need to create a similar set of parameters within Z_CUSTOM_SEARCH, as exists in F4IF_SHLP_EXIT_EXAMPLE. This is shown in Figures 5 & 6.

Figure 5 – SE37 – Function Module Z_CUSTOM_SEARCH with similar ‘Changing’ parameters as function module F4IF_SHLP_EXIT_EXAMPLE

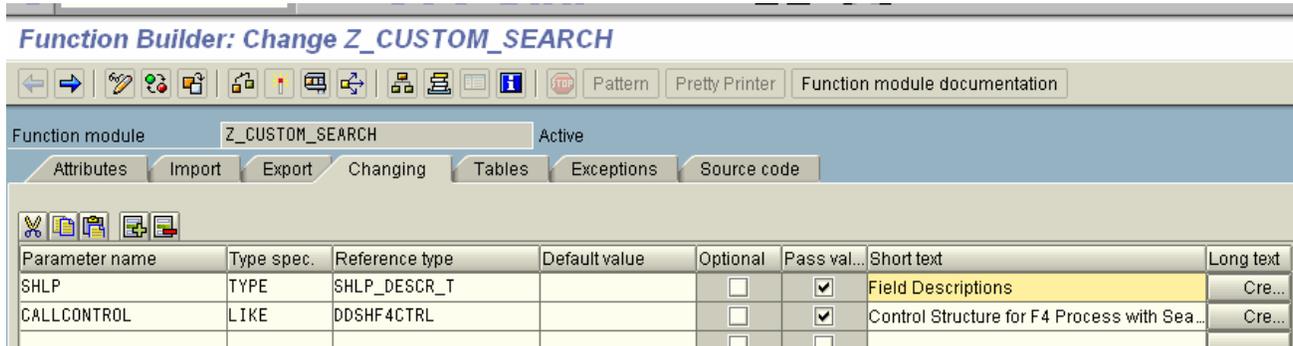


Figure 6 – SE37 - Function Module Z_CUSTOM_SEARCH with similar ‘Tables’ parameters as function module F4IF_SHLP_EXIT_EXAMPLE

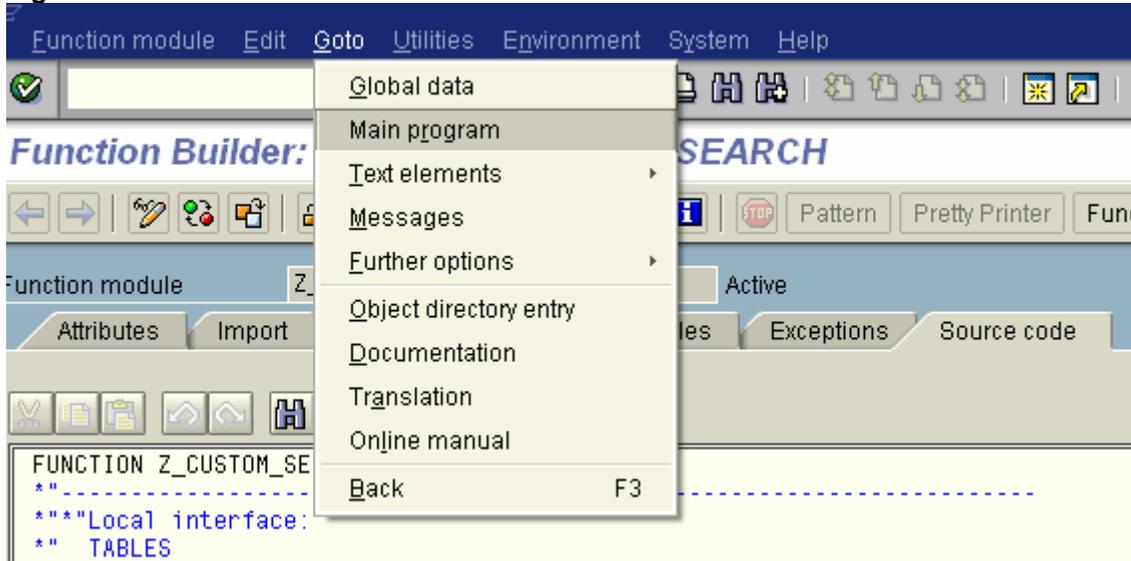


Step 3 –

1) Within Function Module Z_CUSTOM_SEARCH go to Main Program as shown in Figure 7 and in the main program TOP include, in this case INCLUDE LZSHLPTOP insert the following line

```
type-pools shlp.
```

