

Steps for Finding User-Exit or Badi's



Applies to:

SAP ECC 6.0. For more information, visit the [ABAP homepage](#).

Summary

This document helps people to find the corresponding user exit or BADI available for SAP standard transaction code.

Author: Sai Ram Reddy Neelapu

Company: Atos Origin - Singapore

Created on: 23 March 2011

Author Bio

Sai Ram Reddy Neelapu is working as Sr. ABAP Consultant in Atos Origin for past 5+ years.

Table of Contents

Purpose of Enhancements	3
Steps to find USEREXIT or BADI	3
Method 1: (User Exits)	3
Method 2: (Function Module Exit)	5
Method 3: (Identifying BADI)	7
Related Content	11
Disclaimer and Liability Notice	12

Purpose of Enhancements

User-Exits or BADI allow you to add own functionality to SAP standard screen without modifying the standard program.

User-exits are generally collected in includes and attached in standard program by SAP.

As this document is intended mainly to find the list of user exits or BADI that are tied to standard transaction code, I don't want to talk much about the technical side.

Steps to find USEREXIT or BADI

List of Transaction code related to USEREXIT and BADI's

- **CMOD**
- **SMOD**
- **SE18** → Business Add-Ins: Definitions

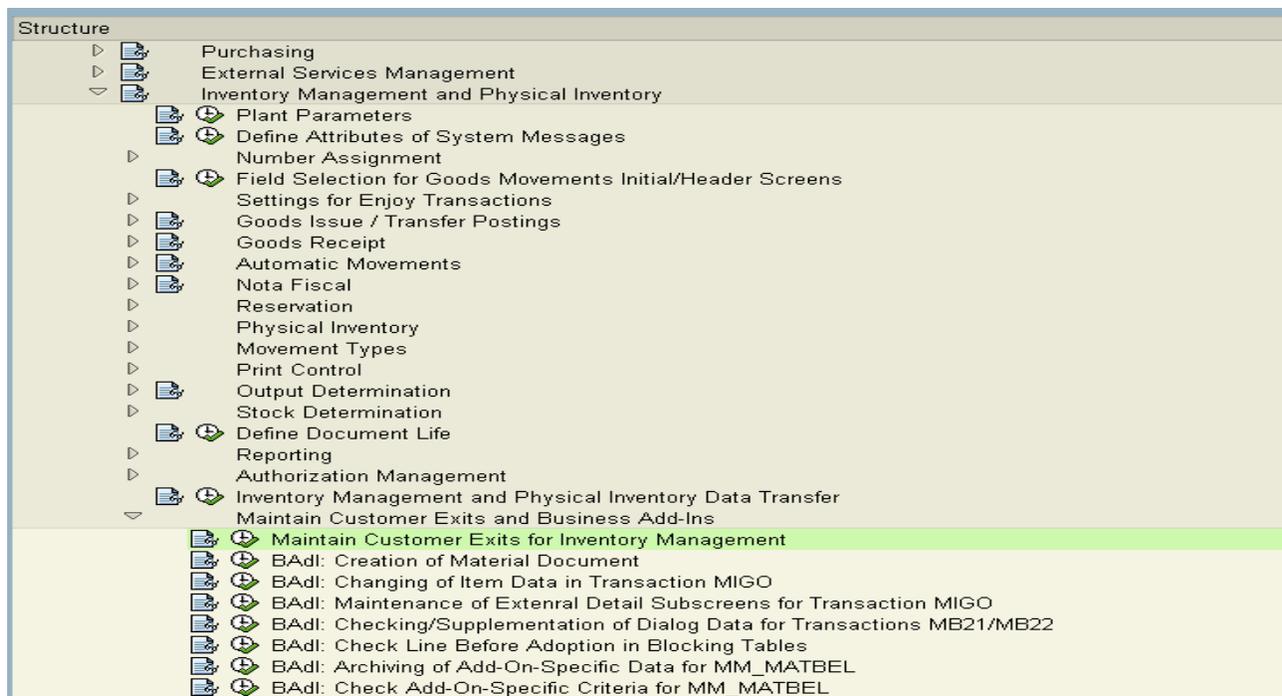
Firstly, I will show you how to find the list of User-Exit that are available for standard transaction code.

Note: There are few different ways of finding out user-exit or BADI's, out of which below scenarios are few ways of doing it.

Method 1: (User Exits)

Business Scenario: When plant 101 store users are performing 311 movement type transfer from store to production in loc, they will like to auto populate the plant and material number into the text field.

