How to debug Adobe Forms in SAP using breakpoints in the SFP interface

You cannot set a debugger directly in the SFP Adobe interface. To debug Adobe Forms in SAP, you set a breakpoint in the interface code and then run the generated global function module. This guide shows you how to choose between a global and a user-specific breakpoint, activate it, and step through the code in the ABAP Debugger.

Prerequisites

Before you start, make sure you have:

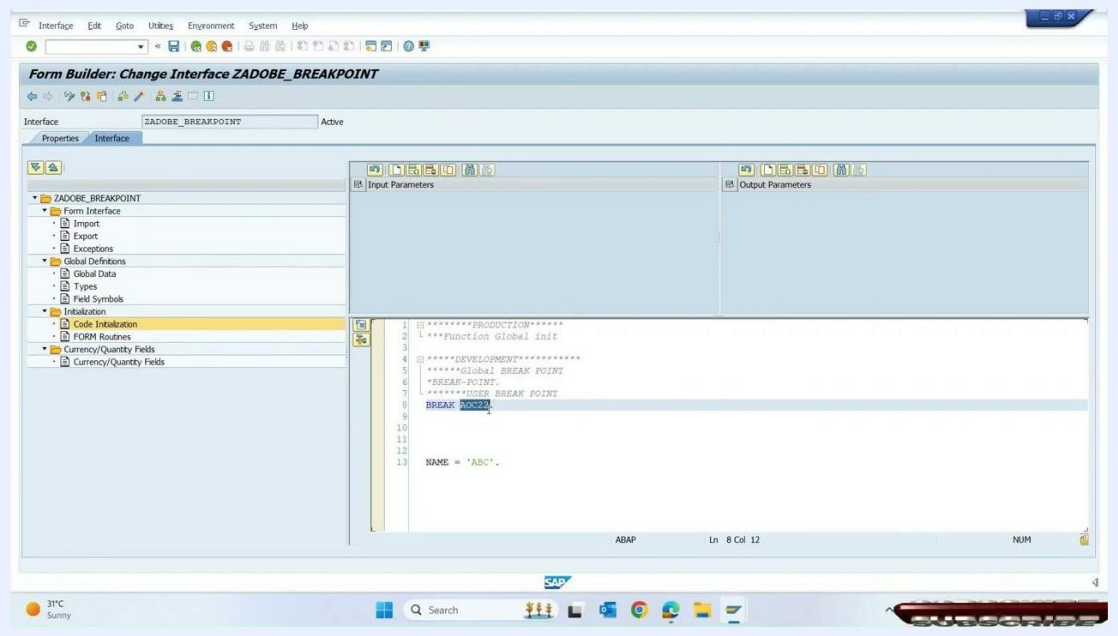
- Access to SAP Form Builder and the interface you want to debug (for example, ZADOBE_BREAKPOINT)
- Access to transaction SE37 (Function Builder)
- The name of the function module associated with your Adobe form (for example, /1BCDWB/SM00000091)

Step 1: Choose the breakpoint type

Because you cannot set a debugger directly in the interface, you use a breakpoint statement in the interface code. The statement you choose determines who it affects:

- BREAK-POINT. triggers for all users running the program.
- BREAK <USER_ID>. (for example, BREAK ACC22.) triggers only for the specified user.

This distinction matters in shared or production environments, where you may not want to interrupt other users.



Form Builder showing the Change Interface ZADOBE_BREAKPOINT screen with the BREAK ACC22 line highlighted for user-specific breakpoints.