Figure 7 – SE37



2) Save and activate the function module. You have the search help enhancement function framework ready

Step 4 –

1) Go to SE11 – create a custom elementary search help called ZS_BILLTO

2) Assign the fields that you want to display on the selection condition as well as on the hit list as shown below. LPos is the Hit-list position and SPos is the order of the fields on the search help selection screen.

Elementary srch hlp: ZS_BILLTO Active
Short description: Custom Bill To Search

Search help exit: Z_CUSTOM_SEARCH

Search help parameter	IMP	EXP	LPos	SPos	SDis	Data element	M...	Default va
KTOKD	<input checked="" type="checkbox"/>	<input type="checkbox"/>		1	<input type="checkbox"/>	KTOKD		
SO RTL	<input checked="" type="checkbox"/>	<input type="checkbox"/>		2	<input type="checkbox"/>	SO RTL		
MCOD1	<input checked="" type="checkbox"/>	<input type="checkbox"/>		3	<input type="checkbox"/>	MCOD1		
KUNNR	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	1	5	<input type="checkbox"/>	KUNNR		
NAME1	<input type="checkbox"/>	<input type="checkbox"/>	2		<input type="checkbox"/>	NAME1 GP		
ORT01	<input checked="" type="checkbox"/>	<input type="checkbox"/>	3	4	<input type="checkbox"/>	ORT01 GP		
STRAS	<input type="checkbox"/>	<input type="checkbox"/>	4		<input type="checkbox"/>	STRAS GP		
KUNN2	<input type="checkbox"/>	<input type="checkbox"/>	5		<input type="checkbox"/>	KUNN2		

3) Hit list fields in the ascending order of LPos values:

- 1 - Ship To Customer Number - KUNNR (Make sure this is an export field as this is what we want to put back on the Ship-to or Sold-to party fields after the user double clicks on a value on the hit list)
- 2 - Ship to Customer Name – NAME1
- 3 - City – ORT01
- 4 - Street – STRAS
- 5 - Bill to Customer Number – KUNN2

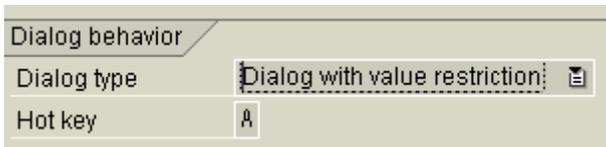
4) The SPos position dictates the order in which the restriction fields will appear on the selection screen in the search help tab. In the ascending order of SPos values:

- 1 - Account group - KTOKD
- 2 - Search term - SORTL
- 3 - Name – MCOD1
- 4 - City – ORT01
- 5 - Customer Number - KUNNR

5) As shown above, assign the custom function module Z_CUSTOM_SEARCH as Search help exit. We are not using any Selection method for this example, since our selection is being done within the function module.



6) Since we want to restrict values in our example, we set the Dialog Type to “Dialog with value restriction” and any Hot key as desired.

