

# Crack Hard to Solve Issues with Debugging for Non-Programmers

**Rohana Gunawardena**  
Director – SAP Practice  
Exium Inc.

SAPinsider  
Las Vegas

---

2023

**SAP**insider



## In This Session

---

- What is a debugger?
- How can it help you?
- Simple scenario for use
- How to use without a programmer

# Rohana Gunawardena

---

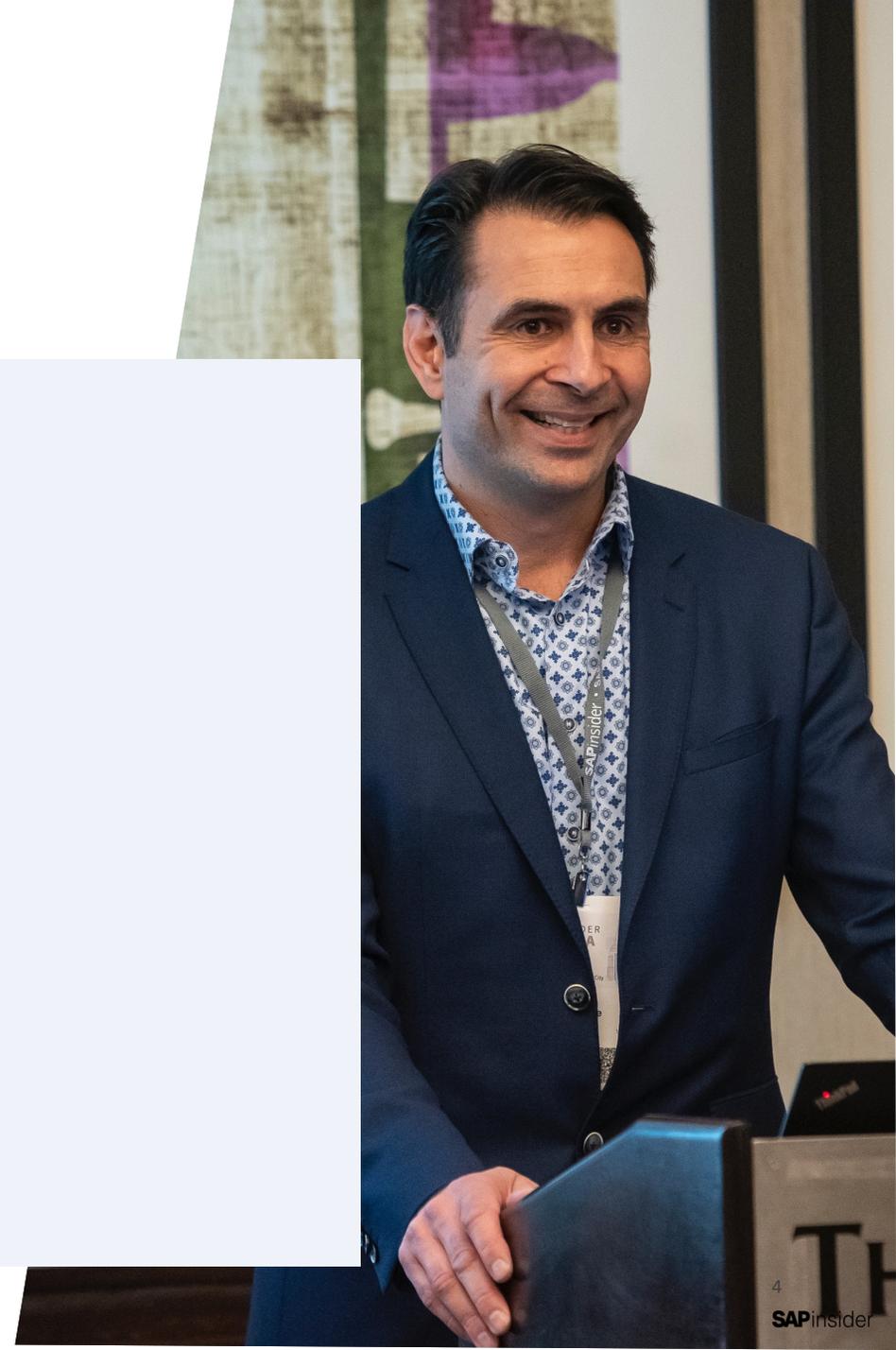
- Rohana Gunawardena heads the SAP practice division at Exium Inc.
- Rohana has been working with SAP since 1992
- Rohana is a Fellow of the Institute of Chartered Accountants in England & Wales.



# What We'll Cover

---

- Introduction
- Step-by-Step Guide
- Debugger Overview
- Where is the Config?
- Breakpoints
- Wrap-up





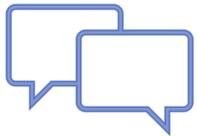
# Introduction

---

## Why This Session?

---

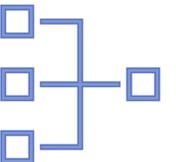
- Have you ever come across a mysterious SAP error message that you spent many hours setting up test scenarios and replicating only to find out the message and the actual cause of the error were two different things?
- I wanted to share with you a tool that will get you to the root cause of the error message quickly and reveal the configuration table which causes the error.



## Why Debug?

---

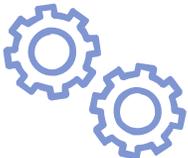
- When trouble shooting configuration often the SAP system does not behave the way documentation says it should
- Error messages say something but its inconsistent with your expectation
- You can spend several days looking at issues like this, researching SAP Notes, Googling, asking colleagues, etc. and not get anywhere
- SAP behavior follows the code not the documentation
  - In the end you need to look at the code to see what SAP is really doing



## What is a Debugger?

---

- It is a tool that lets you see the SAP code as the transaction is running
- See exactly how SAP processes the code
- See exactly which tables SAP looks at for configuration



# Step-by-Step Guide

---



# Scenario

- Let's look at an error message in SAP

The screenshot shows the SAP 'Enter G/L account document: Add G/L account item' screen. The G/L Account is 800200 (Sales revenues) and the Company Code is 1000 (Best Run). The current item is 'Item 2 / Debit entry / 40' with an amount of 1.000,00 EUR and tax code 01. The error message at the bottom states: 'Only output tax is allowed for account 800200 1000, 01 is not al...'. The SAP logo and system information (EH8 (2) 800) are visible in the bottom right corner.

Document Edit Goto Extras Settings Environment System Help

Enter G/L account document: Add G/L account item

More data Account Model G/L item fast entry Taxes

G/L Account 800200 Sales revenues (for eg.s not forwarded to CO-PA)  
Company Code 1000 Best Run (do not change!)

Item 2 / Debit entry / 40

Amount 1.000,00 EUR  
Tax Code 01

W/o Cash Dscnt  
Cost Center Order  
WBS element Profit. segment  
Business Area  
Profit Center Sales Order  
Assignment Asst retirement  
Text Long Texts

More

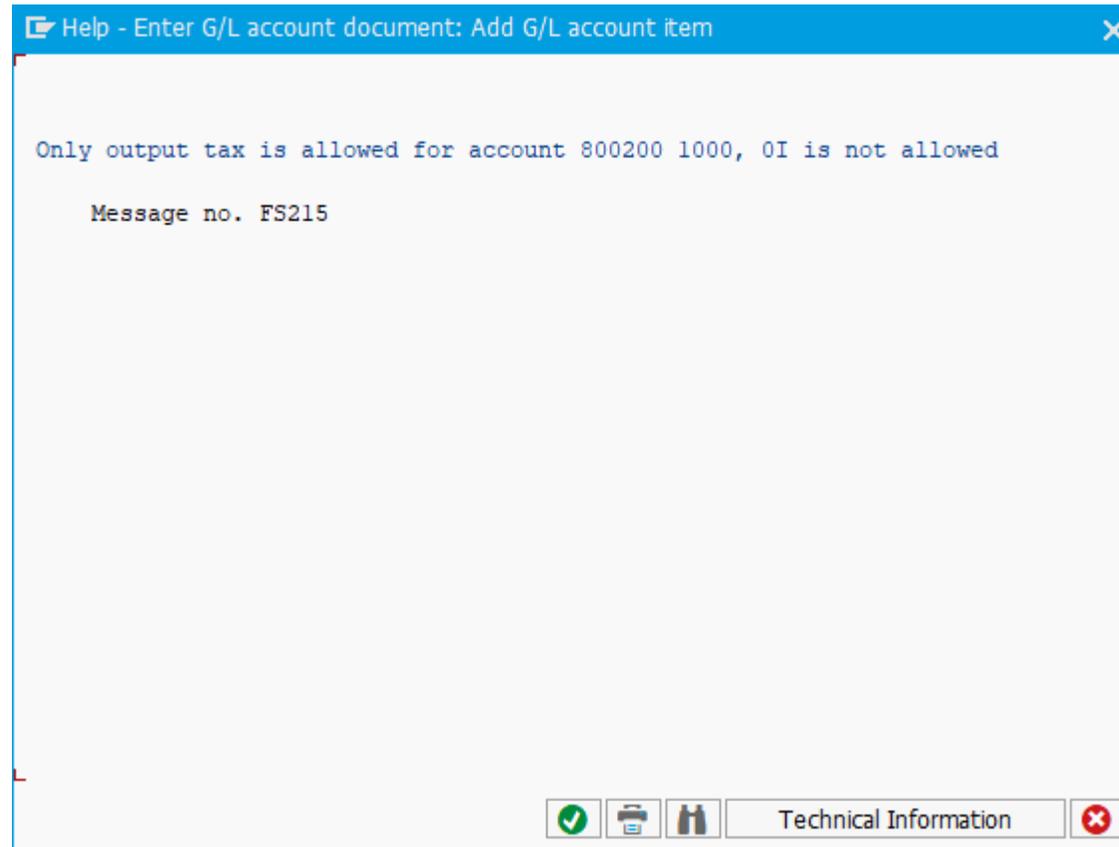
Next Line Item  
PstKy Account SGL Ind TType New co.code

! Only output tax is allowed for account 800200 1000, 01 is not al... SAP >> | EH8 (2) 800 | ehp8ides | OVR

# Step 1

---

- Double click on the error message to get the message code FS215
  - Message ID & message number



## Step 2

- Enter /H in the OK code field to start debugging mode

The screenshot shows the SAP G/L account document entry interface. The menu bar includes Document, Edit, Goto, Extras, Settings, Environment, System, and Help. The title bar shows a green checkmark, a dropdown menu with "/H" selected, and various navigation icons. The main header reads "Enter G/L account document: Add G/L account item". Below this, there are icons for "More data", "Account Model", "G/L item fast entry", and "Taxes".