Step 1: Go to the relevant **SPRO** and look for the documentation.



Step 2: Click on documentation button , it will display all the available user exits.



The screenshot shows a window titled 'Performance Assistant' with a toolbar containing various icons. The main content area has a yellow background and is titled 'Maintain Customer Exits for Inventory Management'. It contains a list of customer exits and their descriptions, followed by a section on activities and further notes.

Maintain Customer Exits for Inventory Management

For Inventory Management, SAP provides the following **customer exits**:

- **MB_CF001** Updating of material document data upon posting
- **MBCF0002** Filling the item text in the material document
- **MBCFC003** Maintaining batch master data upon goods movements
- **MBCFC004** Maintaining batch specifications upon goods movements
- **MBCF0005** Filling the item data on goods receipt/ issue slips
- **MBCF0006** Transferring the number of the WBS element for subcontracting
- **MBCF0007** Posting a reservation
- **MBCF0009** Filling the *Storage location* field
- **IQSM0007** Serial numbers, user exit for goods movements
- **XMBF0001** Stock determination:
 - Changing the stock determination rule
 - Adjusting the stock determination item table

Activities

1. Create the enhancement.

To do this, either create a new project or use an existing project.

2. Activate the project.

Your enhancement will not become effective until you have activated it.

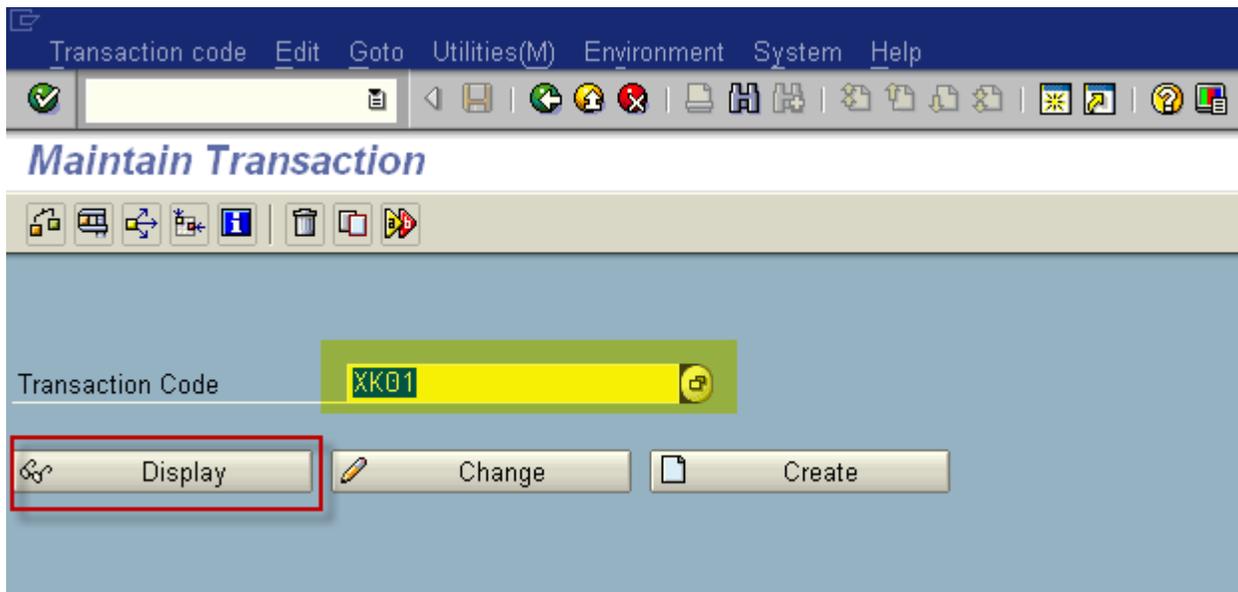
Further notes

Unlike modifications, enhancements are basically release-insensitive, since they are