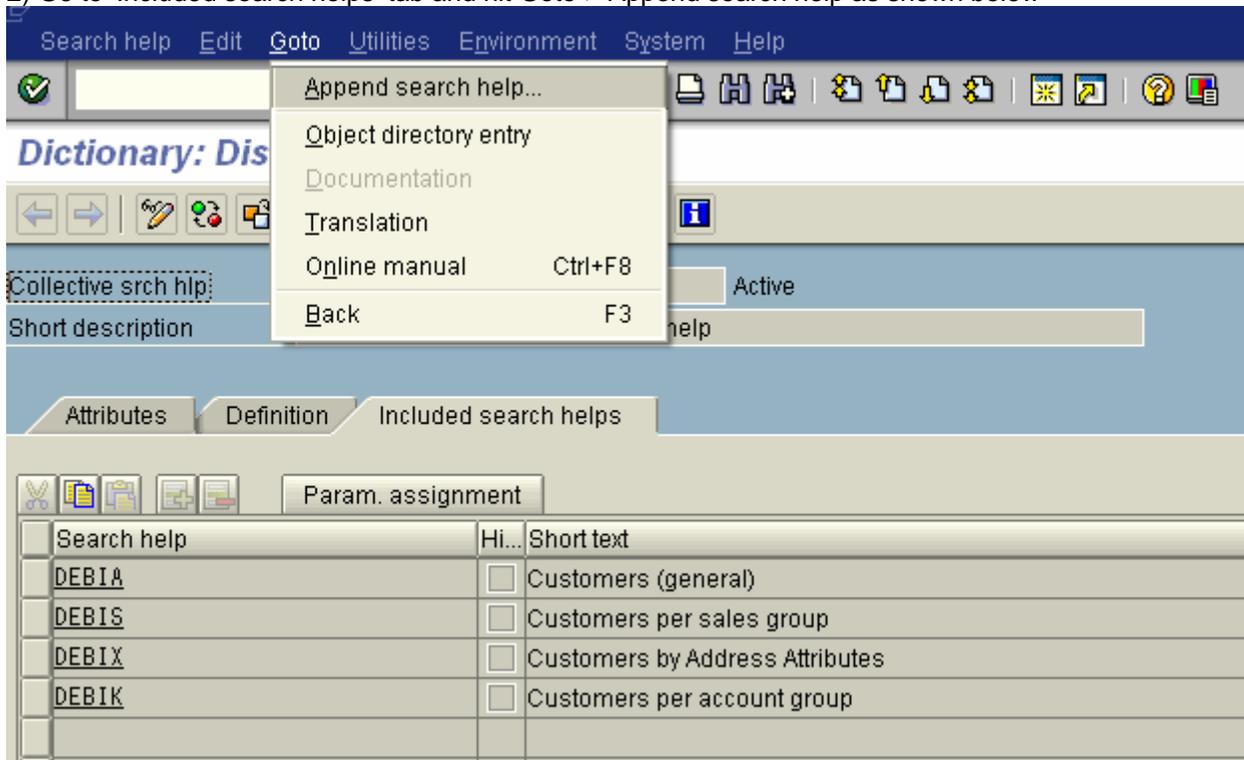


Step 5 –

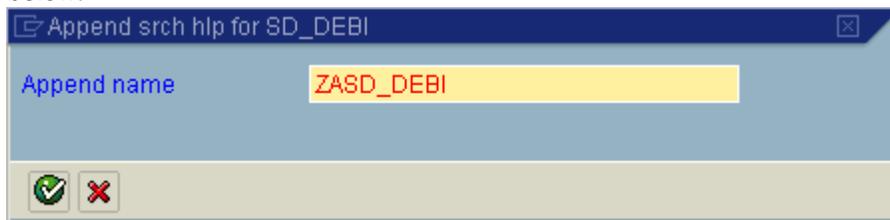
1) Go to SE11 – to Search help SD_DEBI and hit 'Display' button



2) Go to 'Included search helps' tab and hit Goto-> Append search help as shown below



3) You will be prompted to append the SD_DEBI search help with a proposed name ZASD_DEBI as shown below.



Step 6 –

If you read the standard SAP help for the enhancement as shown in the link below, it mentions how each of the events come into play.

http://help.sap.com/saphelp_46c/helpdata/EN/cf/21ee2b446011d189700000e8322d00/frameset.htm

Calling the Search Help Exit

If a search help exit is assigned to a search help, the help processor calls it at the following times:

Before Displaying the Dialog Box for Selecting the Required Search Path.

It is only called for collective search helps. Using the search help exit, the set of elementary search helps available can for example be restricted depending on the context.

Before Displaying the Dialog Box for Entering Search Conditions.