The main data entry area contains the following fields:

- G/L Account: 800200 Sales revenues (for eg.s not forwarded to CO-PA)
- Company Code: 1000 Best Run (do not change!)

The "Item 2 / Debit entry / 40" section includes:

- Amount: 1.000,00 EUR
- Tax Code: 0I
- W/o Cash Dscnt
- Cost Center: [empty]
- Order: [empty]
- WBS element: [empty]
- Profit. segment: [empty]
- Business Area: [empty]
- Profit Center: [empty]
- Sales Order: [empty]
- Assignment: [empty]
- Asst retirement:
- Text: [empty]

The "Next Line Item" section includes:

- PstKy: [empty]
- Account: [empty]
- SGL Ind: [empty]
- TType: [empty]
- New co.code: [empty]

The status bar at the bottom displays a warning icon and the message: "Only output tax is allowed for account 800200 1000, 0I is not al...". The SAP logo and navigation icons are also visible.

## Step 3

- Will see debugging mode is switched on
- Proceed with transaction, e.g. press <enter> or <save>

The screenshot shows the SAP S/4HANA 'Enter G/L account document' transaction. The interface includes a menu bar (Document, Edit, Goto, Extras, Settings, Environment, System, Help) and a toolbar with various icons. The main area displays the following information:

- Transaction: Enter G/L account document: Add G/L account item
- Navigation: More data, Account Model, G/L item fast entry, Taxes
- Account Information:
  - G/L Account: 800200 Sales revenues (for eg.s not forwarded to CO-PA)
  - Company Code: 1000 Best Run (do not change!)
- Item 2 / Debit entry / 40
  - Amount: 1.000,00 EUR (highlighted with a red box)
  - Tax Code: 0I
  - Cost Center: [empty]
  - WBS element: [empty]
  - Profit Center: [empty]
  - Assignment: [empty]
  - Text: [empty]
  - Order: [empty]
  - Profit. segment: [empty]
  - Business Area: [empty]
  - Sales Order: [empty]
  - Asst retirement:
  - W/o Cash Dscnt:
  - Long Texts: [empty]
- Next Line Item
  - PstKy: [empty]
  - Account: [empty]
  - SGL Ind: [empty]
  - TType: [empty]
  - New co.code: [empty]

At the bottom of the screen, a status bar indicates 'Debugging switched on' (highlighted with a yellow box), the SAP logo, and system information: 'EH8 (2) 800 | ehp8ides | OVR | [system icons]'.

# Step 4

- ABAP Debugger will open up
  - I have expanded my screen yours may look different

The screenshot displays the ABAP Debugger (Exclusive) for session ehp8ides\_EH8\_02. The main window shows the source code of the ABAP program, with the following visible lines:

```
10 MODULE valuta_vorschlag.  
11 MODULE gross_income_vorschlag.  
12 MODULE ref_belegzeile.  
13 MODULE ref_fusszeile.  
14 MODULE betraege_rettten.  
15 MODULE anlagen_abgang.  
16 MODULE fill_lpn. "Note 1313752  
17 MODULE type_ahead.  
18 *  
19 * und ab in den Kontierungsblock  
20 *  
21 MODULE send_pbo.  
22 CALL SUBSCREEN block INCLUDING progn dynnr.  
23 *  
24 PROCESS AFTER INPUT.  
25 *  
26 MODULE exit AT EXIT-COMMAND.  
27 MODULE pai_anfang.  
28 *  
29 CHAIN.  
30 FIELD bseg-wrbtr.  
31 FIELD bseg-dmbtr MODULE stern_betrg ON *-INPUT.  
32 FIELD bseg-wrbtr MODULE stern_betrg ON *-INPUT.  
33 FIELD bseg-mwskz MODULE mwskz_pruefen.  
34 FIELD bseg-xskil VALUES (' ', 'X').  
35 MODULE betrag_pruefen.  
36 ENDCHAIN.  
37 *  
38 FIELD bkpfxmwst MODULE xmwst_sako.  
39 FIELD bseg-valut MODULE bseg-valut.  
40 FIELD bseg-zuonr.  
41 FIELD bseg-sgtxt MODULE segmenttext_setzen ON INPUT.  
42 FIELD bseg-txjcd.  
43 *  
44 CHAIN.  
45 FIELD bseg-menge.  
46 FIELD bseg-meins.  
47 MODULE menge_meins.  
48 ENDCHAIN.  
49 * FOR
```

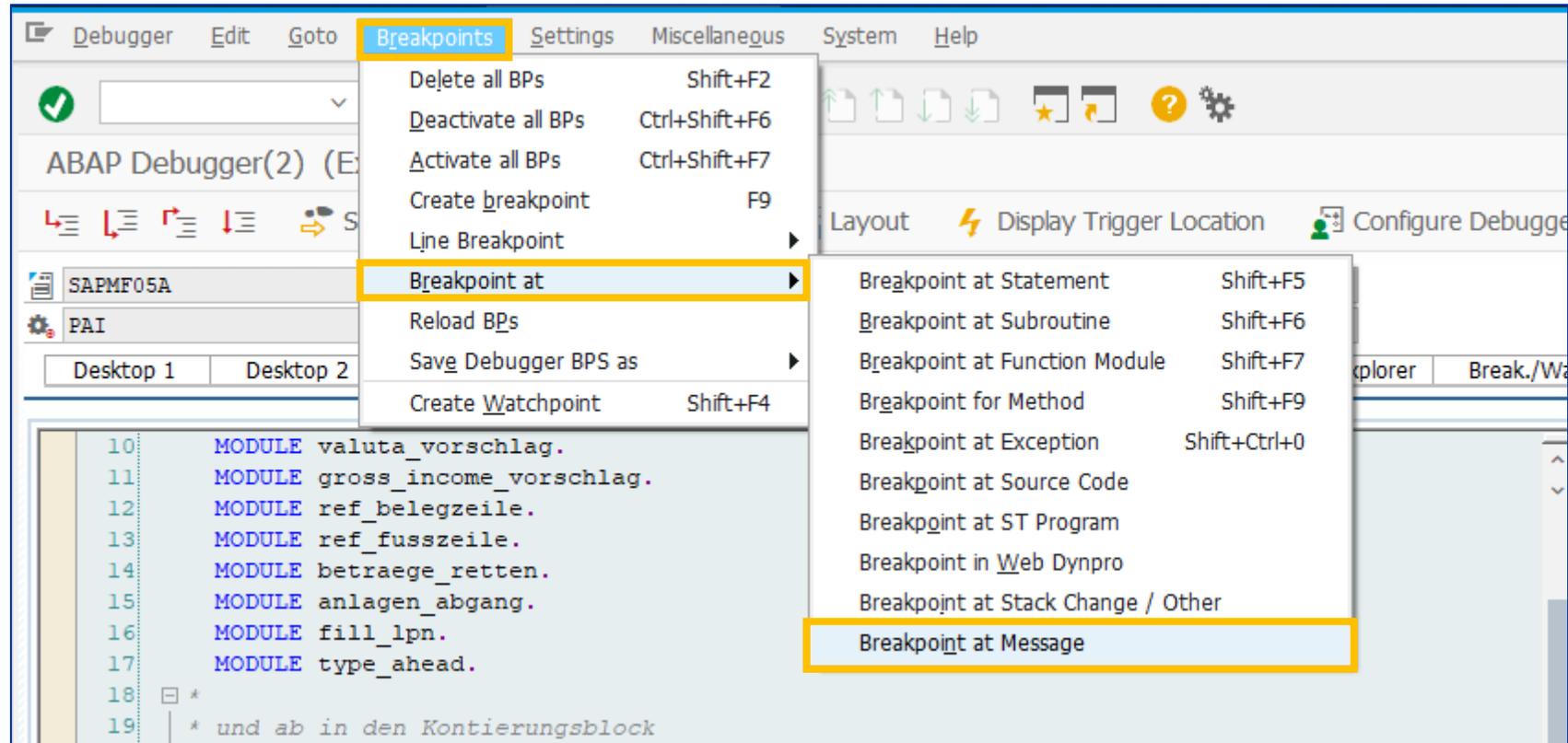
The right-hand side of the debugger shows the 'ABAP and Screen Stack' table:

St...	Sta...	S...	Event Type	Event	Program	Na...	Include	Line
2		PAI	SCREEN	0300	SAPMF05A			30
1		TRANSACTION		FB01(FB01)				0

Below the stack is the 'Variables' window, which is currently empty. The status bar at the bottom indicates the current position: 'ABAP | Ln 30 Col 1 | NUM'.