Step 2: Open the interface in Form Builder

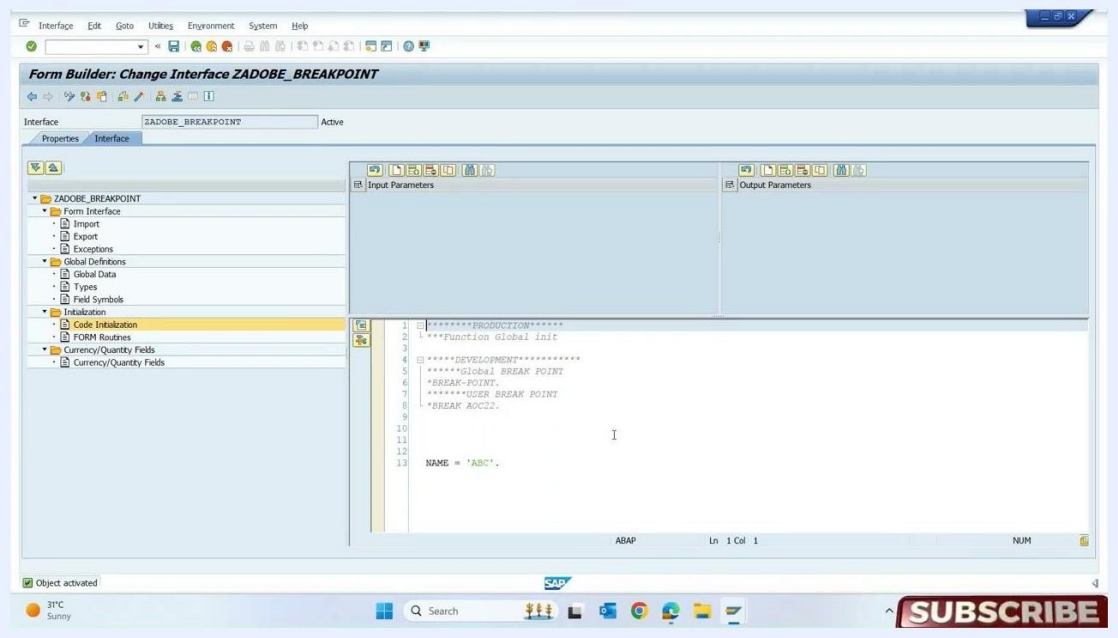
Launch SAP Form Builder and open your interface, for example ZADOBE_BREAKPOINT. Navigate to **Initialization** and select **Code Initialization**. Review the code area for comments and existing breakpoints.

The code area in this example looks like this:

```
******PRODUCTION******
***Function Global init

******DEVELOPMENT**********
******Global BREAK POINT
*BREAK-POINT.
******USER BREAK POINT
*BREAK ACC22.

NAME = 'ABC'.
```



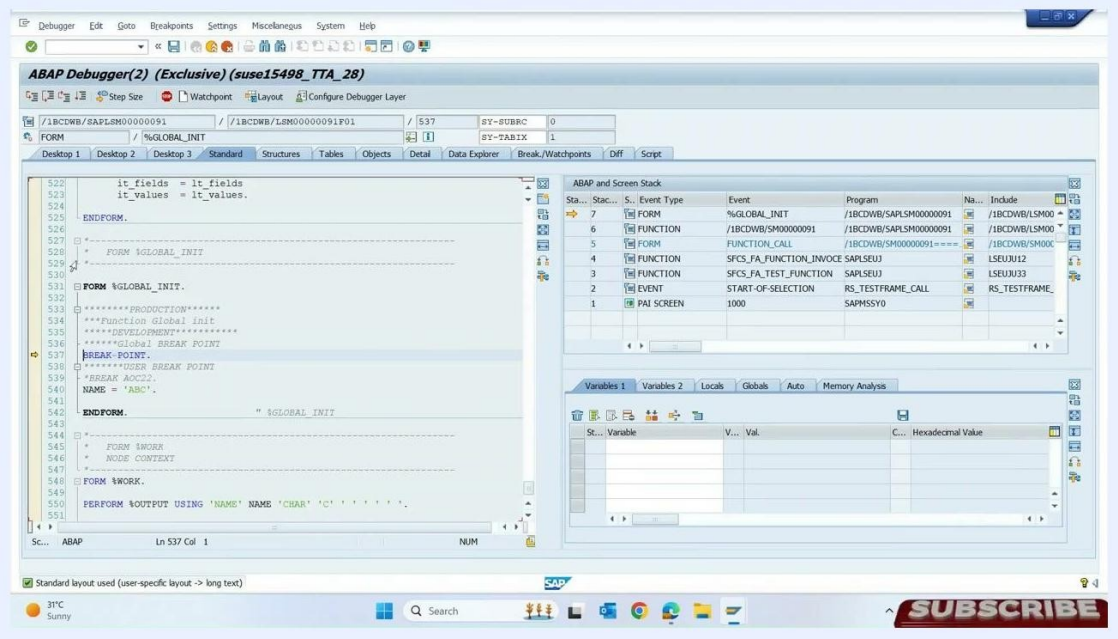
Form Builder showing Initialization > Code Initialization with breakpoint comments and the sample assignment NAME = 'ABC'.

Step 3: Select the breakpoint line to use

In the **Code Initialization** section, you comment or uncomment the breakpoint lines to match your debugging need:

- Uncomment `BREAK-POINT.` for a global breakpoint.
- Uncomment `BREAK <USER_ID>.` (for example, `BREAK ACC22.`) for a user-specific breakpoint.

