

# How to Set a Session Breakpoint in a Transformation Program

## Applies to:

SAP NetWeaver BW 7.0

## Summary

**Topic:** How to set a session breakpoint in an SAP BW 7.0 generated transformation program.

**Use:** Very useful for debugging individual ABAP transformation rules.

**Created on:** 16 October 2007

## Author Bio

**Author:** Brad Dixon

**Company:** MassMutual Financial Group

**Title:** BW Systems Consultant

### SAP Experience:

2000 – 2003	SAP HR & Payroll Configuration
2004 – 2005	SAP HR Lead Designer & Architect
2006	SAP BW Developer
2007	SAP BW Lead Designer

## Table of Contents

Scenario .....	3
Step by step procedure .....	4
Disclaimer and Liability Notice .....	9

## Scenario

You have coded individual transformation rules, and would like to confirm that the code is working as expected. One possible solution is to code a hard breakpoint. This however becomes a nuisance when you are processing many records, and you want to have the ability to continue beyond the breakpoint after you confirm the first few records.

This document outlines an alternative approach; whereby, you set a session breakpoint in the generated transformation program, which can optionally be deleted during the debug session. Also, this eliminates the need to go back and update the code, to remove the hard breakpoint.

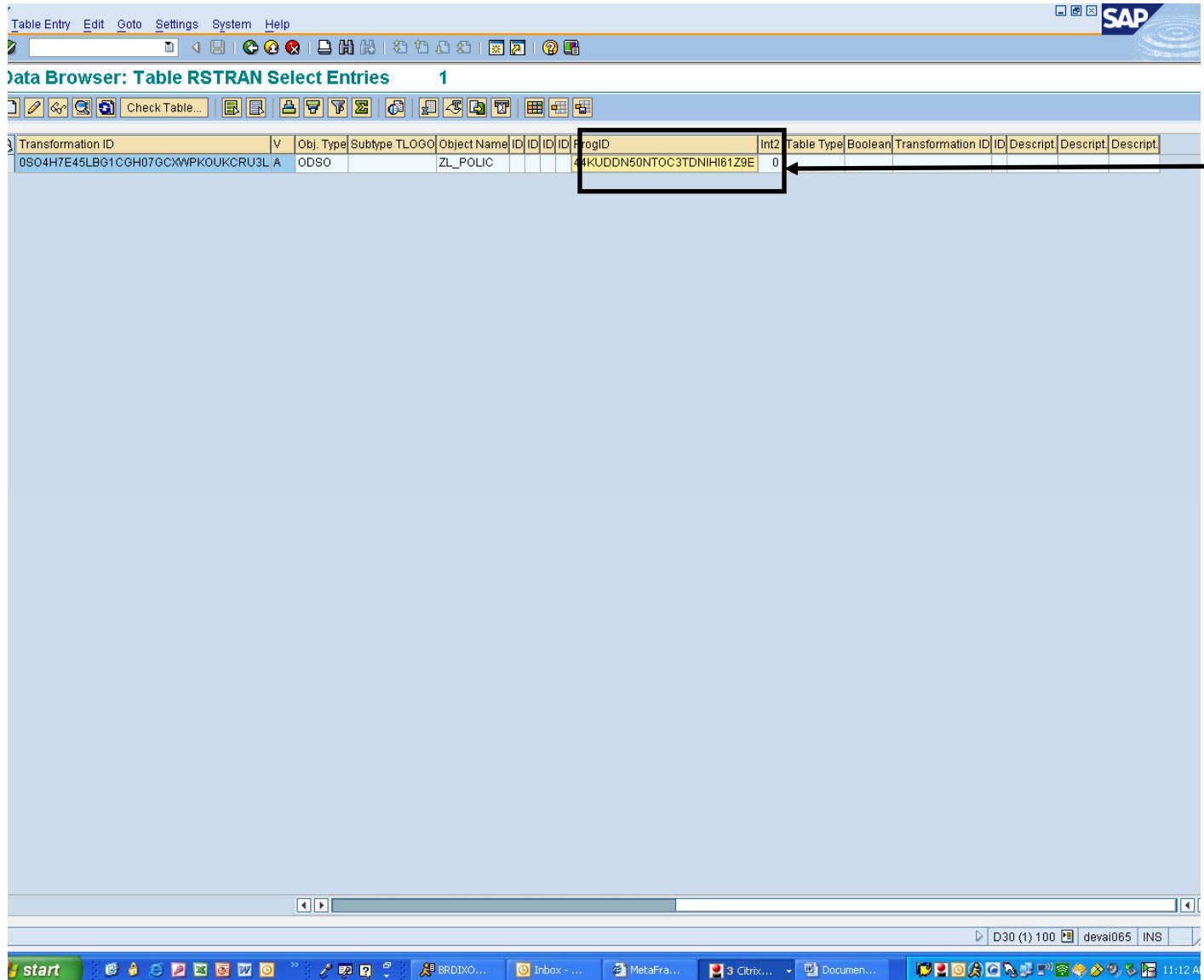
## Step by step procedure

1) In RSA1, get the Transformation ID

InfoProvider	Tech. Name	M	Execute Function	Display Tree	Ob.	Object Informati	Object Inf
Country-Specific Content	0COUNTRY_SPECIFIC		Change				
Supply Chain Management	ZSCM		Change				
Human Resources	0HCM		Change				
Technical Content	0BWTCT		Change				
SAP Verification	0BWVC		Change				
Test Info Area LS - Leslie Smith	TEST_AREA_LS		Change				
Unassigned Nodes	NODESNOTCONNECTED		Change				