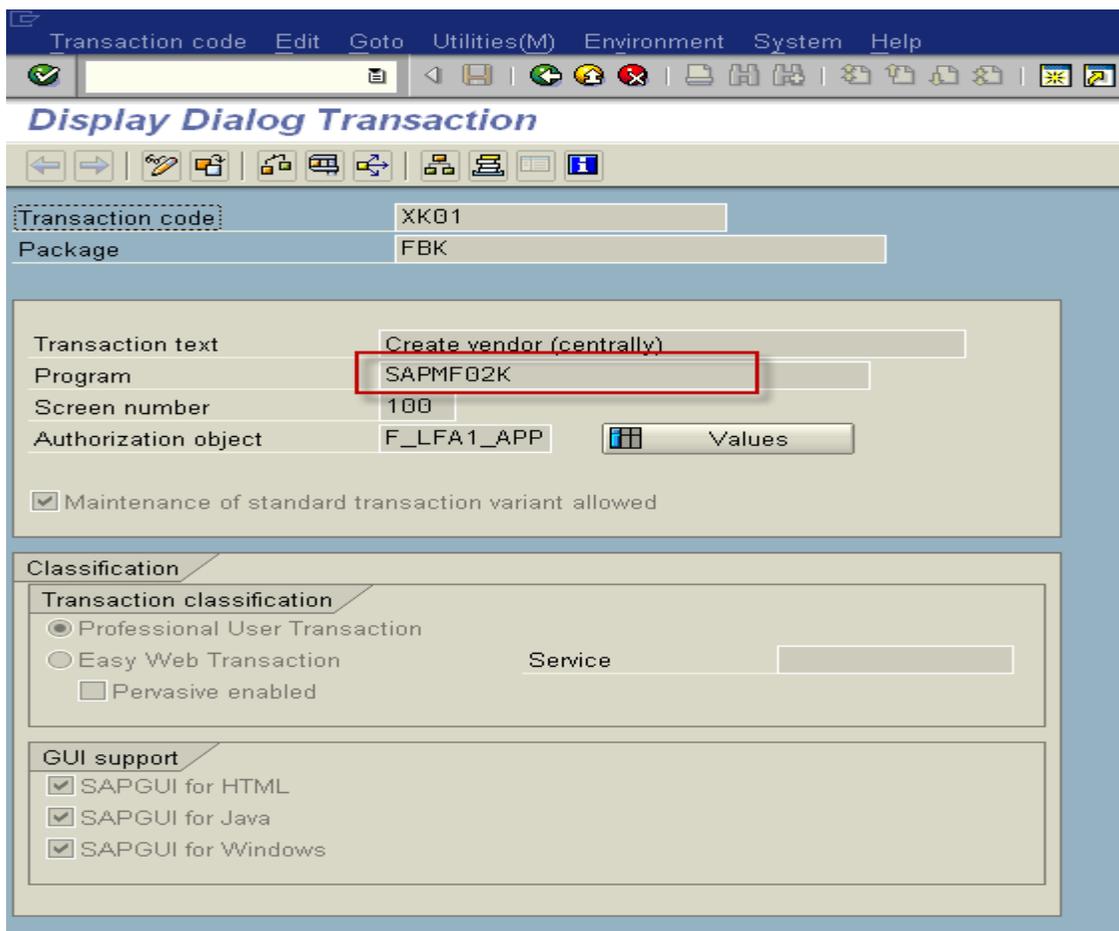
Method 2: (Function Module Exit)

Business Scenario: You want to check, whether there is any suitable exit available before vendor creation using transaction code **XK01**.

Step 1: Go to transaction code **SE93**, enter transaction code **XK01**. Press Display