You can either influence the dialog for entering search conditions or skip it altogether here. You can also influence how the selection screen looks.

Before Selecting Data.

The data selection can be partly or completely copied from the search help exit. This can become necessary if the data selection cannot be implemented with a SELECT statement for a table or a view.

Before Displaying the Hit List.

You can influence the display of the hit list in this step with the search help exit. You can reduce the number of values displayed here. For example, you can display only values for which the person calling the input help has authorization. You can also copy the complete hit list from the search help exit.

Before Returning the Values Selected by the User to the Input Template

It could be advisable to intervene at this time if control of the further transaction flow should depend on the value selected. A typical example is setting set/get parameters.

We are writing our code in the event of “Before Selecting Data” when callcontrol-step = ‘SELECT’. If you wanted to further restrict the data in the hit list, you could filter it when callcontrol-step = ‘DISP’.

Step 7 – Within Function Module Z_CUSTOM_SEARCH put in the custom code for retrieving data within the ‘SELECT’ scenario of CALLCONTROL-STEP as shown in source code below. Make sure to set the next event as CALLCONTROL-STEP = ‘DISP’ within the code.

Code Sample

```
-----BEGIN OF CODE SAMPLE -----  
-----BEGIN OF INCLUDE LZSHLPTOP -----  
  
FUNCTION-POOL zshlp.                "MESSAGE-ID ..
```

TYPE-POOLS shlp.

TYPES:

```
BEGIN OF t_knvp,
  kunnr TYPE kna1-kunnr,
  name1 TYPE kna1-name1,
  ort01 TYPE ort01_gp,
  stras TYPE stras_gp,
  kunn2 TYPE knvp-kunn2,
  name1_2 TYPE kna1-name1,
END OF t_knvp.
```

```
DATA: i_knvp TYPE TABLE OF t_knvp,
      wa_knvp TYPE t_knvp,
      wa_selopt TYPE ddsyselopt,
      wa_fielddescr TYPE dfies.
```

DATA:

```
rc TYPE i,
v_tabix LIKE sy-tabix.
```

```
RANGES: r_ktokd FOR kna1-ktokd,
         r_mcod1 FOR kna1-name1,
         r_sort1 FOR kna1-sort1,
         r_kunnr FOR kna1-kunnr,
         r_ort01 FOR kna1-ort01.
```

-----END OF INCLUDE LZSHLPTOP -----

-----BEGIN OF FUNCTION MODULE Z_CUSTOM_SEARCH -----

FUNCTION z_custom_search.

```
*"-----
*"*Local interface:
*" TABLES
*"   SHLP_TAB TYPE SHLP_DESCR_TAB_T
*"   RECORD_TAB STRUCTURE SEAHLPRES
*" CHANGING
*"   VALUE(SHLP) TYPE SHLP_DESCR_T
*"   VALUE(CALLCONTROL) LIKE DDSHF4CTRL STRUCTURE DDSHF4CTRL
*"-----
```

```
* EXIT immediately, if you do not want to handle this step
CASE callcontrol-step.
```

```
*"-----
* STEP SELECT (Select values)
*"-----
```

```
* This step may be used to overtake the data selection completely.
* To skip the standard selection, you should return 'DISP' as following
* step in CALLCONTROL-STEP.
* Normally RECORD_TAB should be filled after this step.
```

```
WHEN 'SELECT'.
```