Dave's Testing InfoArea	ZDF_INFOAREA		Change				
Financials Management & Controlling	0FMCO		Change				
Customer Relationship Management	0CRM		Change				
Strategic Enterprise Management	0SEM		Change				
Architecture Services Group	Z9ARCH_SVS_GRP		Change				
MATT'S TESTING INFOAREA	ZTESTMATT		Change				
Marketplace	0MA		Change				
Cross-Application Components	0CA		Change				
Supplier Relationship Management	0SRM		Change				
Information Security	ZINFSEC		Change				
Joe's Test Area	ZJFF_TEST		Change				
Cash Equivalents	Z9CASHIEQ		Change				
Yan's Testing Infoarea	ZYANTEST		Change				
Purchasing - MMFG	ZPUR_MM		Change				
BW Core Administration	ZBW_CORE_ADMIN		Change				
Life Information Factory	ZLIF		Change				
LIF - Inbound	ZINBOUND		Change				
LIF - Outbound	ZOUTBOUND		Change				
LIF - Consolidation	ZCONSOLIDATION		Change				
Party Consolidated	ZL_PARTYC		Change				
Policy Consolidated	ZL_POLICYC		Change				
Policy - Consolidated	ZL_POLIC		Manage				
ODSO ZLPLBUFA -> ODSO ZL_POLIC	0S04H7E45L BG1CGH07GCXWPKOUKCRU3L		Change				
ODSO ZL_IDVYR -> ODSO ZL_POLIC	0C73CN658B025628C11_21002T004T1NBRV		Change				
ODSO ZL_INPOL -> ODSO ZL_POLIC	02J8JT6EQXT8E9Z0THALV741URA430BD		Change				
ODSO ZL_INPVL -> ODSO ZL_POLIC	07DLW1W499KG1PT0NKBOZCP9QEPZ02WVL		Change				
ODSO ZL_PMODC -> ODSO ZL_POLIC	0HHOPPJDFOX2S9N7MLSHX7X3W14FDZ9		Change				
ODSO ZL_RSXBI -> ODSO ZL_POLIC	0N64UOEFWXY7YC4VA12JZRC1YM5X9U7G		Change				
ODSO ZL_RVLNI -> ODSO ZL_POLIC	065H88BMIM3LM54J41RW18DASIB68S0Y		Change				
ODSO ZL_TXBEE -> ODSO ZL_POLIC	0LZSU3UF29I2LXPRE4NCD3DE7WD46CPZ		Change				
Data Transfer Process	ZL_POLIC		Create Data Tra...				
Policy Consolidated History	ZL_POLIH		Manage				
Policy Modal Premium Amt Consolidated	ZL_PMODC		Manage				
Policy, Base Units & Face Amount Consolidated	ZLPLBUFA		Manage				
Coverage Consolidated	ZL_COVERAGEC		Change				



