

Debugging a Background Task in SAP Business Workflow



Applies to:

SAP ECC 6.0.

For more information, visit the [Business Process Modeling homepage](#).

Summary

The article demonstrates how to debug a background task in SAP Business Workflow.

Author: Vikram Shah

Company: L&T Infotech

Created on: 24 June 2009

Author Bio

Vikram Shah has an experience of 2 years in SAP. He has worked extensively in ABAP, HR-ABAP and Workflow. He is currently working for L&T Infotech.

Table of Contents

Background Task.....	3
Debugging Background Method	3
Related Content.....	9
Disclaimer and Liability Notice.....	10

Background Task

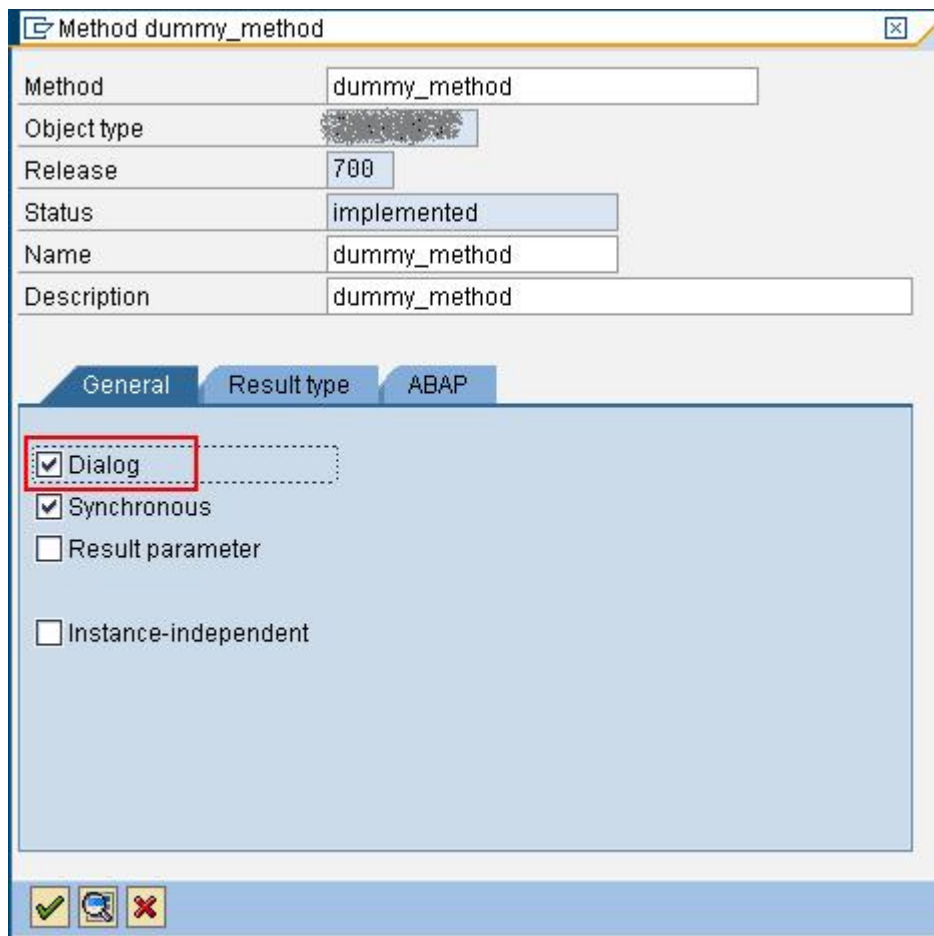
Quite often while testing a WF (Workflow), we come across a situation where the WF hangs at some background step but the WF log and transaction ST22 does not show any error/dump which can help us identify the problem.

In such cases it becomes necessary to debug the background method that is used by the WF task in question.

Debugging Background Method

Following steps must be followed in order to debug a background method and determine root cause of the problem.

1. Set the BO method as well as the WF task as Dialog (both were originally required to be set as background activities) for testing purpose.



The screenshot shows the configuration interface for a task named 'dummy_wi'. The interface is divided into several sections:

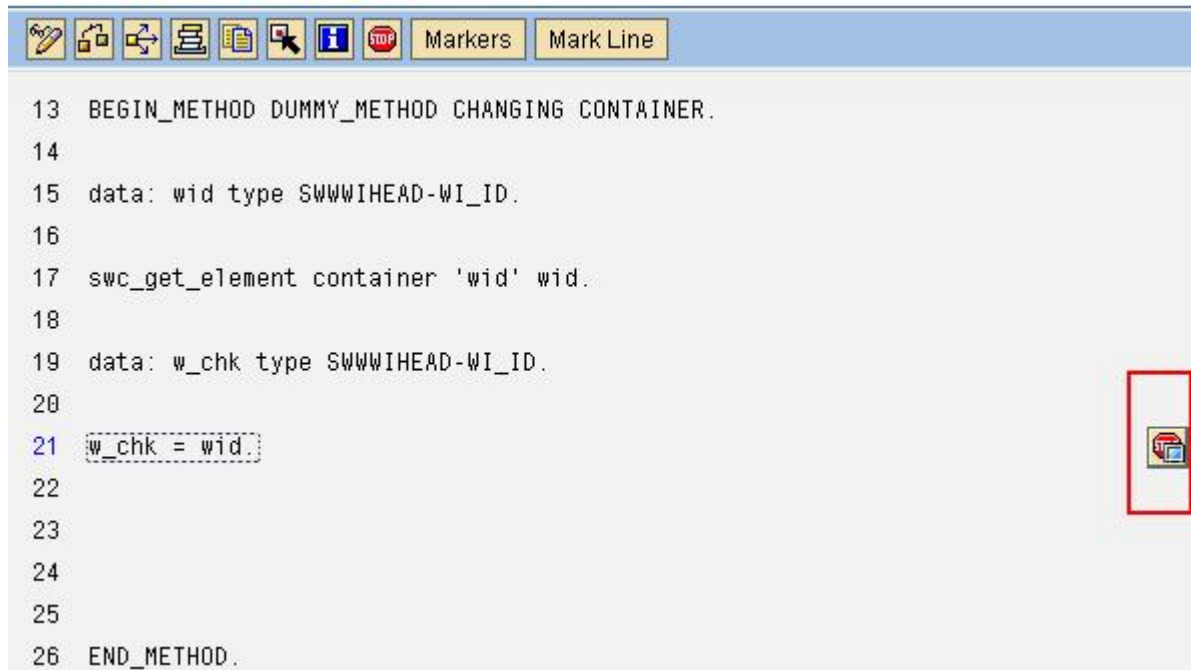
- Basic data:** Contains fields for 'Name' (dummy_wi), 'Abbr.' (dummy_wi), and 'Release status' (Not defined).
- Work Item Text:** A field for 'Work item text'.
- Object method:** Contains fields for 'Object Category' (BOR Object Type), 'Object Type' (blurred), and 'Method' (DUMMY_METHOD). It also has checkboxes for 'Synchronous object method' (checked) and 'Object method with dialog' (checked and highlighted with a red box).
- Execution:** Contains checkboxes for 'Background processing' (unchecked and highlighted with a red box), 'Confirm end of processing' (unchecked), and 'Executable with SAPforms' (unchecked).

2. Set the agent for the task as your own user ID.

The screenshot shows the configuration interface for a task named 'dummy_wi' with ID '000090'. The interface is divided into several sections:

- Task:** Contains fields for 'Task' (TS99701126) and 'Step Name' (dummy_wi).
- Agents:** Contains a field for 'User' (developer) which is highlighted with a red box.
- Task Properties:** Contains checkboxes for 'Agent Assignment' (checked), 'Background Processing' (unchecked), 'Task Complete' (checked), and 'Confirm End of Processing' (unchecked).