- * **Change column header texts appearing on the search help hit list**

```
LOOP AT shlp-fielddescr INTO wa_fielddescr.
  v_tabix = sy-tabix.

  CASE wa_fielddescr-fieldname.

    WHEN 'KUNNR'.
      wa_fielddescr-fieldtext = 'ShipToCustomer#'.
      wa_fielddescr-reptext   = 'ShipToCustomer#'.
      wa_fielddescr-scrtext_s = 'ShipTo #'.
      wa_fielddescr-scrtext_m = 'ShipToCustomer#'.
      wa_fielddescr-scrtext_l = 'ShipToCustomer#'.
      MODIFY shlp-fielddescr FROM wa_fielddescr
      INDEX v_tabix TRANSPORTING fieldtext reptext scrtext_s
      scrtext_m scrtext_l.

    WHEN 'KUNN2'.
      wa_fielddescr-reptext   = 'BillToCustomer#'.
      wa_fielddescr-fieldtext = 'BillToCustomer#'.
      wa_fielddescr-scrtext_s = 'BillTo #'.
      wa_fielddescr-scrtext_m = 'BillToCustomer #'.
      wa_fielddescr-scrtext_l = 'BillToCustomer #'.
      MODIFY shlp-fielddescr FROM wa_fielddescr
      INDEX v_tabix TRANSPORTING fieldtext reptext scrtext_s
      scrtext_m scrtext_l.

    WHEN 'NAME1'.
      wa_fielddescr-fieldtext = 'ShipToCustomer Name'.
      wa_fielddescr-reptext   = 'ShipToCustomer Name'.
      wa_fielddescr-scrtext_s = 'ShipTo Name'.
      wa_fielddescr-scrtext_m = 'ShipToCustomer Name'.
      wa_fielddescr-scrtext_l = 'ShipToCustomer Name'.
      MODIFY shlp-fielddescr FROM wa_fielddescr
      INDEX v_tabix TRANSPORTING fieldtext reptext scrtext_s
      scrtext_m scrtext_l.

  ENDCASE.

ENDLOOP.
```

- * **Select the Bill to party customer based on the select options**

```
FREE: r_ktokd, r_kunnr, r_sort1, r_mcod1, r_ort01, i_knvp.

LOOP AT shlp-selopt INTO wa_selopt.
```

- * **Build a Range for the 5 selection options of the search help**

```
CASE wa_selopt-shlpfield.
  WHEN 'KTOKD'.
    r_ktokd-sign   = wa_selopt-sign.
    r_ktokd-option = wa_selopt-option.
    r_ktokd-low    = wa_selopt-low.
    r_ktokd-high   = wa_selopt-high.
```

```

APPEND r_ktokd.
CLEAR r_ktokd.
WHEN 'KUNNR'.
  r_kunnr-sign = wa_selopt-sign.
  r_kunnr-option = wa_selopt-option.
  r_kunnr-low = wa_selopt-low.
  r_kunnr-high = wa_selopt-high.
  APPEND r_kunnr.
  CLEAR r_kunnr.
WHEN 'SORTL'.
  r_sortl-sign = wa_selopt-sign.
  r_sortl-option = wa_selopt-option.
  r_sortl-low = wa_selopt-low.
  r_sortl-high = wa_selopt-high.
  APPEND r_sortl.
  CLEAR r_sortl.
WHEN 'MCOD1'.
  r_mcod1-sign = wa_selopt-sign.
  r_mcod1-option = wa_selopt-option.
  r_mcod1-low = wa_selopt-low.
  r_mcod1-high = wa_selopt-high.
  APPEND r_mcod1.
  CLEAR r_mcod1.
WHEN 'ORT01'.
  r_ort01-sign = wa_selopt-sign.
  r_ort01-option = wa_selopt-option.
  r_ort01-low = wa_selopt-low.
  r_ort01-high = wa_selopt-high.
  APPEND r_ort01.
  CLEAR r_ort01.

```

ENDCASE.

ENDLOOP.

* Retrieve data from KNVP table for the above selected ranges

* Doing query to retrieve data for the search help

```

SELECT knvp~kunnr
      kna1~name1
      kna1~ort01
      kna1~stras
      knvp~kunn2
INTO TABLE i_knvp
FROM knvp
INNER JOIN kna1
ON knvp~kunnr = kna1~kunnr
WHERE
  knvp~parvw = 'RE' AND " Bill to Party
  knvp~kunnr IN r_kunnr AND
  kna1~ktokd IN r_ktokd AND
  kna1~sortl IN r_sortl AND
  kna1~mcod1 IN r_mcod1 AND
  kna1~ort01 IN r_ort01.

