Fetching Data from Memory Stack Using New Tool (in Debugger)

By Prapti Vyas - September 5, 2017

Many times we receive a requirement to add validations for standard transactions *wherein the field value is not directly accessible in user exit or BADI*. So if you are also struggling with such requirement, your wait is over. Prapti is here to your rescue. 😉

In this article, we will be learning a short and simple trick to access screen values in standard transaction code.

Basically, we would exploit the **Memory Stack** at **Run-Time** using the **New Tool** in **Debugger**. If in any interview, someone asks you, **do you know Memory Stack Programming?** After going through this article, your answer should be a big confident affirmative. 😛

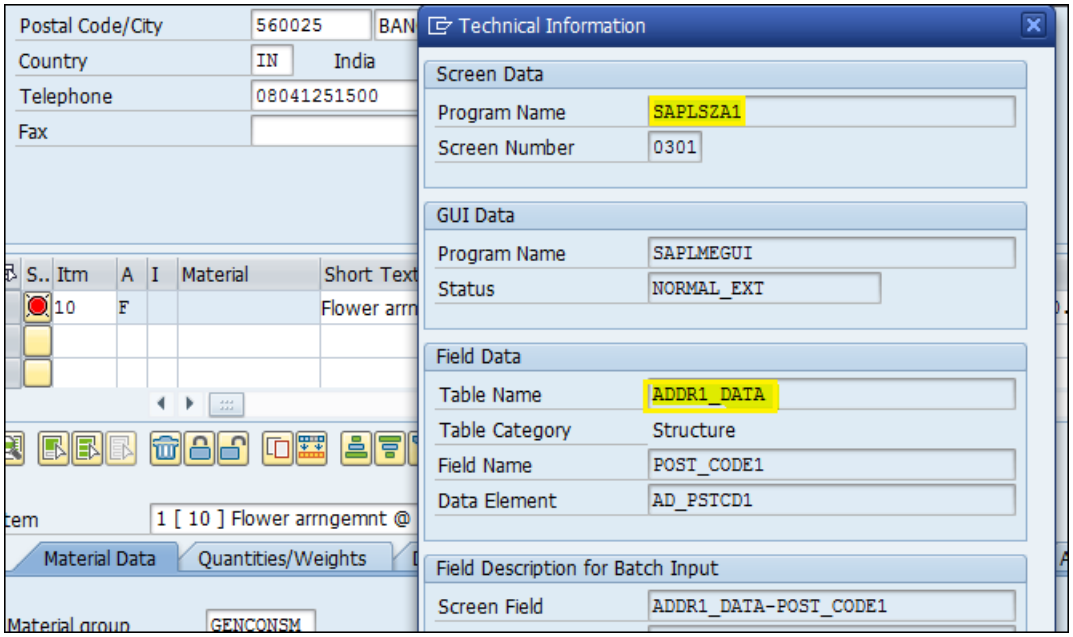
Following are the steps that need to be followed:

1. Get the program name and field data from the screen (using F1)
2. Add a breakpoint in the exit / BADI
3. Execute the transaction and check the value at runtime using New Tool (Tools -> Special Tool -> Loaded Programs)
4. Add code in the exit / BADI for fetching data from the screen field