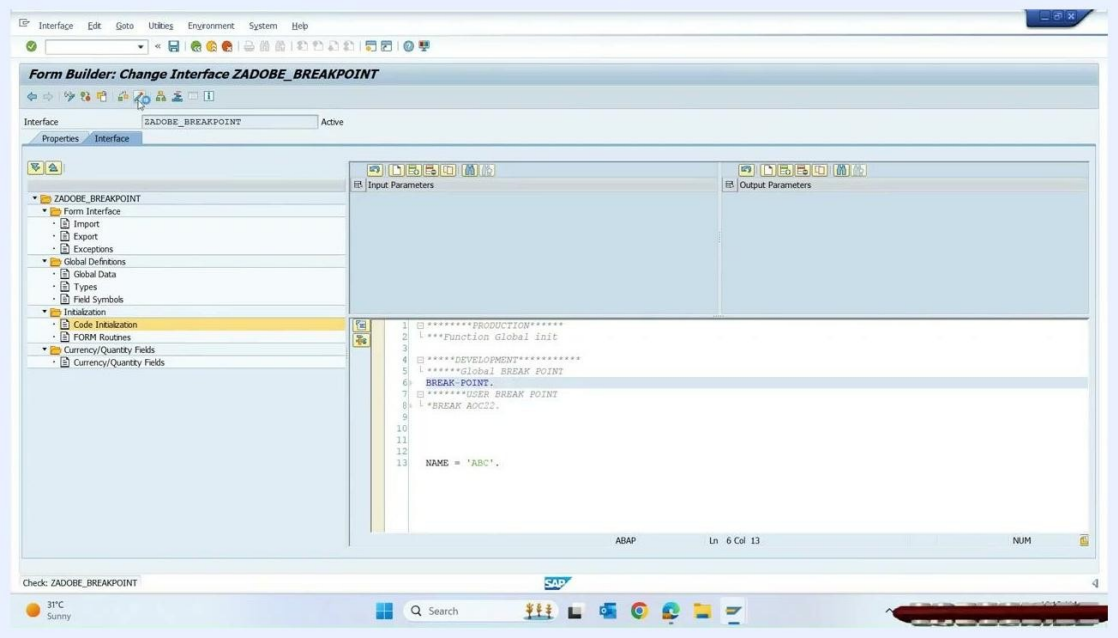
Use global breakpoints for broad debugging needs. Use user-specific breakpoints to limit the debugging session to your user ID and avoid disrupting others.



Form Builder showing both the BREAK-POINT and BREAK ACC22 lines in the code initialization section.

Step 4: Activate the breakpoint

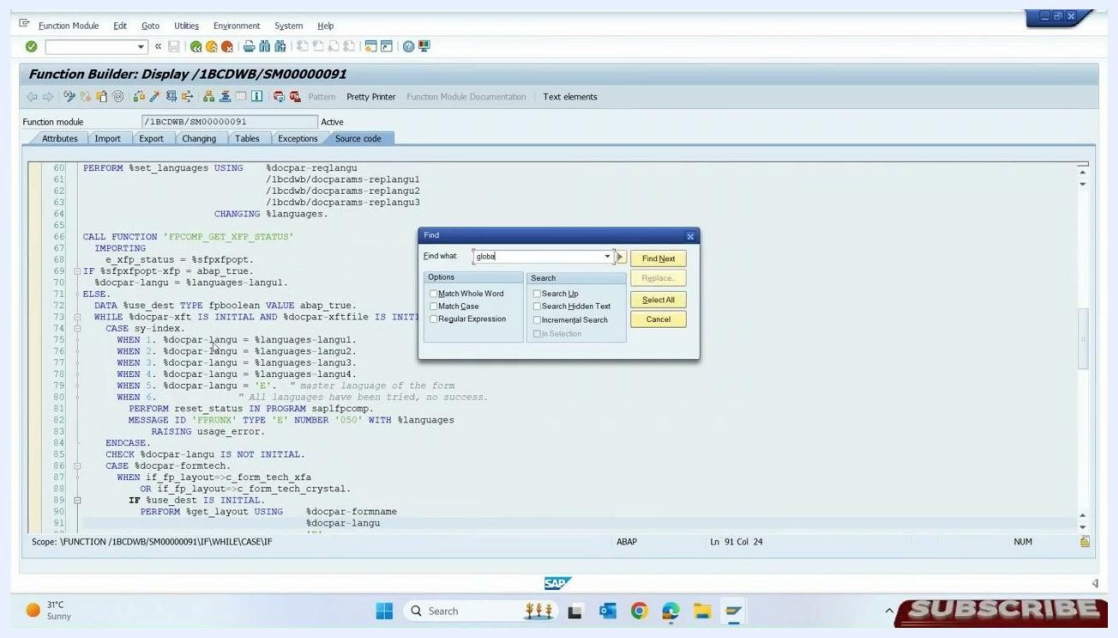
Make sure the breakpoint line you want is active, meaning it is not commented out. Then save and activate the interface object.