3. Set a breakpoint in the BO method.



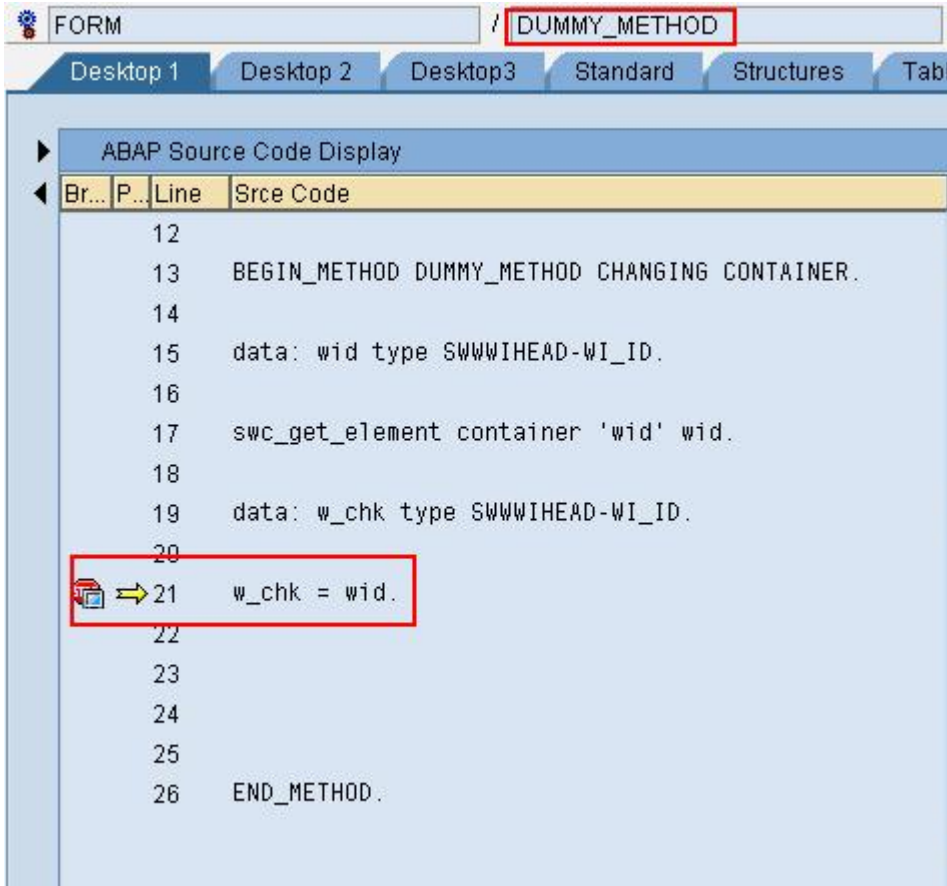
The screenshot shows the SAP Business Workflow Editor interface. At the top, there is a toolbar with icons for editing, deleting, and other actions, along with buttons for 'Markers' and 'Mark Line'. Below the toolbar, the code for a BO method is displayed. The code consists of several lines of ABAP-like syntax. Line 21, which contains the assignment statement 'w_chk = wid.', is highlighted with a dashed border, indicating that a breakpoint has been set at this line. A red rectangular box on the right side of the editor highlights a small icon, likely representing the breakpoint status.

```
13 BEGIN_METHOD DUMMY_METHOD CHANGING CONTAINER.  
14  
15 data: wid type SWWIHEAD-WI_ID.  
16  
17 swc_get_element container 'wid' wid.  
18  
19 data: w_chk type SWWIHEAD-WI_ID.  
20  
21 w_chk = wid.  
22  
23  
24  
25  
26 END_METHOD.
```

4. Trigger the WF. Now the step at which the WF had stopped/hanged earlier will appear as a WI (Workitem) in your inbox.

Ex...	Title	Status
	dummy_wi	

5. Go to the inbox and execute the WI. Since a breakpoint was set in the BO method, debugger will start on execution of the WI and then we can proceed with normal debugging.



The screenshot shows the SAP ABAP Source Code Display interface. The title bar indicates the current object is 'FORM' and the method being viewed is 'DUMMY_METHOD'. The interface includes several tabs: Desktop 1, Desktop 2, Desktop 3, Standard, Structures, and Table. The main area displays the source code of the 'DUMMY_METHOD' with the following lines:

```
12  
13 BEGIN_METHOD DUMMY_METHOD CHANGING CONTAINER.  
14  
15 data: wid type SWWWIHEAD-WI_ID.  
16  
17 swc_get_element container 'wid' wid.  
18  
19 data: w_chk type SWWWIHEAD-WI_ID.  
20  
21 w_chk = wid.  
22  
23  
24  
25  
26 END_METHOD.
```

A red box highlights the breakpoint icon (a red square with a white bug) and the yellow arrow pointing to line 21, which contains the statement 'w_chk = wid.'.

- When we are done with debugging, BO method and WF task must be once again set to their original status i.e. background activities.

The screenshot shows the 'Method dummy_method' dialog box. The 'General' tab is selected, and the 'Dialog' checkbox is highlighted with a red box. The 'Synchronous' checkbox is checked, while 'Result parameter' and 'Instance-independent' are unchecked. The 'ABAP' tab is also visible.

Method	dummy_method
Object type	
Release	700
Status	implemented
Name	dummy_method
Description	dummy_method

General | Result type | ABAP

Dialog
 Synchronous
 Result parameter
 Instance-independent

✓ | 🔍 | ✗

The screenshot displays the configuration interface for a background task in SAP Business Workflow. The interface is organized into several sections:

- Name:** Fields for Abbr. (dummy_wi), Name (dummy_wi), and Release status (Not defined).
- Work Item Text:** A field for Work item text.
- Object method:** Fields for Object Category (BOR Object Type), Object Type (blurred), Method (DUMMY_METHOD), and dummy_method. It includes checkboxes for Synchronous object method (checked) and Object method with dialog (unchecked).
- Execution:** Checkboxes for Background processing (checked), Confirm end of processing (unchecked), and Executable with SAPforms (unchecked).

Since debugging is one of the most basic approaches while identifying the cause of a problem, the approach mentioned above would be of great help in case of WF testing.

Related Content

[Business Process Management page in SDN](#)

[Business Process Modeling page in SDN](#)

[SAP Business Workflow in SDN](#)

[SAP Business Workflow documentation \(http://help.sap.com/\)](http://help.sap.com/)

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.