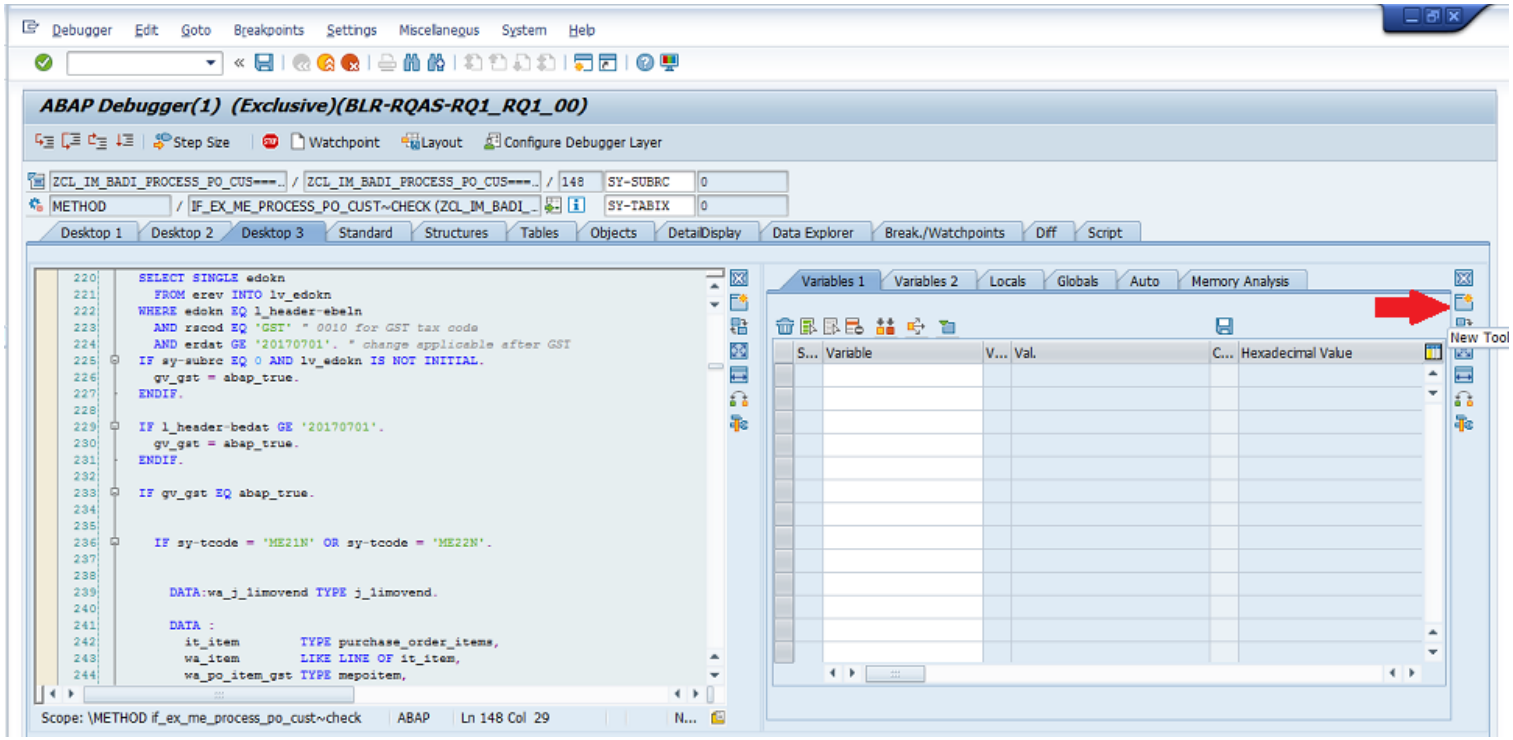
Seeing is Believing. Let me take a very simple real time practical example. While creating a purchase order in SAP, there is a provision to add one-time vendor details. Our requirement is to add a **Validation on PINCODE length**.

We were using **CHECK** method of BADI **ME_PROCESS_PO_CUST** for validation in PO. *But found that the address details were not accessible in the BADI.*

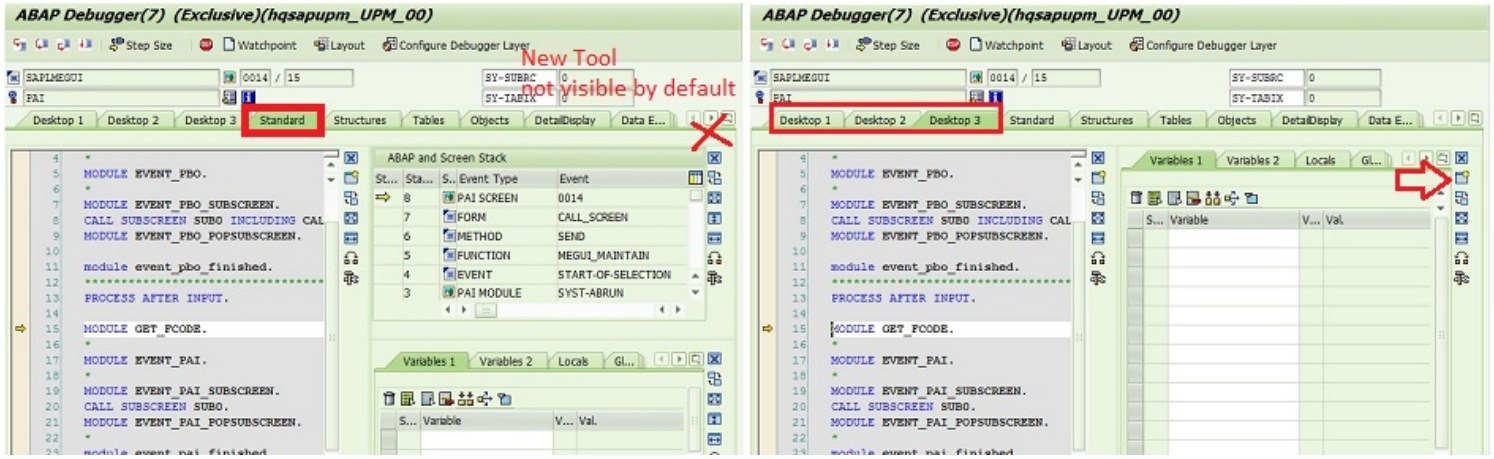
We followed the 4 steps mentioned above. We found the program name and field data.



Added the debugger and executed the transaction. In the debugger tool, go to **New Tool** option.