```

```
CHECK sy-subrc = 0.
DELETE ADJACENT DUPLICATES FROM i_knvp.

* Select the short text for kunn2 from kna1.
* Move all the selected records to Record_Tab
LOOP AT i_knvp INTO wa_knvp.
  v_tabix = sy-tabix.
  SELECT SINGLE name1 FROM kna1
  INTO wa_knvp-name1_2
  WHERE kunnr = wa_knvp-kunnr.

  MOVE wa_knvp TO record_tab-string.
  APPEND record_tab.
  CLEAR record_tab.
  MODIFY i_knvp FROM wa_knvp INDEX v_tabix.
  CLEAR wa_knvp.

ENDLOOP.

rc = 0.

IF rc = 0.
  callcontrol-step = 'DISP'.
ELSE.
  callcontrol-step = 'EXIT'.
ENDIF.

EXIT. "Don't process STEP DISP additionally in this call.

WHEN 'PRESEL1'.

WHEN 'DISP'.

WHEN OTHERS.

ENDCASE.

ENDFUNCTION.

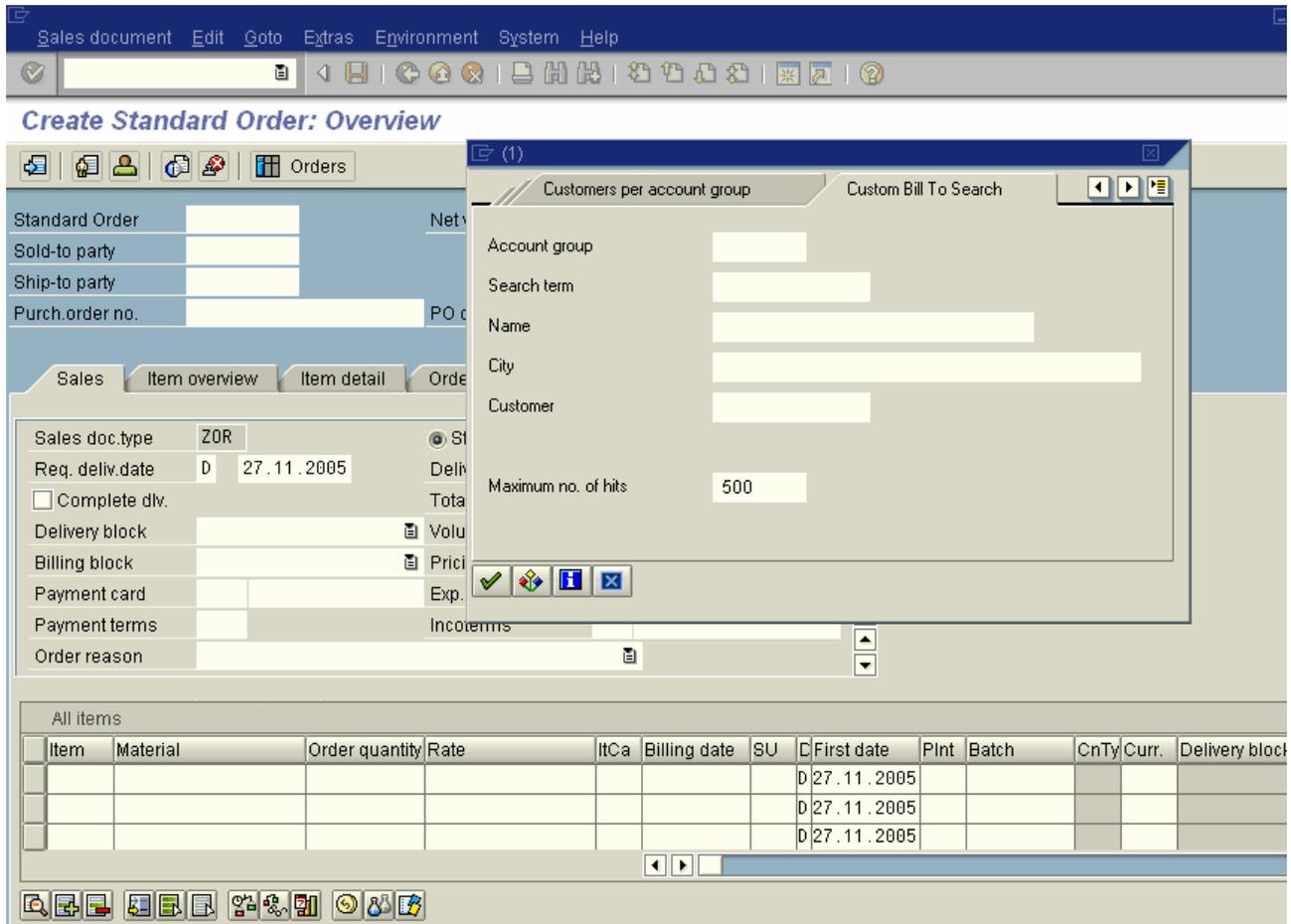
-----END OF FUNCTION MODULE Z_CUSTOM_SEARCH -----

-----END OF CODE SAMPLE -----
```