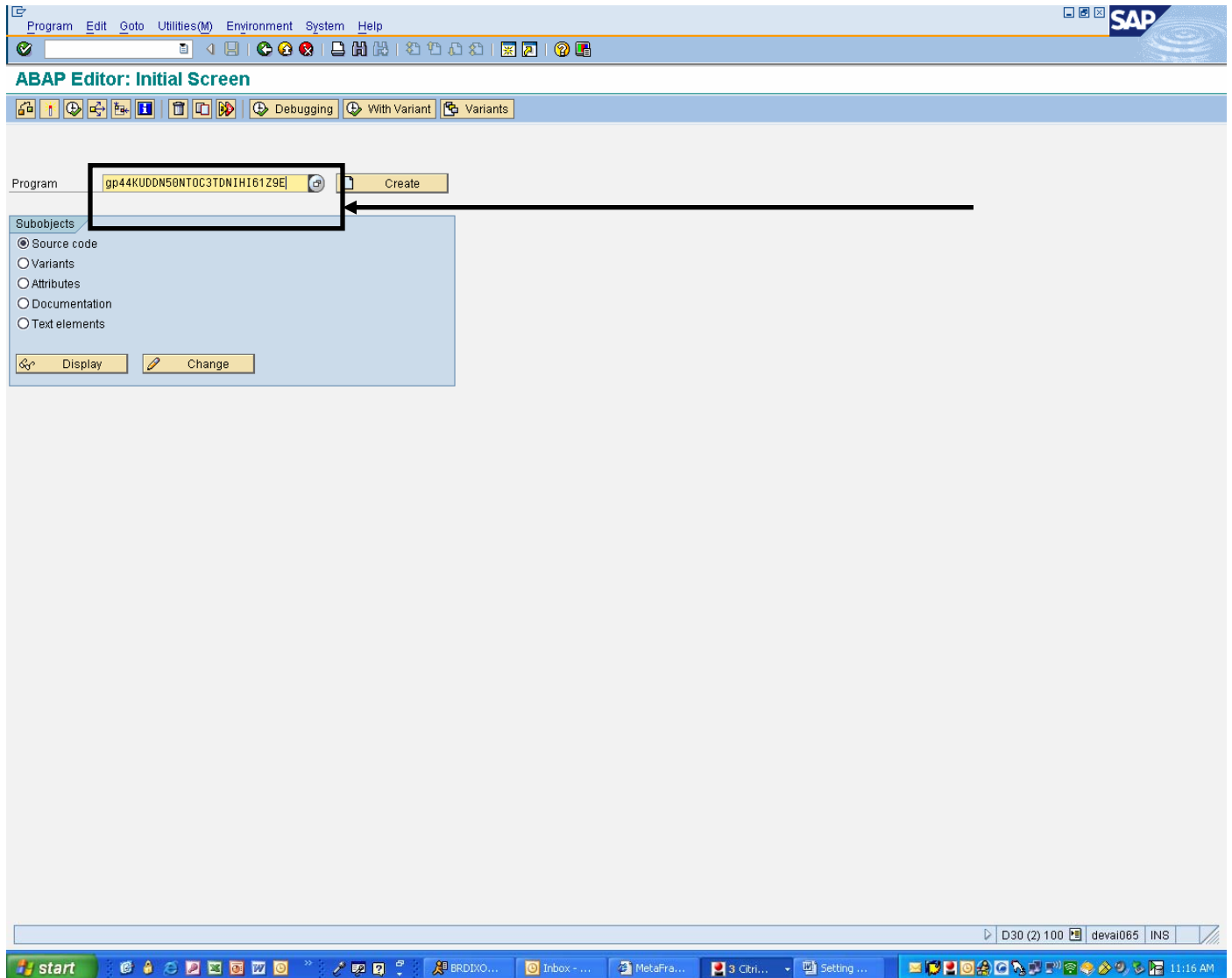
## 3) Capture the Program ID (PROG ID) from the selected table row