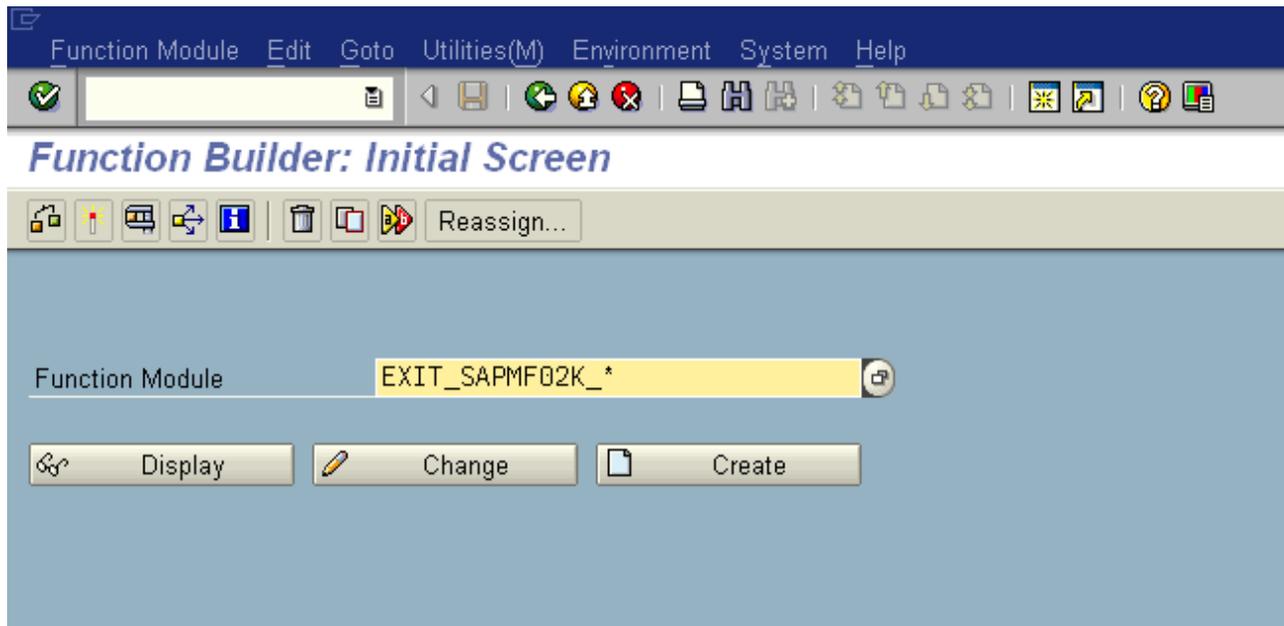


Step 2: Copy the program name that is tied to the transaction code.



Step 3: Go to transaction code **SE37**, enter Function module name as **EXIT_<Program name>_*** as shown below.

Note: Here program name which is identified in step 2.



Step 4: Press F4, value help. You will see all the available list of function modules.

Function group	Function group short text
Function Module Name	Short text for function module
XF05	User Exit: Vendor Master Data
EXIT_SAPMF02K_001	Vendors: User Exit for Checks prior to Saving
XPS0	Local Authority-Specific User Exits
EXIT_SAPMF02K_800	Check Vendor Address Data
EXIT_SAPMF02K_801	Check Vendor Bank Details

As this document aim is to identify the user exits, I am not explaining much about the implementation of these user-exits or function module exits.

Method 3: (Identifying BADI)

Business Scenario: You want to warn the user whenever user **SAVE** incoming invoice with zero amount in transaction code **MIRO**.

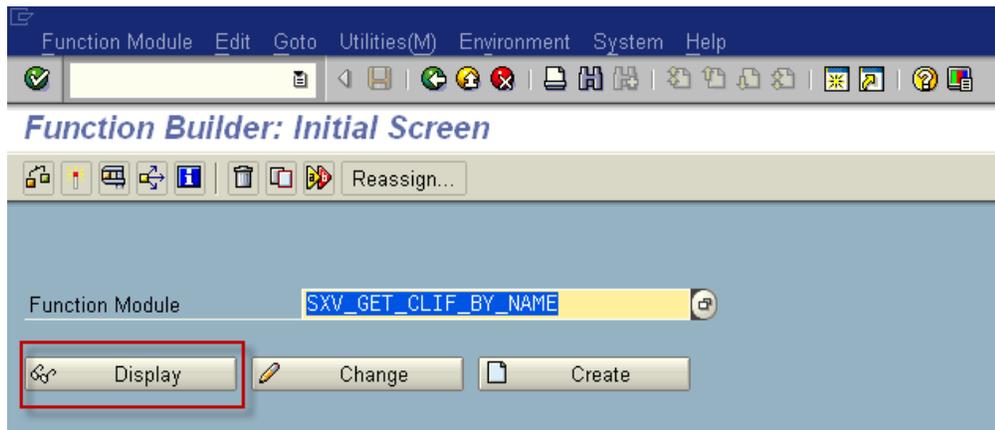
Here in this scenario, we will identify the list of BADI's that are available for **MIRO** transaction code. I will be using de-bugging technique here to identify the BADI.

Overview of Debugging Keys:

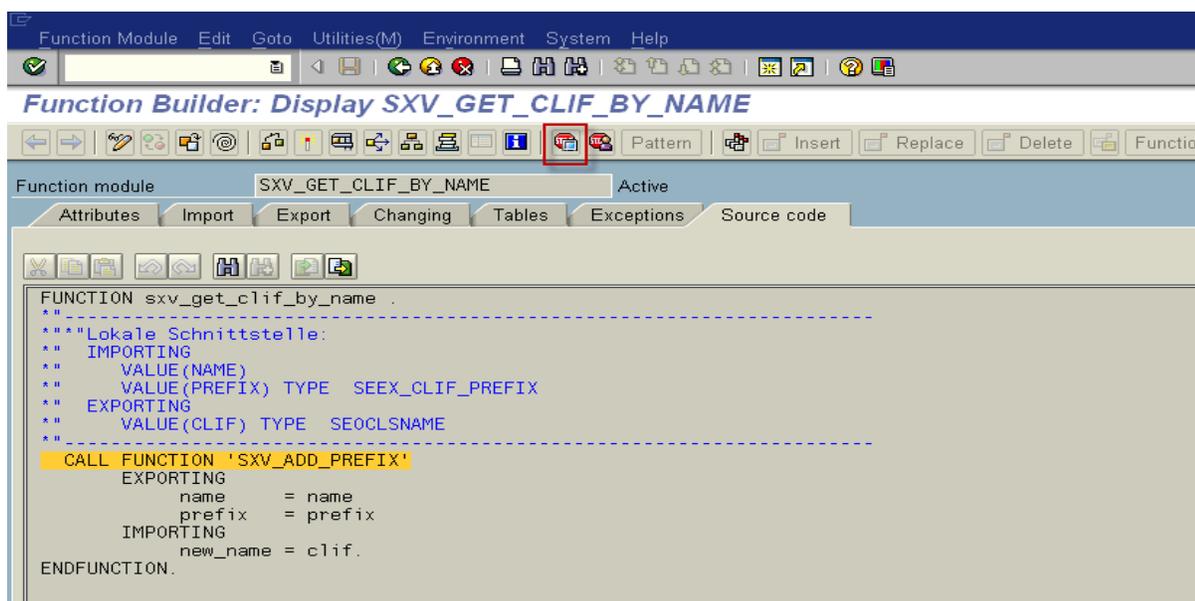
Key	Description	Action
F5	Single step	Execute one line at a time, even if it is nested
F6	Execute	Execute one line at a time, only if it is not nested.
F7	Return	Return to next higher level..
F8	Continue	Keep running till finish or next breakpoint...whichever is comes earlier.

Steps involved in identifying the BADI.

- 1) Go to transaction code **SE37**, and enter the following function module name "**SXV_GET_CLIF_BY_NAME**", press display



- 2) Set a break-point, by selecting first line and clicking stop icon as highlighted below.



3) Run transaction code **MIRO**, program will stop at the break point that is set in step 2.

Double click on the name variable; it will display all the BADI's used in MIRO. Press F8 to find all BADI's.

Example: **MRM_TRANSACT_DEFAULT** (BADI name, Use **SE18** to see the BADI attributes)

The screenshot displays the ABAP Debugger interface. At the top, there is a menu bar with options: Debugging, Edit, Goto, Breakpoints, Settings, Development, System, and Help. Below the menu is a toolbar with various icons for navigation and execution. The main window is titled "ABAP Debugger" and contains several tabs: Fields, Table, Breakpoints, Watchpoints, Calls, Overview, and Settings. The "Fields" tab is active, showing the following information:

- Main Program: SAPLSE XV
- Source code of: LSE XVU36

The source code editor shows the following code:

```

FUNCTION SXV_GET_CLIF_BY_NAME
  *" IMPORTING
  *"   VALUE(NAME)
  *"   VALUE(PREFIX) TYPE SEEX_CLIF_PREFIX
  *" EXPORTING
  *"   VALUE(CLIF) TYPE SEOCLSNAME
  *" -----
  → STOP CALL FUNCTION 'SXV_ADD_PREFIX'
      EXPORTING
          name      = name
          prefix    = prefix
      IMPORTING
          new_name  = clif.
ENDFUNCTION.
  
```

At the bottom of the debugger, there is a "Field names" table with the following content:

Field names	Field contents
name	MRM_TRANSACT_DEFAULT

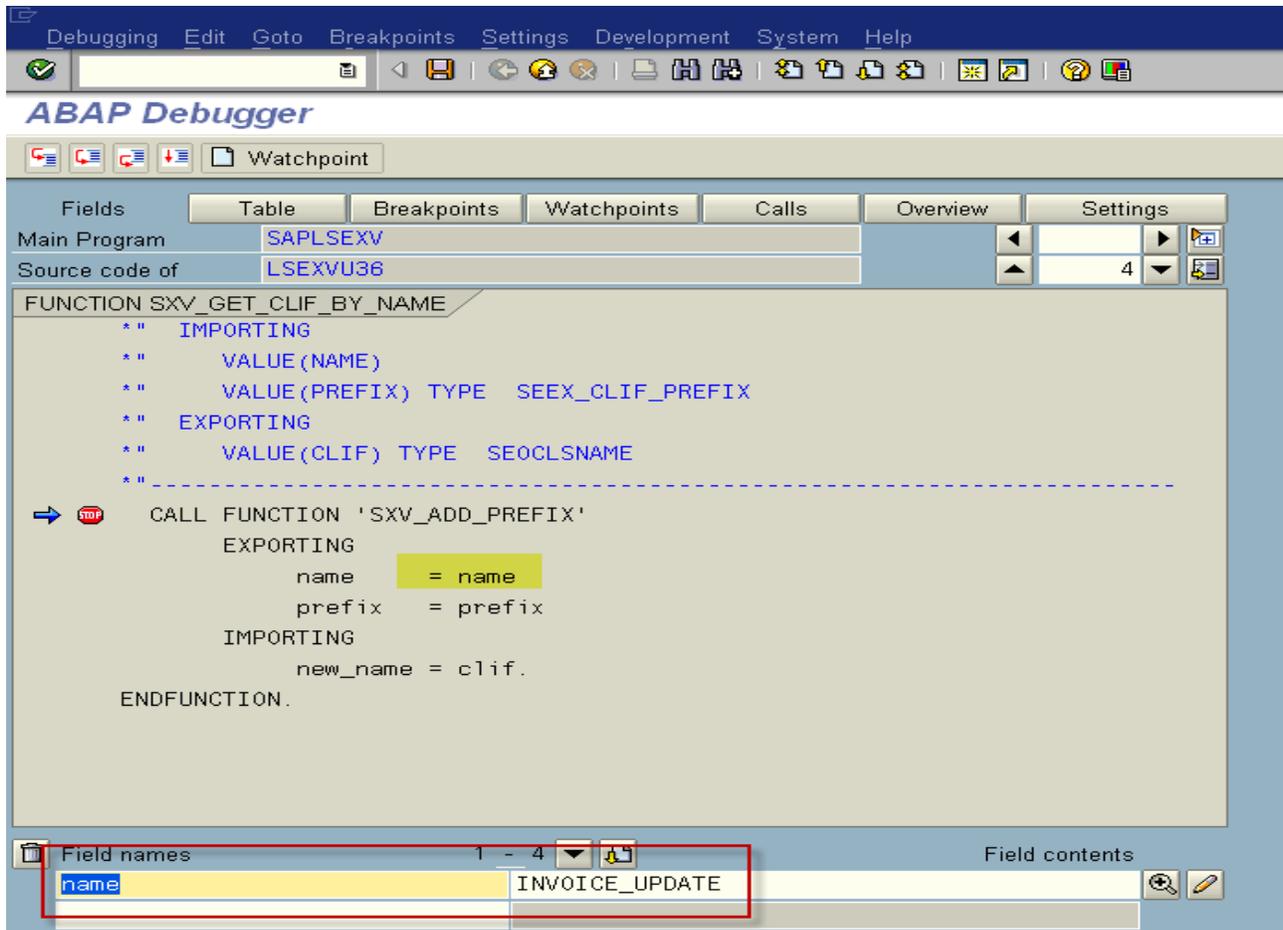
4) As our intention is to find a BADI, when the save button is clicked in **MIRO**.

Enter all the required information and press **SAVE**, as shown

The screenshot shows the SAP MIRO 'Enter Incoming Invoice' screen. The top toolbar contains a 'Save' button (represented by a floppy disk icon) which is highlighted with a red box. The main form area is titled 'Enter Incoming Invoice:' and contains several tabs: 'Basic data', 'Payment', 'Details', 'Tax', 'Contacts', and 'Note'. The 'Basic data' tab is active, showing fields for 'Invoice date' (23.02.2011), 'Posting Date' (23.02.2011), 'Amount' (highlighted with a red box), 'Tax Amount', 'Text', 'Paymt terms' (60 Days net), and 'Baseline Date' (23.02.2011). Below this, there are tabs for 'PO reference', 'G/L account', and 'Material'. The 'PO reference' tab is active, showing '1 Purchase Order/Scheduling Agreement'. At the bottom, there is a table with columns: Item, Amount, Quantity, Or..., Purchase Or..., Item, and PO. The table contains one row with the following data:

Item	Amount	Quantity	Or...	Purchase Or...	Item	PO
1	15,225.00	105,000	PC	45000	10	TC-

5) It will stop you in debugging mode; by pressing F8 it will show all the available BADI's that are tied to MIRO save option.



Make use of the relevant BADI that suits your requirement. Here in this Scenario “INVOICE_UPDATE” BADI will fulfill the requirement.

Related Content

www.help.sap.com

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.