Form Builder showing BREAK-POINT active and BREAK ACC22 commented out.

Step 5: Open the function module in test mode

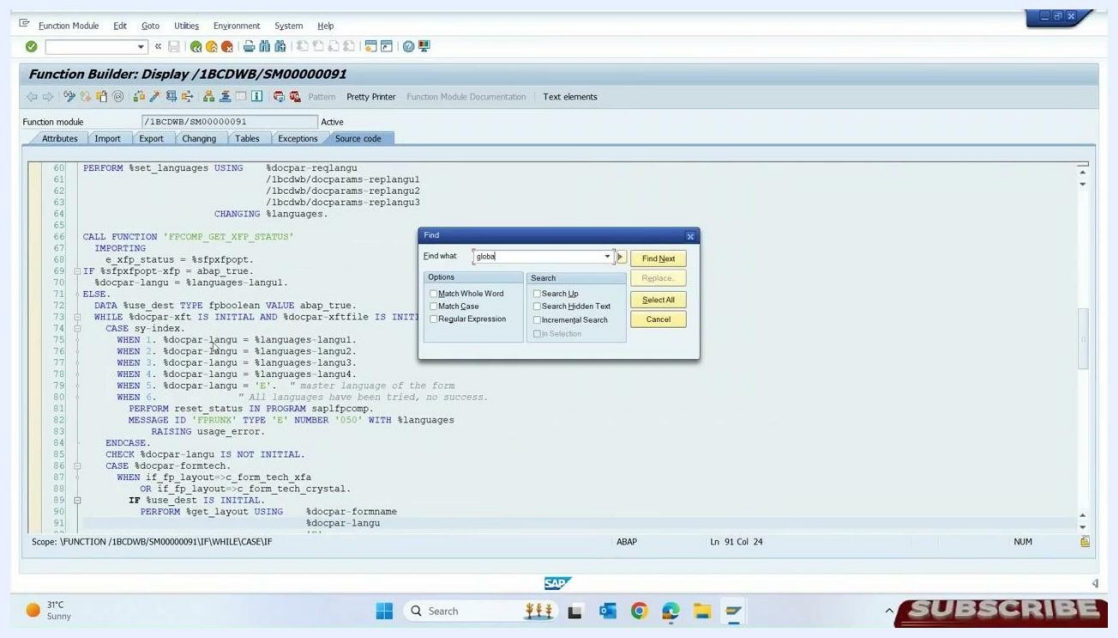
Select the function module associated with your Adobe form and open transaction SE37 (Function Builder). On the function module initial screen, enter the function module name, for example /1BCDWB/SM00000091. Make sure the **Debugging** option is checked if it is available.



Test Function Module initial screen showing the function module name and import parameters, with the NAME parameter set to TEST.

Step 6: Find the global initialization section

In the Function Builder, switch to the **Source code** tab. Use the Find dialog to search for the keyword `global` to locate the global initialization section.



Function Builder displaying /1BCDWB/SM00000091 with the Find dialog searching for "global" in the source code.