Save and activate at every step.

Output Result

1) After doing all of the above steps, its time to see the output! Go to transaction VA01, enter details like Order Type and hit 'Enter' to go to the second screen. Place cursor on the Ship-to party field at the header section and hit 'F4'. The last tab shows our custom search help 'Custom Bill To Search'.



The screenshot shows the SAP VA01 'Create Standard Order: Overview' screen. A search help dialog box is open over the 'Ship-to party' field. The dialog has two tabs: 'Customers per account group' and 'Custom Bill To Search'. The 'Custom Bill To Search' tab is active, showing search criteria: Account group, Search term, Name, City, and Customer. The 'Maximum no. of hits' is set to 500. Below the dialog, the 'All items' table is visible with columns for Item, Material, Order quantity, Rate, ItCa, Billing date, SU, C, First date, Plnt, Batch, CnTy, Curr., and Delivery block. The table contains three rows of data with 'First date' values of 27.11.2005.

Item	Material	Order quantity	Rate	ItCa	Billing date	SU	C	First date	Plnt	Batch	CnTy	Curr.	Delivery block
								27.11.2005					
								27.11.2005					
								27.11.2005					

2) Enter restriction values and hit enter to see the hit list result

The screenshot shows the SAP 'Create Standard Order: Overview' dialog box. A search window is open, displaying a list of 100 entries found. The table below shows the search results:

ShipTo #	ShipToCustomer Name	City	Street	BillTo #
1	DP USA	Edison	street1	2
6	Ebay Inc	Edison		1
11	ACME	Edison	499 thornall st	12
12	SteelCORP	Brandrapids	5 1431, Hidden valley drive	20
13	Dezzer Inc	Cedar Rapids	1540 28th street	45
14	Z Company	edison	499 Thornall Street0	17
15	Z1 Company	edison	499 Thornall Street	15
16	Ameritrade	Woodbridge	1 Mutton Hollow Road	16
17	AKME	Kentwood	1431-11 Hidden Valley Dr	17
22	Pocahontas	Sioux	kingsport streets	22
276	Group01	Edison	1236 Pershing Dr	276
286	Test bw Extraction Customer	Iselin	499 Thornall Street	286

Below the search results, there is an 'All items' table with columns: Item, Material, Order quantity, Rate, ItCa, Billing date, SU, C, First date, Plnt, Batch, CnTy, Curr., and Delivery block stat.

3) Double clicking on any line, will bring back the value to the screen as seen below:

The screenshot shows the 'Create Standard Order: Overview' dialog box after a double-click on a search result. The 'Ship-to party' field is now populated with the value '14'. The 'Net value' field shows '0.00'.

Conclusion

From the above example, we are able to filter based on the selection fields, change the column headers' text appearing on the hit list and content of internal table RECORD_TAB, the contents of which are shown on the hit list. As you can see, the search help exit is a powerful feature that greatly enhances the whole search help functionality and makes life easier for business. You can do this for any standard SAP search help which will reflect across all transactions which use that search help, thus making it global.

Author Bio



Anand Nanjundaiah is a Principal Consultant with Intelligroup Inc. He is currently working as a Team Lead on a SAP implementation project in the US.