## Step 5

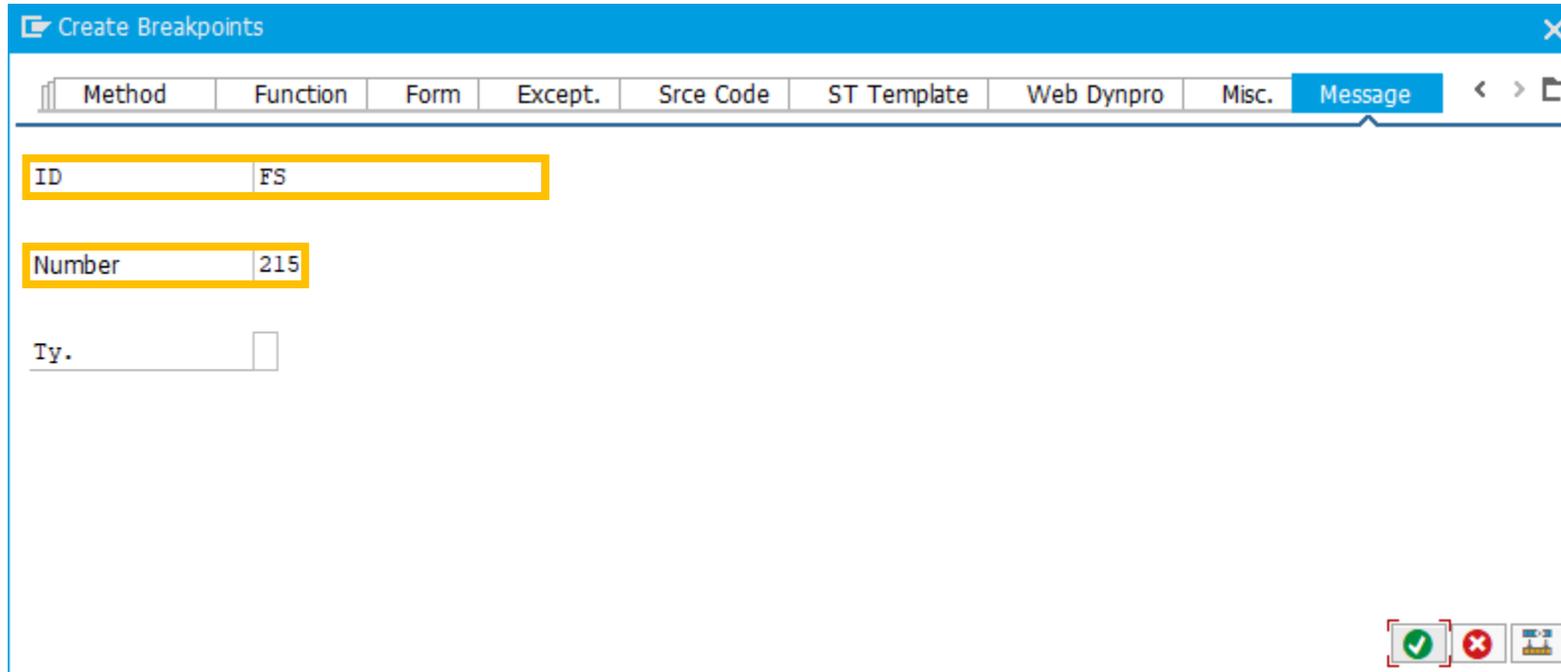
- Select breakpoint at message option



## Step 6

---

- Enter the message code you identified earlier
  - FS215 last three digits are always the number, the ID can be 2 or more characters
    - e.g. 107\_CUST002 is also a valid error message ID & number



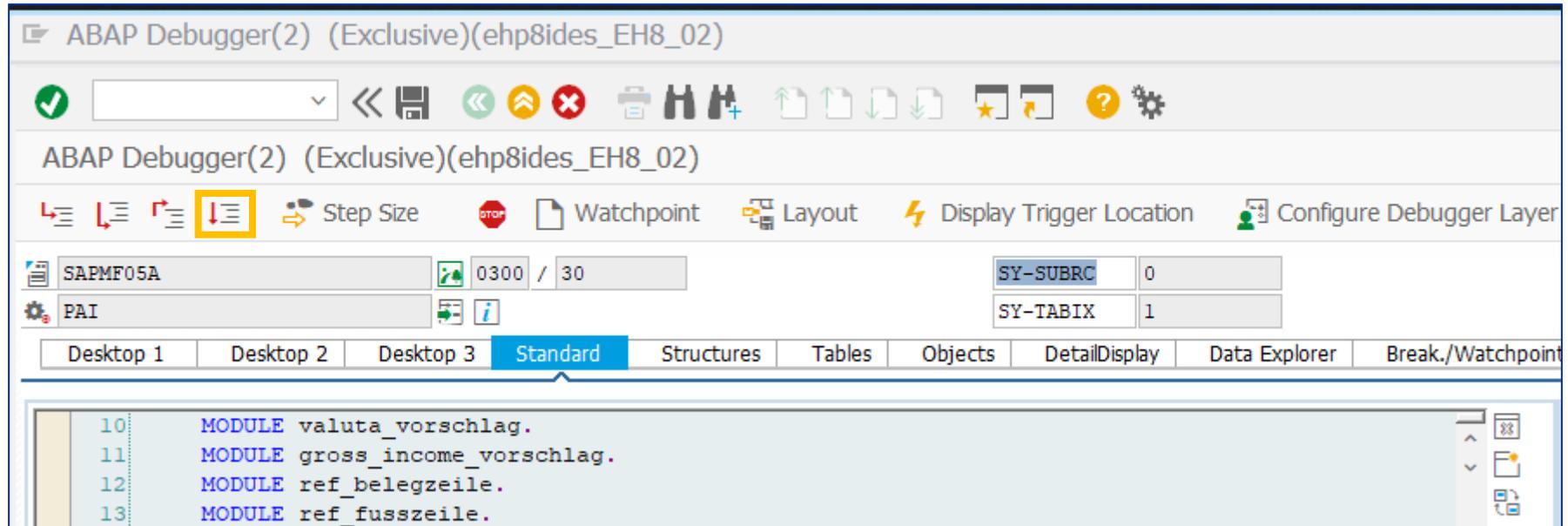
The screenshot shows the 'Create Breakpoints' dialog box with the 'Message' tab selected. The 'ID' field is highlighted in yellow and contains the text 'FS'. The 'Number' field is also highlighted in yellow and contains the text '215'. The 'Ty.' field is empty. At the bottom right, there are three icons: a green checkmark, a red X, and a blue document icon.

Method	Function	Form	Except.	Srcce Code	ST Template	Web Dynpro	Misc.	Message
ID	FS							
Number	215							
Ty.								

## Step 7

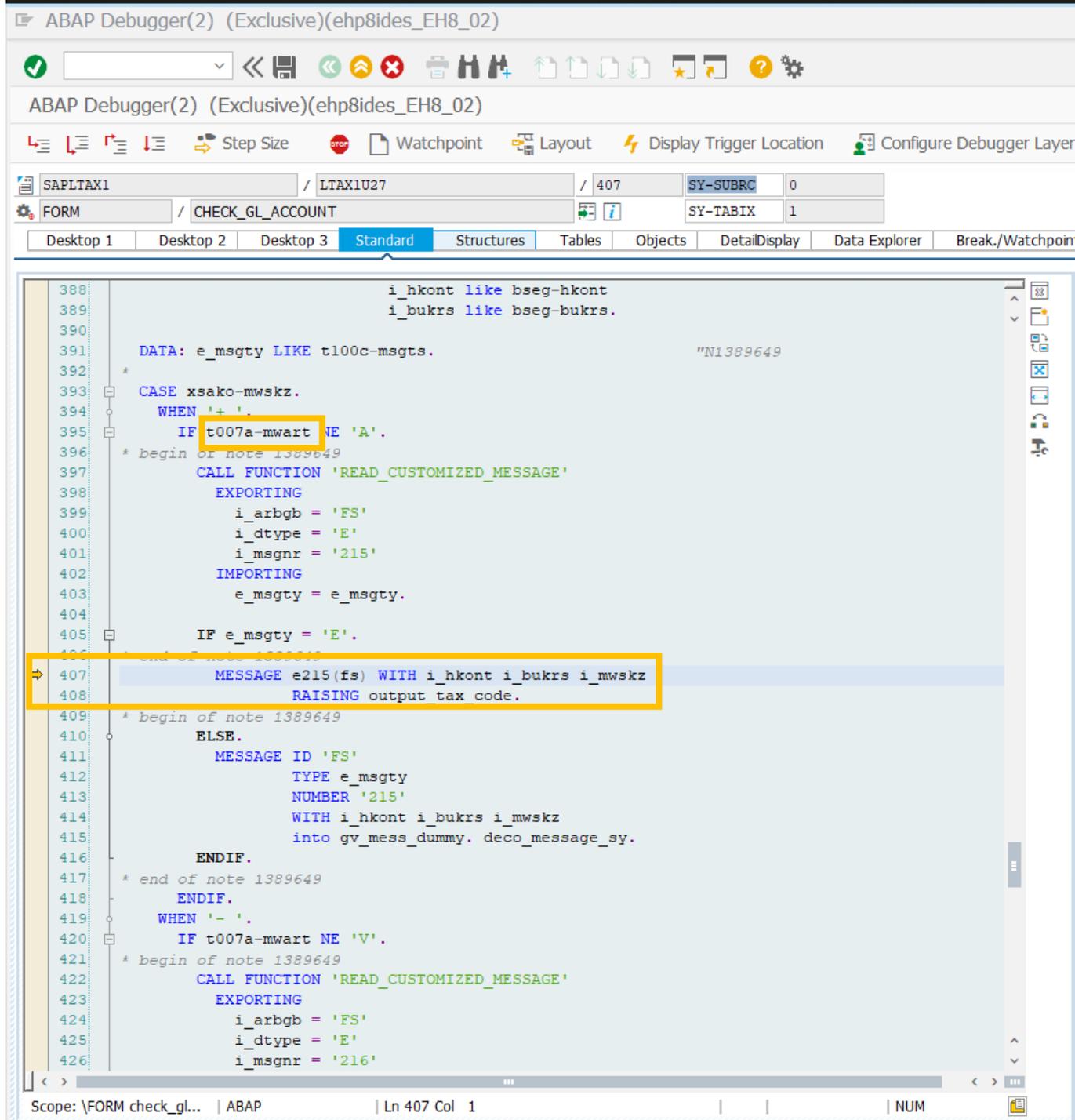
---

- Press continue (F8) in the debugger tool bar



## Step 8

- Debugger will proceed to where the error message is issued in the code
- Here you can see the key configuration table is T007A
- Depending on your situation the key table maybe in a slightly different position or you may see code which is not meaningful to you
  - You may need technical help