Step 7: Set a breakpoint in the global initialization

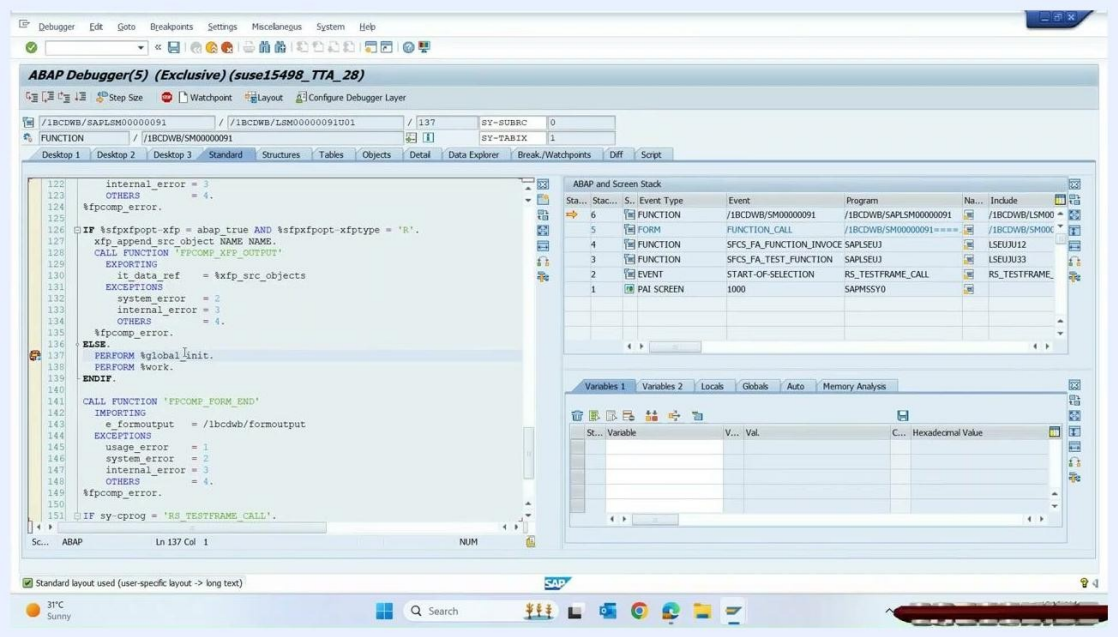
Locate the `PERFORM global_init.` line or a similar initialization routine, and set a breakpoint at this line. Confirm that the session breakpoint is set. A red marker should appear.

Step 8: Run the function module with test input

Return to the Test Function Module screen and enter the required input parameters. For example, set `NAME = TEST`. Then execute the function module.

Step 9: Verify the breakpoint in the ABAP Debugger

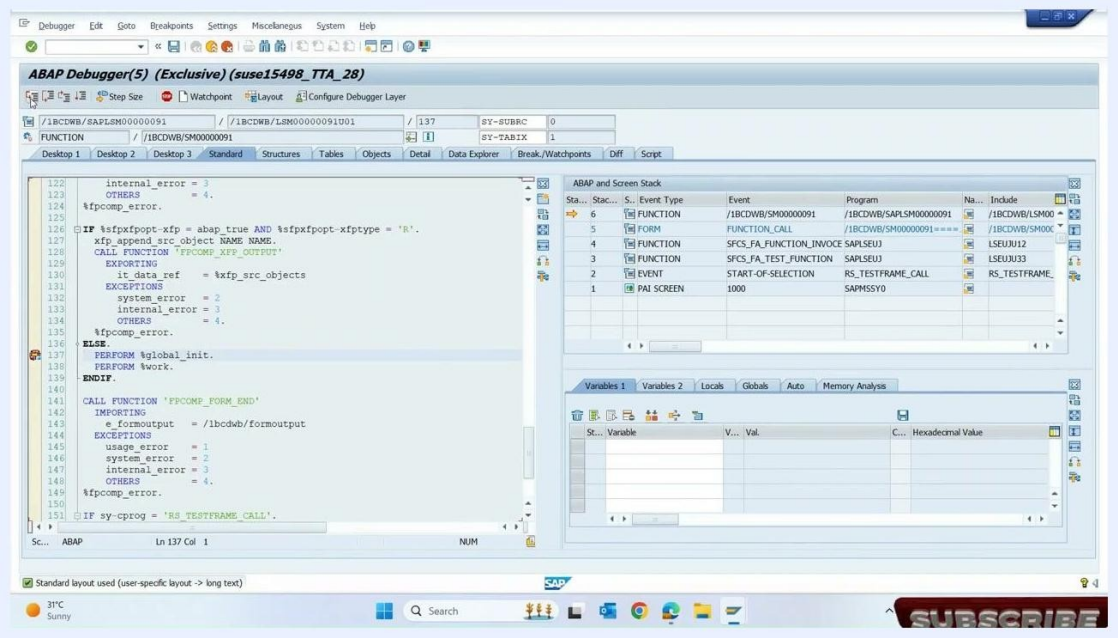
The ABAP Debugger opens when the breakpoint is hit. Confirm that execution stops at the global initialization section, and observe the variable values.



ABAP Debugger with execution stopped at PERFORM global_init., showing the call stack and variable windows.

Step 10: Step through the code and watch the variable change

Press F5 to step through the code in the ABAP Debugger. Execution enters your interface, and the variable NAME changes from TEST to ABC. After execution, the value is displayed as ABC. This confirms that debugging works at the interface level.



ABAP Debugger showing the NAME variable changing from TEST to ABC, with the call stack and variable windows visible.

Summary

You now know how to debug Adobe Forms in SAP even though the SFP Adobe interface does not support direct debugging. You choose a global or user-specific breakpoint, activate it in the interface, set a session breakpoint in the global function module, and step through the code in the ABAP Debugger.