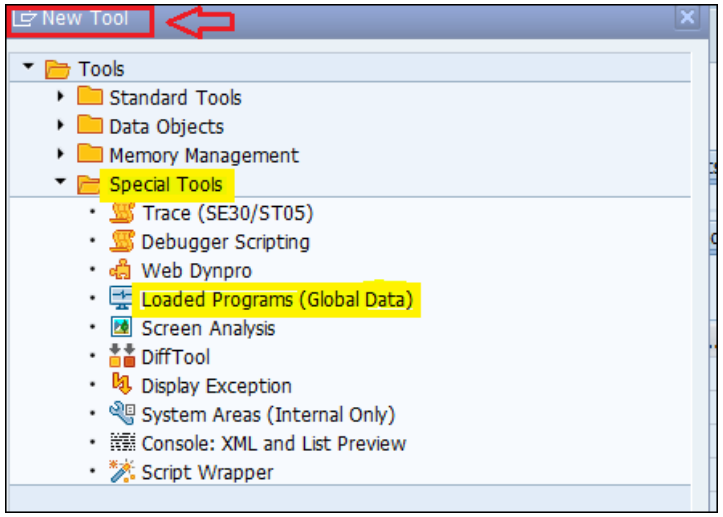


Alert: *If your debugger takes you to* ***Standard Tab*** *by default, then you would not find the "****New Tool****". Do not panic. Just click Desktop 1 or Desktop 2 or Desktop 3 and New Tool would be right there for your service.* 🙂

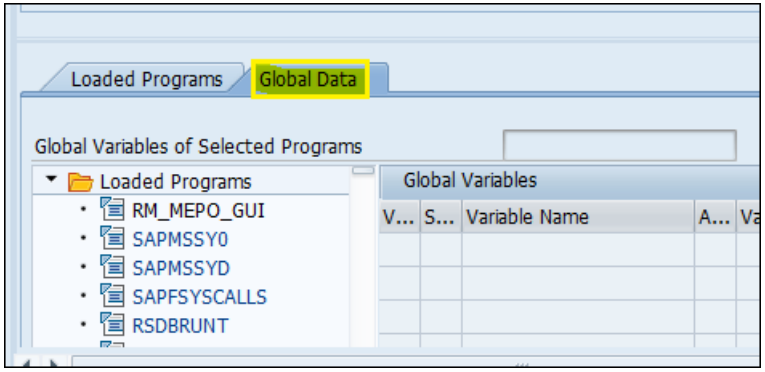


Also Read: A to Z of Integration of SAP Ariba with SAP ECC

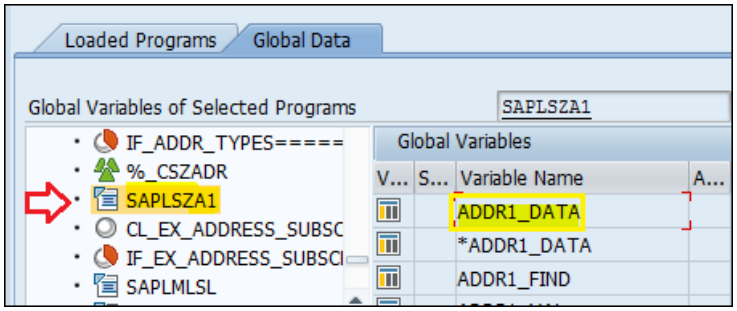
Go to **Loaded Programs** option in **Special Tools**.



Select **Global Data** tab where we will be able to see all the global variables of each program.



Let us now find the program and field for Address data.



Yeah, we got the ADDR1_DATA values. So now we reached the last step – the code :).

```
  FIELD-SYMBOLS:<fs1> TYPE any.

* Make the system know from which Program we need the data
  DATA:post_code(40) TYPE c VALUE '(SAPLSZA1)ADDR1_DATA-POST_CODE1'.

* Assign the value to a field system
  ASSIGN (post_code) TO <fs1>.

  IF  <fs1> IS ASSIGNED.
* Do your validation
    IF strlen( <fs1> ) NE '6'.
          MESSAGE 'Postal Code should be of length 6.' TYPE 'E'.
    ENDIF.

  ENDIF.
```

Basically, we just dug deep into the memory stack at the run-time and *looked for all programs that were loaded in the system for completion of the particular transaction*. Using the New Tool, we found the Global Data. Once you have the Global Data, you can just fetch that data in your program using Field Symbol.

Hope you like this trick. This is a simple solution which we need to do in every project. If you are a beginner on SAP ABAP, bookmark this article. You would need this for your first Memory Stack Programming. 😛

You can implement this same trick to fetch any global data like table, structures, variables, objects etc.

Note: *There is a **limitation**. If the variable, work area, table etc are **NOT** declared as **Global** on the loaded programs, then we **CANNOT** retrieve its value using the above method.*

Also Check: How to write your first Program in HANA ABAP?

Now I want to hear from you. Do you have any Tip or Trick to share?

Please do not shy away from sharing your knowledge. Together we learn better. Please leave your quick feedback below. Please **Like** Us on Facebook and **Share** our Contents.

Prapti Vyas

https://sapyard.com/

Prapti is a senior IT Analyst. She has more than 5 years of experience in SAP ABAP and around 2 years in SAP PI. She is a vivid reader (mostly novels) and a versatile dancer. Dancing is her passion and her favourite past time. Please find more information about Prapti on LinkedIn.