The screenshot shows the ABAP Debugger interface. The main window displays the source code of a program, with the following code snippet highlighted in blue:

```
388          i_hkont like bseg-hkont
389          i_bukrs like bseg-bukrs.
390
391          DATA: e_msgty LIKE t100c-msgts.                "N1389649
392          *
393          CASE xsako-mwskz.
394              WHEN '+' .
395                  IF t007a-mwart NE 'A'.
396                      * begin of note 1389649
397                      CALL FUNCTION 'READ_CUSTOMIZED_MESSAGE'
398                          EXPORTING
399                              i_arbgb = 'FS'
400                              i_dtype = 'E'
401                              i_msgnr = '215'
402                          IMPORTING
403                              e_msgty = e_msgty.
404
405                      IF e_msgty = 'E'.
406                          * begin of note 1389649
407                          MESSAGE e215(fs) WITH i_hkont i_bukrs i_mwskz
408                              RAISING output tax code.
409                      * end of note 1389649
410                      ELSE.
411                          MESSAGE ID 'FS'
412                              TYPE e_msgty
413                              NUMBER '215'
414                              WITH i_hkont i_bukrs i_mwskz
415                              INTO gv_mess_dummy. deco_message_sy.
416                      ENDIF.
417                      * end of note 1389649
418                      ENDIF.
419              WHEN '-' .
420                  IF t007a-mwart NE 'V'.
421                      * begin of note 1389649
422                      CALL FUNCTION 'READ_CUSTOMIZED_MESSAGE'
423                          EXPORTING
424                              i_arbgb = 'FS'
425                              i_dtype = 'E'
426                              i_msgnr = '216'
```

The debugger interface includes a toolbar at the top with various icons for navigation and execution. The status bar at the bottom indicates the current scope as `\FORM check_gl...` and the current line and column as `Ln 407 Col 1`.



# Debugger Overview

---

# Debugger Windows

The screenshot displays the SAP ABAP Debugger interface for program SAPMF05A. The main window shows the source code with line 30 highlighted. The 'ABAP and Screen Stack' window shows the current call stack with the PAI SCREEN event at line 30. The 'Variables' window shows the current variable values.

**1. Code**

```
10  MODULE valuta_vorschlag.  
11  MODULE gross_income_vorschlag.  
12  MODULE ref_belegzeile.  
13  MODULE ref_fusszeile.  
14  MODULE betraege_rettten.  
15  MODULE anlagen_abgang.  
16  MODULE fill_lpn.                "Note 1313752  
17  MODULE type_ahead.  
18  *  
19  * und ab in den Kontierungsblock  
20  *  
21  MODULE send_pbo.  
22  CALL SUBSCREEN block INCLUDING progn dynnr.  
23  
24  PROCESS AFTER INPUT.  
25  *  
26  MODULE exit AT EXIT-COMMAND.  
27  MODULE pai_anfang.  
28  *  
29  CHAIN.  
30  FIELD bseg-wrbtr.  
31  FIELD bseg-dmbr MODULE stern_betrg ON *-INPUT.  
32  FIELD bseg-wrbtr MODULE stern_betrg ON *-INPUT.  
33  FIELD bseg-mwskz MODULE mwskz_pruefen.  
34  FIELD bseg-xskrl VALUES (' ', 'X').  
35  MODULE betrag_pruefen.  
36  ENDCHAIN.  
37  *  
38  FIELD bkpfxm-wst MODULE xmwst_sako.  
39  FIELD bseg-valut MODULE bseg-valut.  
40  FIELD bseg-zuonr.  
41  FIELD bseg-sgtxt MODULE segmenttext_setzen ON INPUT.  
42  FIELD bseg-txjcd.  
43  *  
44  CHAIN.  
45  FIELD bseg-menge.  
46  FIELD bseg-meins.  
47  MODULE menge_meins.  
48  ENDCHAIN.  
49  * POP
```

**2. Call Stack**

St...	Sta...	S.. Event Type	Event	Program	Na...	Include	Line
	2	PAI SCREEN	0300	SAPMF05A			30
	1	TRANSACTION	FB01(FB01)				0

**3. Variables**

S...	Variable	V... Val.	C... Hexadecimal Value	Technical Type	Absolute Type
------	----------	-----------	------------------------	----------------	---------------

# Code Window

- Shows processing in the code
- In the header area shows main program, include and section of code
  - Form, function, method, etc.
- Even if you don't read code seeing the reference to table T007A will help you investigate the issue

The screenshot shows the ABAP Debugger interface. The title bar reads "ABAP Debugger(1) (Exclusive)(ehp8ides\_EH8\_02)". The main window displays the source code of a program. The header area shows the program name "SAPLTAX1" and the function "FORM CHECK\_GL\_ACCOUNT". The code is as follows:

```
388      i_hkont like bseg-hkont
389      i_bukrs like bseg-bukrs.
390
391      DATA: e_msgty LIKE t100c-msgts.                "N1389649
392
393      *
394      CASE xsako-mwskz.
395          WHEN '+' .
396              IF t007a-mwart NE 'A'.
397                  * begin of note 1389649
398                  CALL FUNCTION 'READ_CUSTOMIZED_MESSAGE'
399                      EXPORTING
400                          i_arbgb = 'FS'
401                          i_dtype = 'E'
402                          i_msgnr = '215'
403                  IMPORTING
404                      e_msgty = e_msgty.
405
406                  IF e_msgty = 'E'.
407                      * end of note 1389649
408                      MESSAGE e215(fs) WITH i_hkont i_bukrs i_mwskz
409                          RAISING output_tax_code.
410                      * begin of note 1389649
411                      ELSE.
412                          MESSAGE ID 'FS'
413                              TYPE e_msgty
414                              NUMBER '215'
415                              WITH i_hkont i_bukrs i_mwskz
416                              INTO gv_mess_dummy. deco_message_sy.
417                      ENDIF.
418                      * end of note 1389649
419                  ENDIF.
420              WHEN '-' .
421                  IF t007a-mwart NE 'V'.
422                      * begin of note 1389649
423                      CALL FUNCTION 'READ_CUSTOMIZED_MESSAGE'
424                          EXPORTING
425                              i_arbgb = 'FS'
426                              i_dtype = 'E'
427                              i_msgnr = '216'
```

The status bar at the bottom indicates "ABAP | Ln 407 Col 1 | NUM".

# Call Stack

- Shows the sequence of transaction codes, programs, functions, forms, etc. followed to get to the current code
- For most functional people will not make sense
- OK to ignore this screen

The screenshot displays the ABAP Debugger interface. The top toolbar includes icons for Step Size, Watchpoint, Layout, Display Trigger Location, and Configure Debugger Layer. The main window is divided into two panes. The left pane shows the source code for program SAPLTAX1, with the current execution point at line 407. The right pane, titled 'ABAP and Screen Stack', shows a list of call stack entries.

St...	Sta...	S..	Event Type	Event	Program
7		FORM	CHECK_GL_ACCOUNT	SAPLTAX1	SAPLTAX1
6		FUNCTION	FI_TAX_INDICATOR_CH...	SAPLTAX1	SAPLTAX1
5		FORM	PAI_MWSKZ_PRUEFEN	SAPMF05A	SAPMF05A
4		MODULE (PAI)	MWSKZ_PRUEFEN	SAPMF05A	SAPMF05A
3		PAI MODULE	MWSKZ_PRUEFEN	SAPMF05A	SAPMF05A
2		PAI SCREEN	0300	SAPMF05A	SAPMF05A
1		TRANSACTION	FB01(FB01)		

# Variables

- Allows you to see the value of variables in the code, e.g. T007A-MWART = V
- Useful to get additional information
- Double click on variable in code window to make it appear in the variables window
  - Also can manually type in a variable name