The screenshot shows the SAP Data Browser interface for the table 'RSTRAN Select Entries'. The table has the following columns: Transformation ID, Obj. Type, Subtype, TLOGO, Object Name, ID, ID, ID, ProgID, Int2, Table Type, Boolean, Transformation ID, ID, Descript, Descript, Descript. The first row is highlighted, and the ProgID column value '44KUDDN50NTOC3TDNIHI61Z9E' is enclosed in a red box. An arrow points from the right side of the table to the ProgID column.

Transformation ID	Obj. Type	Subtype	TLOGO	Object Name	ID	ID	ID	ProgID	Int2	Table Type	Boolean	Transformation ID	ID	Descript	Descript	Descript
0S04H7E45LBG1CGH07GCXWPKOUKCRU3L A	ODSO			ZL_POLIC				44KUDDN50NTOC3TDNIHI61Z9E	0							

## 4) Go to SE38, type in 'GP', and then paste the Program ID in after the 'GP'



- 5) Display the program, search for the line you want to set the breakpoint on, put the cursor on the line, and click the Stop Sign (w/ monitor) icon

The screenshot shows the SAP ABAP Editor interface. The title bar reads "ABAP Editor: Display Report GP44B0MOVR49UVXS02V44QR1Y9U". The menu bar includes "Program", "Edit", "Goto", "Utilities(M)", "Environment", "System", and "Help". The toolbar contains various icons, with a red square highlighting the "Set/Delete Session Breakpoint" icon (a stop sign with a monitor). Below the toolbar, the editor displays the source code of a class method. A red square highlights the "Set/Delete Session Breakpoint" icon in the toolbar, and a black arrow points from it to a specific line of code in the editor. The code is as follows:

```

1707 if sy-subrc = 0.
1708     result = wa_polic_guids-/bic/zl_poguid.
1709 else.
1710     call function 'GUID_CREATE'
1711         importing
1712             ev_guid_32 = w_guid.
1713     result = w_guid.
1714 endif.
1715
1716 *-- fill table "MONITOR" with values of structure "MONITOR_REC"
1717 *-- to make monitor entries
1718 ... "to cancel the update process
1719 * raise exception type CX_RSROUT_ABORT.
1720 ... "to skip a record
1721 * raise exception type CX_RSROUT_SKIP_RECORD.
1722 ... "to clear target fields
1723 * raise exception type CX_RSROUT_SKIP_VAL.
1724
1725 * result value of the routine
1726
1727 endmethod. "compute 39 1
1728 method invert_39_1.
1729
1730 *$*$ begin of inverse routine - insert your code only below this line*-*
1731 ... "insert your code here
1732 *$*$ end of inverse routine - insert your code only before this line *-*
1733
1734 endmethod. "invert 39 1
1735
1736
1737 endclass. "lcl transform IMPLEMENTATION
1738
1739 *-----*
1740 *f Form get_runtime_ref
1741 *f-----*
1742 * text
1743 *-----*
1744 * -->C_R_EXE text
1745 *-----*

```

The status bar at the bottom of the editor shows "ABAP Ln 1700 Col 7". Below the editor, a message bar displays "Session breakpoint was set". The Windows taskbar at the bottom shows the system tray with the time "12:34 PM".

#### NOTE:

Refresh RSA1 after setting the session breakpoint, as this will refresh the cache, and will ensure that the breakpoint will be active when running the DTP in debug mode.



## 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.