The screenshot displays the ABAP Debugger interface. The top bar shows the debugger title 'ABAP Debugger(1) (Exclusive)(ehp8ides\_EH8\_02)' and various control buttons like 'Step Size', 'Watchpoint', 'Layout', 'Display Trigger Location', and 'Configure Debugger Layer'. Below this, the current program and transaction information is shown: 'SAPLTAX1 / LTAXIU27 / 407' and 'FORM / CHECK\_GL\_ACCOUNT / SY-SUBRC 0 / SY-TABIX 1'. The main area is divided into 'Desktop 1', 'Desktop 2', and 'Desktop 3', with 'Desktop 3' selected. The code window shows the following ABAP code:

```
388      i_hkont like
389      i_bukrs like
390
391      DATA: e_msgty LIKE t100c-msgts.
392
393      *
394      CASE xsako-mwskz.
395      WHEN '+' .
396          IF t007a-mwart NE 'A'.
397              * begin of note 1389649
398              CALL FUNCTION 'READ_CUSTOMIZED_ME
399                  EXPORTING
400                      i_arbgb = 'FS'
401                      i_dtype = 'E'
402                      i_msgnr = '215'
403                  IMPORTING
404                      e_msgty = e_msgty.
405
406              IF e_msgty = 'E'.
407                  * end of note 1389649
408                  MESSAGE e215(fs) WITH i_hkont i
409                      RAISING output_tax_code
410                  * begin of note 1389649
411                  ELSE.
412                      MESSAGE ID 'FS'
413                      TYPE e_msgty
414                      NUMBER '215'
415                      WITH i_hkont i_bukrs i
416                      into gv_mess_dummy. dec
417                  ENDIF.
418              * end of note 1389649
419              ENDIF.
420          WHEN '- ' .
421              IF t007a-mwart NE 'V'.
422                  * begin of note 1389649
423                  CALL FUNCTION 'READ_CUSTOMIZED_ME
424                      EXPORTING
425                          i_arbgb = 'FS'
426                          i_dtype = 'E'
427                          i_msgnr = '216'
```

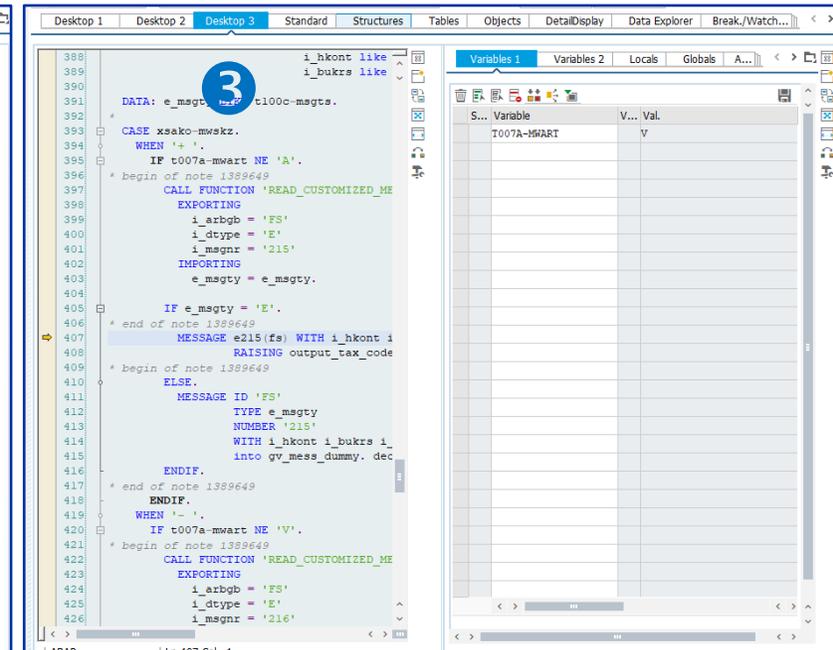
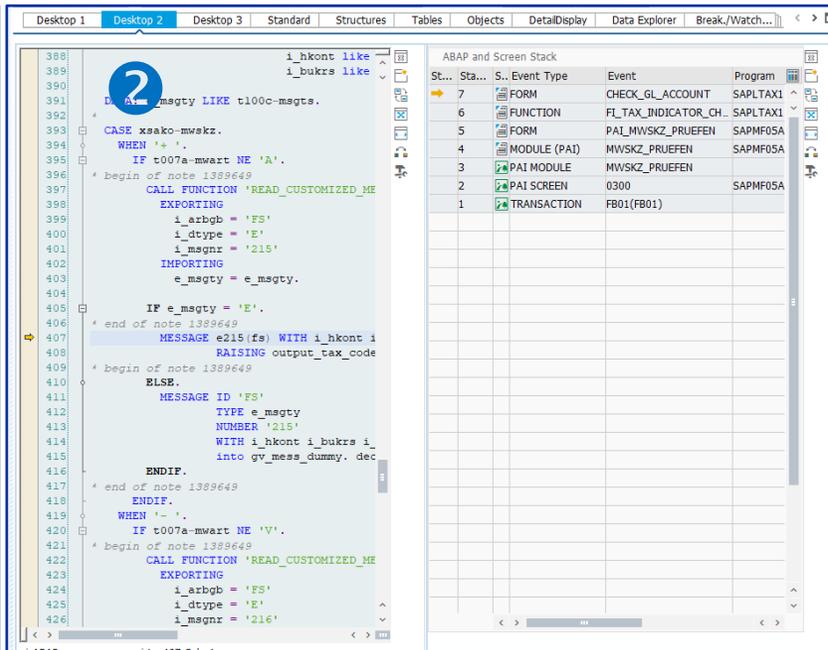
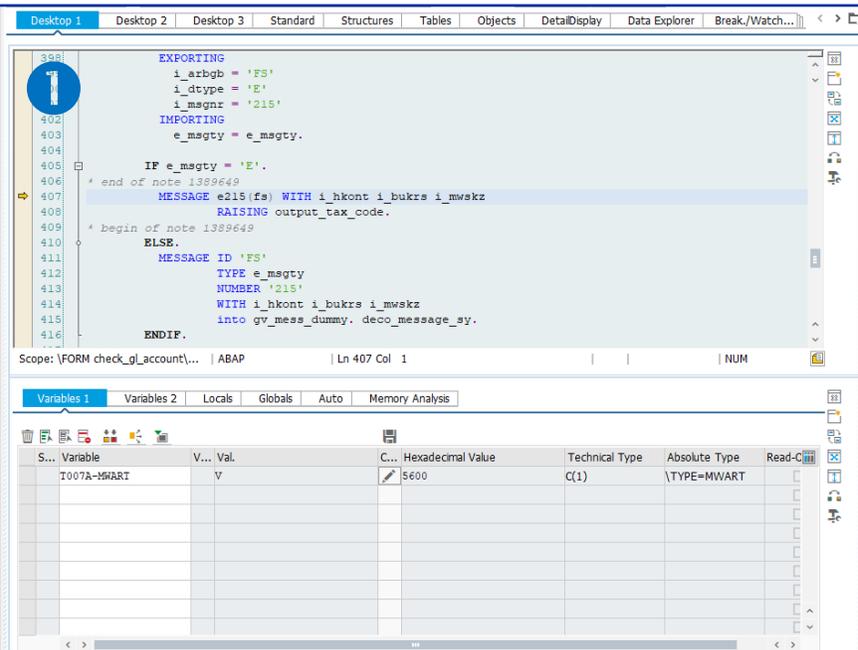
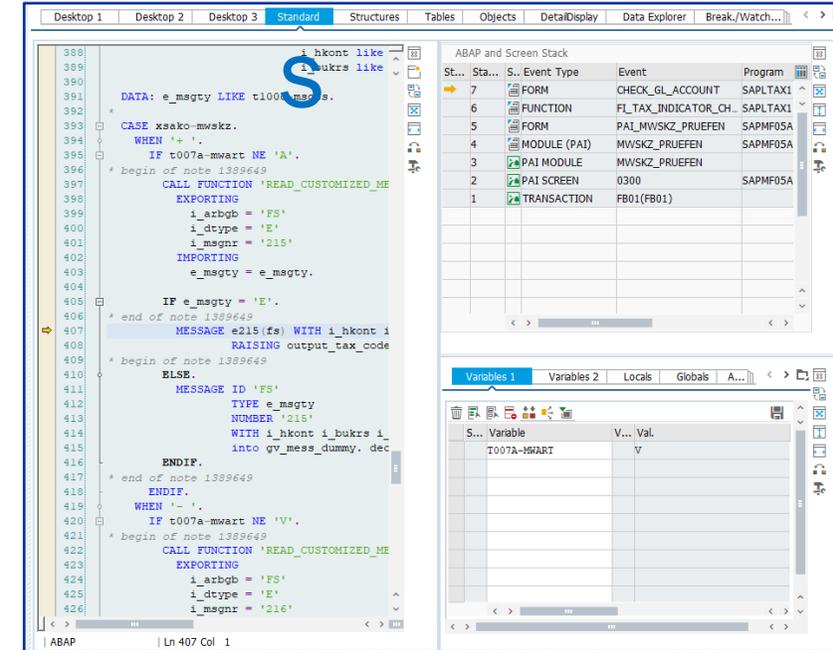
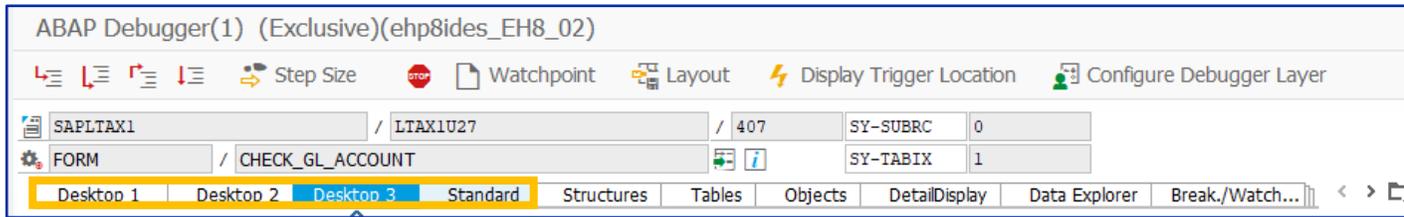
The variables window on the right shows a table with columns 'S...', 'Variable', 'V...', and 'Val.'. The first row contains the variable 'T007A-MWART' with the value 'V'.

S...	Variable	V...	Val.
	T007A-MWART		V

The status bar at the bottom indicates 'ABAP | Ln 395 Col 19'.

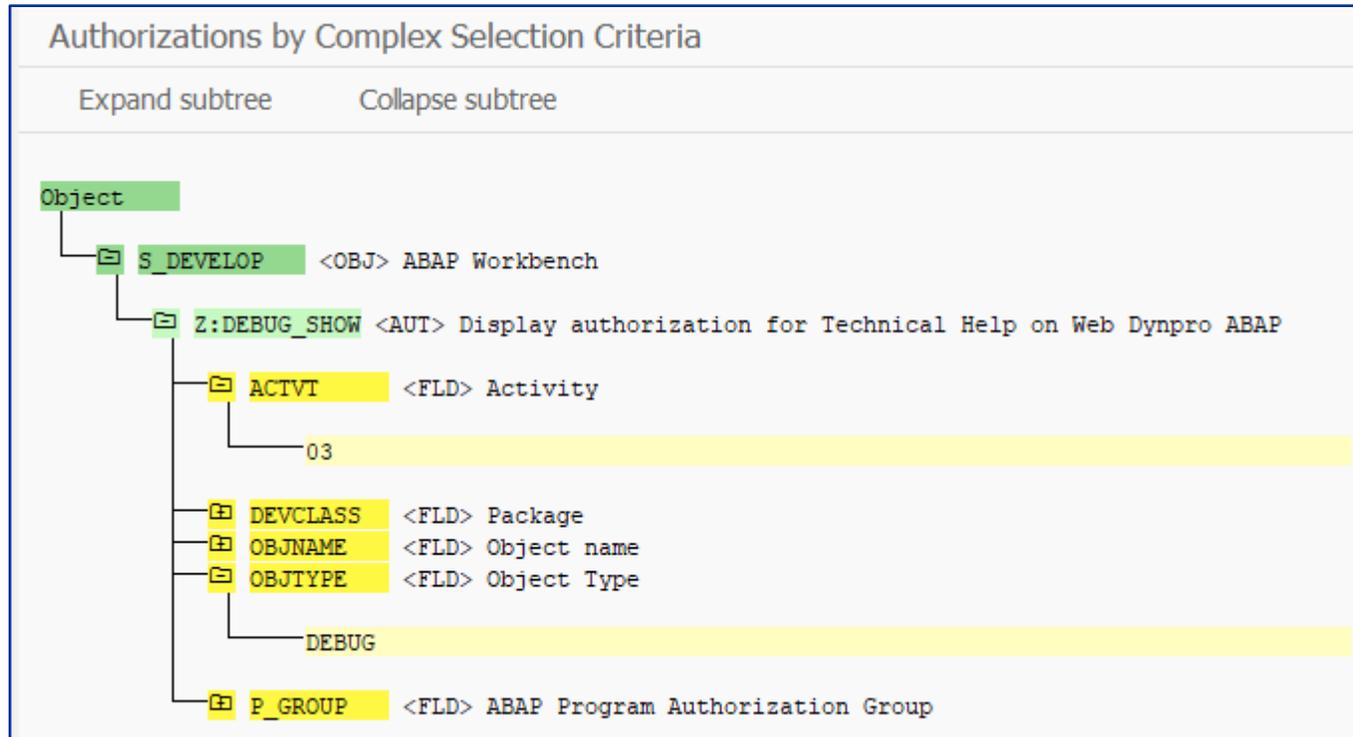
# Desktops

- There are 4 desktops 1, 2, 3 & Standard



# Security

---



- You will need security authorization for debug access
- Policy on access will vary by company and by system (Dev, QA, Prod)



# Features

---

- **There are many debugger features**
  - You don't need to know them all to get value from the debugger
- **In this session want to keep things simple and just focus on what you need to get started**
- **Don't want to scare you off with too much detail**



# Where is the Config?

---





# Display Raw Data

- Once you have a table name like T007A what do you do?
- Can look at raw data using SE16, SE16H, SE16N or SE17

1

Data Browser: Initial Screen

Table Name

3

Data Browser: Table T007A Select Entries 2

Table: T007A  
Displayed Fields: 17 of 28 Fixed Columns: 3 List Width 0250

	Client	Procedure	Tax Code	Deviation error	Tax type	Single-Character Flag	Target tax code	EU Code / Code
<input type="checkbox"/>	800	TAXD	OI		V			
<input type="checkbox"/>	800	TAXRU	OI		V			

2

Data Browser: Table T007A: Selection Screen

Number of Entries

Procedure  to

Tax Code  to

Deviation error  to

Tax type  to

Single-Character Flag  to

4

Table T007A Display

Check Table...

Client

Procedure

Tax Code

Check

Tax type

Einstelliges Kennzeichen

Target tax code

EU Code / Code

Sls taxes

# Find IMG Location

- Enter table name in SM30 and press customizing

Edit Table Views: Initial Screen

Find Maintenance Dialog

Table/View: I007A

Restrict Data Range

- No Restrictions
- Enter conditions
- Variant

Display | Maintain | Transport | **Customizing**

Choose Customizing Project

Project: [ ]

Continue w/o Specifying Project

Edit Table Views: Initial Screen

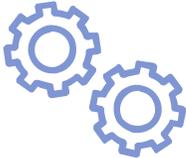
Find Maintenance Dialog

IMG activities overview

Table/View	Implementation guide	IMG activity
	Tax Calculation Brazil	Define FI/MM Tax Codes
	Financial Accounting Global Settings	Define Tax Codes for Sales and Purchases
	Financial Accounting Global Settings (New)	Define Tax Codes for Sales and Purchases
	India: Excise Duty	Define Tax Code for Purchasing Documents

Restrict Data Range

- No R
- Ente
- Varia

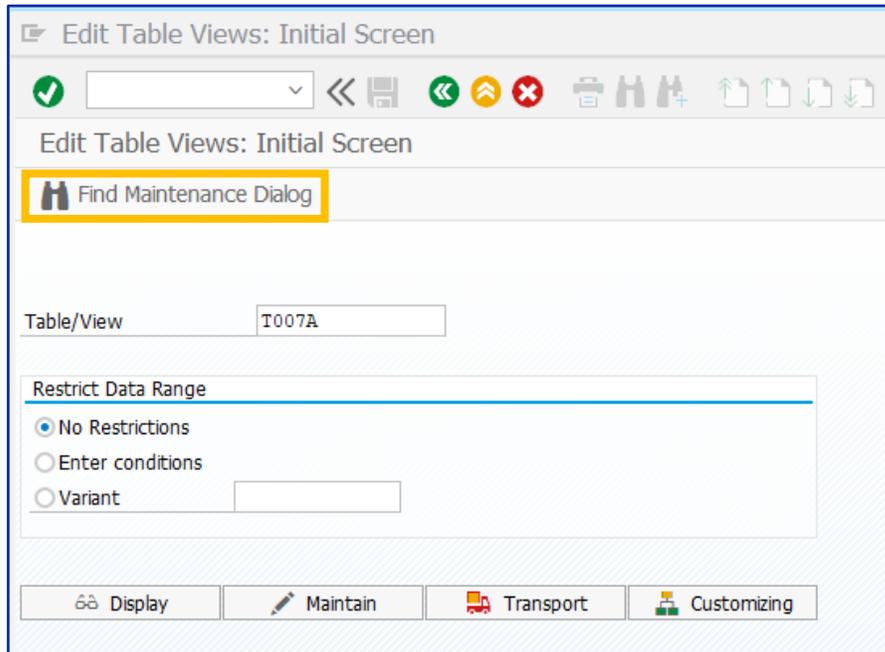




# SM30 View Name

- Note the table name you find in the debugger may not be the view name used in SM30, use the Find Maintenance Dialog button to see the correct name to use.

 The maintenance dialog for T007A is incomplete or not defined



Edit Table Views: Initial Screen

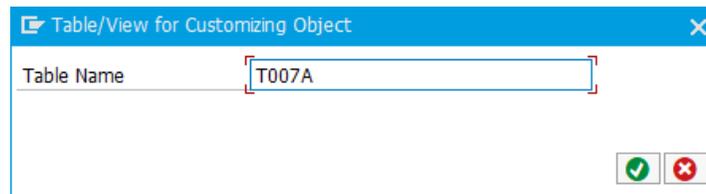
Find Maintenance Dialog

Table/View: T007A

Restrict Data Range

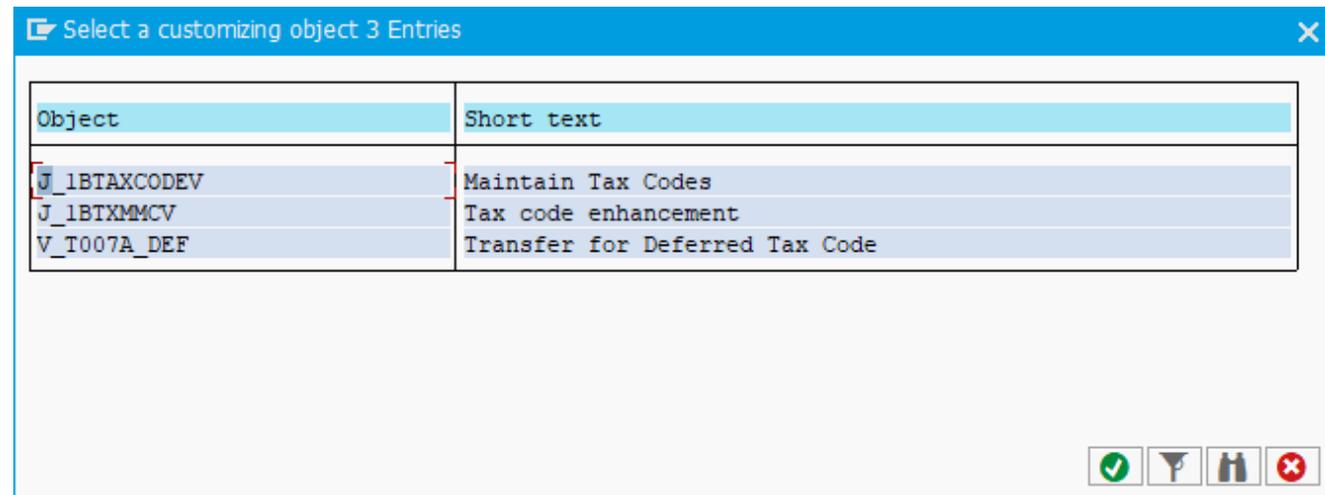
- No Restrictions
- Enter conditions
- Variant

Display Maintain Transport Customizing



Table/View for Customizing Object

Table Name: T007A



Select a customizing object 3 Entries

Object	Short text
J_1BTAXCODEV	Maintain Tax Codes
J_1BTXMMCV	Tax code enhancement
V_T007A_DEF	Transfer for Deferred Tax Code

# Breakpoints

---



## What are Breakpoints?

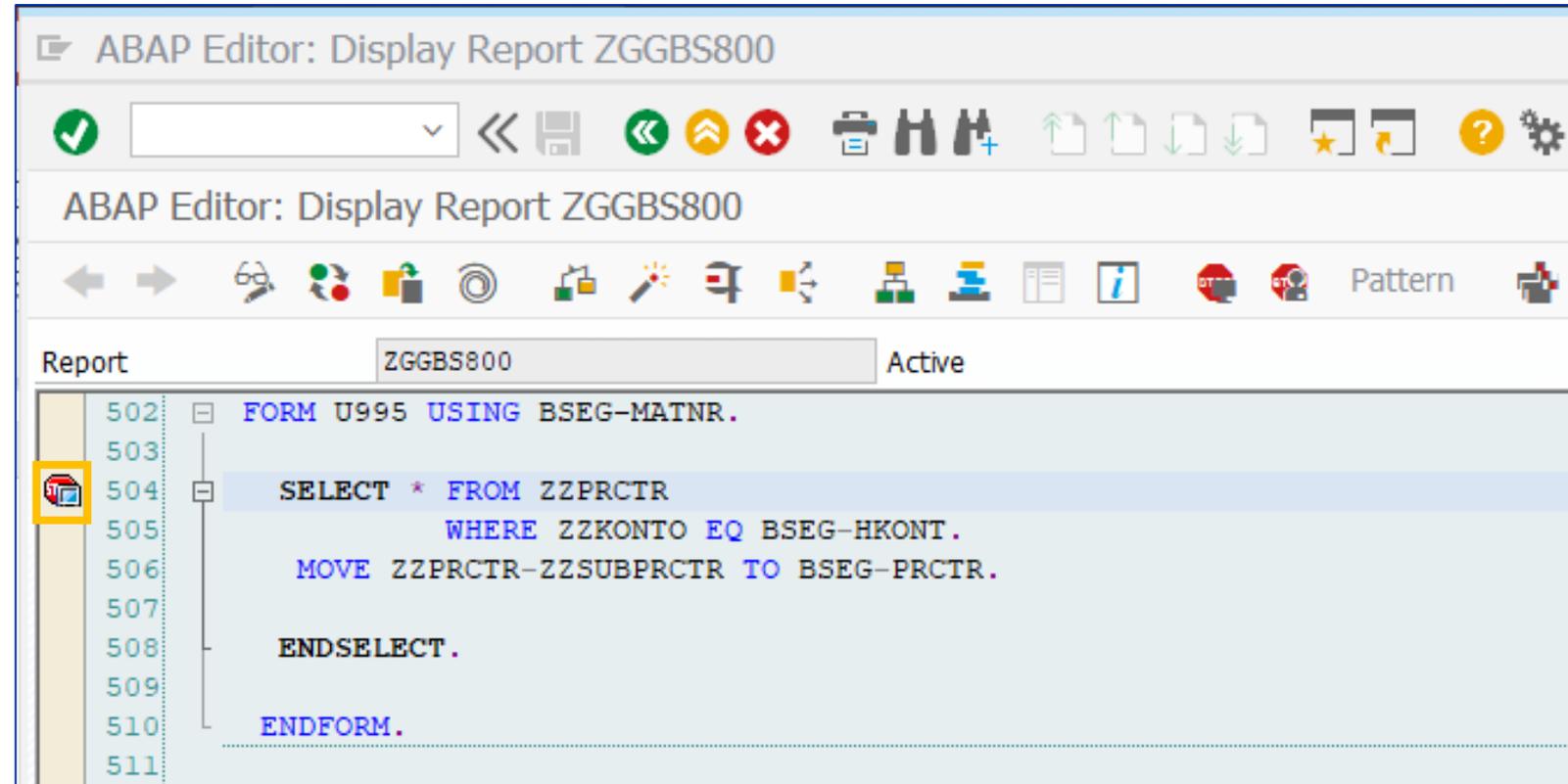
---

- A breakpoint is a specific place in the SAP code you ask the system to stop and open the debugger
- If you are working on a specific enhancement spot or other custom code and you know where it is you can set a breakpoint to look at the code and variable values
- Seeing variable values at run time can be very helpful



## Set Breakpoint

- Transaction SE38
- Click in right hand grey column to set break point at line
- Note must be executable code, not comments, etc.



The screenshot shows the ABAP Editor interface for the report ZGGBS800. The editor displays the following code:

```
502  FORM U995 USING BSEG-MATNR.  
503  
504  SELECT * FROM ZZPRCTR  
505         WHERE ZZKONTO EQ BSEG-HKONT.  
506         MOVE ZZPRCTR-ZZSUBPRCTR TO BSEG-PRCTR.  
507  
508  ENDSELECT.  
509  
510  ENDFORM.  
511
```

A red breakpoint icon is visible in the right-hand grey column next to line 504. The editor title bar reads "ABAP Editor: Display Report ZGGBS800".



Sample code is a custom validation/substitution rule routine

# Run Transaction

- Run the transaction, in this case FB01
- When the transaction gets to the breakpoint there is a message and a new window with the debugger will appear
- Even if it does not open it can be meaningful
  - e.g. now know substitution prerequisite is not met

A screenshot of the SAP software interface. The window title is "Document Edit Goto Extras Settings Environment System Help". The main area shows the transaction "FB01" with the following details:

- Enter G/L account document: Correct G/L account item
- More data Account Model G/L item fast entry Taxes
- G/L Account: 810201 Sales revenues - domestic - finished goods
- Company Code: 1000 Best Run (do not change!)

The "Item 1 / Debit entry / 40" section contains the following fields:

- Amount: 1,000.00 EUR
- Tax Code: 00
- Cost Center: [empty]
- WBS element: [empty]
- Profit Center: PC01
- Assignment: [empty]
- Text: [empty]
- Calculate tax:
- W/o Cash Dscnt:
- Order: [empty]
- Profit. segment: [empty]
- Business Area: [empty]
- Sales Order: [empty]
- Asst retirement:

The "Next Line Item" section contains the following fields:

- PstKy: [empty]
- Account: [empty]
- SGL Ind: [empty]
- TType: [empty]
- New co.code: [empty]

The bottom status bar shows "Session '1' connected to Debugger ..." and the SAP logo.

# Debugger

- Debugger will open at the breakpoint
- You can look at variable values

The screenshot displays the ABAP Debugger interface. The main window shows the source code of a program, with a breakpoint set at line 504. The code includes a form definition and a SQL query. The ABAP and Screen Stack on the right shows the current execution context, including the transaction FB01 and the screen 0300. The Variables window at the bottom right shows the current values of variables, with BSEG-HKONT highlighted.

```
485 | ENDFORM.
486 |
487 | * *
488 | * *      form u150
489 | * *
490 | * *      reads the personnel number from
491 | * *
492 | * *      form u150 using pernum.
493 | * *      clear pernum.
494 | * *      select * from zzper
495 | * *          where subuser eq syst-una
496 | * *          move zzper-subper to pernum
497 | * *
498 | * *      endselect.
499 | * *
500 | * *      endform.
501 |
502 | FORM U995 USING BSEG-MATNR.
503 |
504 | SELECT * FROM ZZPRCTR
505 |     WHERE ZZKONTO EQ BSEG-HKONT.
506 | MOVE ZZPRCTR-ZZSUBPRCTR TO BSEG-PRCTR
507 |
508 | ENDSELECT.
509 |
510 | ENDFORM.
511 |
512 |
513 | * *
514 | * *      FORM UGAI
515 | * *
516 | * *      Inserts the sum of the posting in
517 | * *      This exit can be used in FI for t
518 | * *      The complete data is passed in or
519 | * *
520 | * *
521 | * *      FORM UGAI
522 | * *
523 | * *      *
```

St...	Sta...	S..	Event Type	Event	Program
	11	FORM	FORM	U995	ZGGBS800
	10	FORM	FORM	SUB_OC-EPRC	GBTTQFI0
	9	FORM	FORM	E15_SUBSTITUTION_CA...	GBTTQFI0
	8	FUNCTION	FUNCTION	G_VSR_SUBSTITUTION_...	SAPLGBL5
	7	FUNCTION	FUNCTION	FI_SUBSTITUTION_ITEM	SAPLFACTS
	6	FORM	FORM	SUBSTITUTION_POSITI...	SAPMF05A
	5	FORM	FORM	PAI_ZEILE_SPEICHERN	SAPMF05A
	4	MODULE (PAI)	MODULE (PAI)	ZEILE_SPEICHERN	SAPMF05A
	3	PAI MODULE	PAI MODULE	ZEILE_SPEICHERN	
	2	PAI SCREEN	PAI SCREEN	0300	SAPMF05A
	1	TRANSACTION	TRANSACTION	FB01(FB01)	

S...	Variable	V...	Val.
	BSEG-HKONT		0000810201
	ZZPRCTR-ZZSUBPRCTR		
	BSEG-PRCTR		

# Debugger Step

- Press the single step button (F5) to process code one-line at a time to see values selected
- Here can see select statement is completed and display value from table ZZPRCTR

ABAP Debugger(1) (Exclusive)(ehp8ides\_EH8\_02)

Step Size Watchpoint Layout Configure Debugger Layer

ZGGBS800 / ZGGBS800 / 506 SY-SUBRC 0

FORM / U995 SY-TABIX 0

Desktop 1 Desktop 2 Desktop 3 Standard Structures Tables Objects DetailDisplay Data Explorer Break./Watch...

```
485 ENDFORM.
486
487 * * -----
488 * *      form u150
489 * *
490 * *      reads the personnel number fr
491 * * -----
492 * *      form u150 using pernum.
493 * *      clear pernum.
494 * *      select * from zzper
495 * *              where subuser eq syst-una
496 * *              move zzper-subper to pernu
497 * *
498 * *      endselect.
499 * *
500 * *      endform.
501
502 FORM U995 USING BSEG-MATNR.
503
504 SELECT * FROM ZZPRCTR
505        WHERE ZZKONTO EQ BSEG-HKONT.
506        MOVE ZZPRCTR-ZZSUBPRCTR TO BSEG-PRCTR
507
508 ENDSELECT.
509
510 ENDFORM.
511
512
513 * * -----
514 * *      FORM UGAI
515 * * -----
516 * *      Inserts the sum of the posting in
517 * *      This exit can be used in FI for t
518 * *      The complete data is passed in or
519 * * -----
520 * *
521 * *      FORM UGAI
522 * * -----
523 * *      *
```

ABAP and Screen Stack

St...	Sta...	S..	Event Type	Event	Program
→	11		FORM	U995	ZGGBS800
	10		FORM	SUB_OC-EPRC	GBTTQFI0
	9		FORM	E15_SUBSTITUTION_CA...	GBTTQFI0
	8		FUNCTION	G_VSR_SUBSTITUTION_...	SAPLGBL5
	7		FUNCTION	FI_SUBSTITUTION_ITEM	SAPLFACTS
	6		FORM	SUBSTITUTION_POSITI...	SAPMF05A
	5		FORM	PAI_ZEILE_SPEICHERN	SAPMF05A
	4		MODULE (PAI)	ZEILE_SPEICHERN	SAPMF05A
	3		PAI MODULE	ZEILE_SPEICHERN	
	2		PAI SCREEN	0300	SAPMF05A
	1		TRANSACTION	FB01(FB01)	

Variables 1 Variables 2 Locals Globals A...

S...	Variable	V...	Val.
	BSEG-HKONT		0000810201
	ZZPRCTR-ZZSUBPRCTR		PC01
	BSEG-PRCTR		

ABAP | Ln 506 Col 1



# Wrap Up

---

# Where to Find More Information - SAP Help



- [https://help.sap.com/doc/saphelp\\_nw75/7.5.5/en-US/a4/19aa9c58c2473bb4e3ae3c2a00b7b8/frameset.htm](https://help.sap.com/doc/saphelp_nw75/7.5.5/en-US/a4/19aa9c58c2473bb4e3ae3c2a00b7b8/frameset.htm)

The screenshot shows the SAP Help documentation page for the ABAP Debugger. The page is titled "ABAP Debugger" and includes a navigation sidebar on the left, a main content area with a description and a note, and a "Related Topics" section at the bottom.

**SAP Documentation** Advanced Online Search  
SAP Library | Glossary | Support | Copyright | Disclaimer

**ABAP Debugger** ⓘ

The ABAP Debugger is an integrated test tool within the ABAP Workbench with which you can stop running program und then execute them line-by-line or section-by-section. With this tool, you can display contents of data objects and check the flow logic of programs so that any errors in the ABAP source code of the programs can be found quickly.

**Note**  
Note that the (old) Classic ABAP Debugger has been declared as obsolete. We recommend using the Standard ABAP Debugger. Therefore it is not possible any more to set the Classic ABAP Debugger as default Debugger using the Workbench settings. If you still like to use the Classic ABAP Debugger, then click the info button on the debugging settings screen for more information. To access these settings, follow the menu path: **Utilities > Settings > ABAP Editor > Debugging**

**Related Topics**

- [Standard ABAP Debugger \(Recommended!\)](#)
- [Classic ABAP Debugger](#)

**Navigation Sidebar:**

- ABAP Test and Analysis Tools
  - ABAP Test and Analysis Tools
    - Extended Program Check
    - Code Inspector
    - Quality Checking with the ABAP Test Cockpit
  - ABAP Unit
  - eCATT: Extended Computer Aided Test Tool (BC-TWB-TST-ECA)
  - CATT Computer Aided Test Tool (BC-CAT-TOL)
  - SAP NetWeaver System Logging (SM21)
  - Coverage Analyzer
    - Using the Coverage API
  - ABAP Dump Analysis
  - Runtime Analysis
  - Performance Analysis
  - Dynpro Trace
  - ABAP Debugger**
    - Standard ABAP Debugger
    - Classic ABAP Debugger

# Where to Find More Information - IDES

---



- **IDES – Internet Demonstration and Evaluation System**

- <https://wiki.scn.sap.com/wiki/display/HOME/IDES> \*
- <https://launchpad.support.sap.com/#/notes/799639>
  - SAP system with sample transactions and configuration
  - Website and note explains how to get an IDES system

\* Requires login credentials to the SAP Service Marketplace

The screenshot displays the SAP Community Wiki interface. At the top, there is a navigation bar with the SAP logo, 'Community WIKI', and 'SAP Community'. On the right side of the navigation bar, there are links for 'Getting Started' and 'Store', and a search bar labeled 'Search the Community'. Below the navigation bar, there are 'Spaces' and 'Browse' tabs. The main content area shows the title 'IDES' under the heading 'Wiki / FAQ'. Below the title, it states 'Created by Marilyn Pratt, last modified by Former Member on Jun 22, 2017'. The main text of the page reads: 'IDES Welcome to the IDES -Internet Demonstration and Evaluation System FAQ. This is the SDN WIKI starting point for topics around IDES. Feel free to create and correct entries.'

## Where to Find More Information – SAP Experts

---



- Gaurav Agarwal, “Tips for Debugging ABAP for Functional Teams” (SAPinsider).
  - <https://sapinsider.org/tips-for-debugging-abap-for-functional-teams/>

## Where to Find More Information – SAP Experts

---

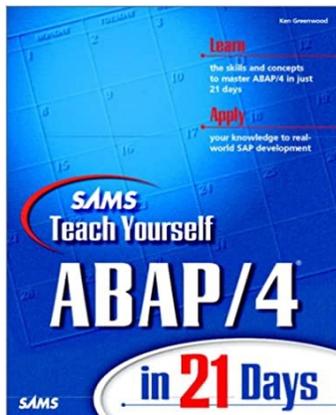


- Rohana Gunawardena, “Discover Hidden Parameter IDs to Simplify Your FI Settings” (Financials Expert, October 2003).
  - <https://www.sapinsideronline.com/articles/discover-hidden-parameter-ids-to-simplify-your-fi-settings/>
- Rohana Gunawardena, “Update Was Terminated: What Every FI/CO User Should Know About This Error Message” (Financials Expert, February 2002).
  - <https://sapinsider.org/Update-Was-Terminated-What-Every-FI-CO-User-Should-Know-About-This-Error-Message/>

# Where to Find More Information – SAP Press



- Nachiket Chavan, Learn to Debug ABAP Programs: Debugging for Functional Consultant (Amazon Kindle).
  - Debugging in more detail



- Ken Greenwood, SAMS Teach Yourself ABAP/4 in 21 Days (SAMS 1998).
  - Learn ABAP basics

# Key Points to Take Home

---

- **SAP behavior is based on the code not documentation or SAP Notes**
  - Need to look at the code to see what is occurring
- **Often SAP error messages are unclear**
  - Need to look at code to find the root cause
- **Debugging can be used by functional analysts**
  - A small sub-set of skills can be very powerful
- **For more complex scenarios you will need technical help**
  - Not all scenarios follow the simple examples I have provided



# Key Points to Take Home

---

- **As long as you don't have change variable access in production you can't break anything**
  - Debugging seems dangerous, but in reality you can not break anything
- **Do need appropriate security roles to access debugger**
  - Usually provided to functional analysts
- **This presentation touches the tip of the iceberg**
  - There is a lot more to debugging and coding, but you don't need to know it all to get value



# Thank you! Any Questions?

---

- Rohana Gunawardena
- Exium Inc.
- Rohana@Exium.com

Please remember to complete  
your session evaluation.

# SAPinsider



## SAPinsider.org

PO Box 982Hampstead, NH 03841  
Copyright © 2023 Wellesley Information Services.  
All rights reserved.

SAP and other SAP products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of SAP SE (or an SAP affiliate company) in Germany and other countries. All other product and service names mentioned are the trademarks of their respective companies. Wellesley Information Services is neither owned nor controlled by SAP SE.

---

SAPinsider comprises the largest and fastest growing SAP membership group worldwide, with more than 600,000 members across 205 countries.

---