

Web Dynpro Debugging, Solution



Topic: Web Dynpro Debugging



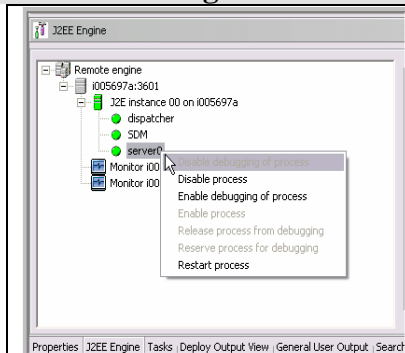
At the end of this Exercise, you are able to:

- 1) Debug and change the Component Context of a running Web Dynpro application.
- 2) *Optional* – HotSwap code

4 Debugging, Step-by-Step

4-1 Web Dynpro Debugging

4-1-1 Switching Server Nodes to Debug Mode.



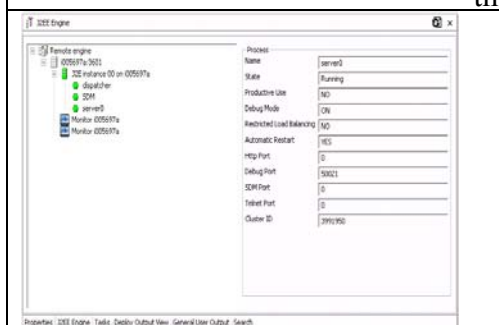
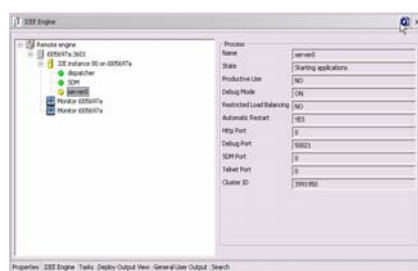
From Web Dynpro Explorer Perspective:

To be able to debug within a running Web Dynpro application, you must activate debugging for the server process of the J2EE Engine. You activate this in the *J2EE Engine* view.

1. If necessary, open the *J2EE Engine* view. To do so, choose *Window* → *Show View* → *Other* and then select *J2EE* → *J2EE Engine*. Choose *OK* to confirm your entries. The system displays a view containing status information about the running J2EE Engine.
2. Expand the tree display fully until you can see the actual server process (for example *server0*).
3. Right-click the server node and then choose *Enable debugging of process* from the context menu.

Result

The server process is stopped and restarted in debugging mode. Only the **ON** value is shown for **Debug Mode**. To display the current status of the server, in the view toolbar, choose *Refresh*. Wait until the server has the status **Running**



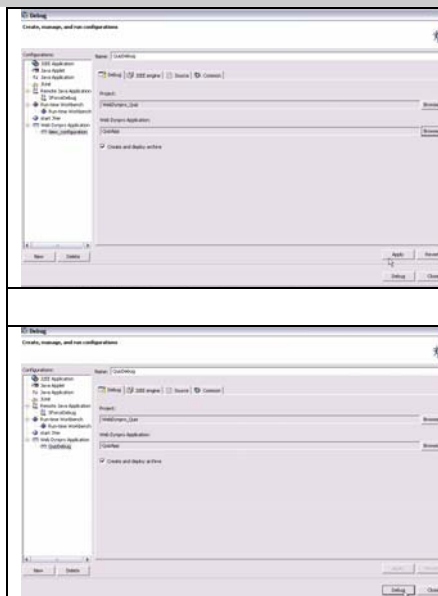
4-1-2 Setting a Breakpoint



1. Open the implementation page of the Quiz Component from the QuizApp application. To do so, in the Web Dynpro Explorer, edit the Quiz Component Controller and go to the Implementation tab page.
2. The Editor displays the source code. Navigate to wdDoInit() method. Right-click on the marker bar (along the left edge of the editor area) frame next to line of code to open the context menu and choose
3. Add Breakpoint. You can also doubleclick in the markerbar to achieve the same results. The breakpoint lines are highlighted with a blue dot.

A suggested breakpoint would be at line
`wdContext.nodeQuizData().bind(questions);`

4-1-3 Defining a Debug Configuration and Starting the Debug Mode

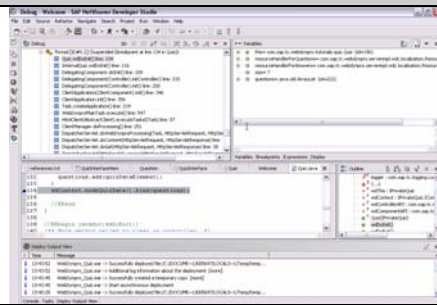


To start the Web Dynpro application in the debugger, you require a *launch configuration*.

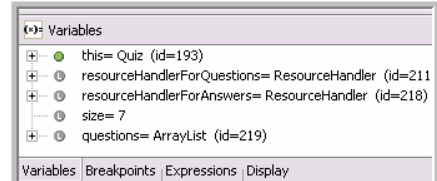
1. Choose **Run** → **Debug...** in the main menu.
2. In the list of possible configurations, select **Web Dynpro Application** and then choose **New**.
3. Under **Name**, enter **QuizDebug** as the name of the configuration.
4. Choose **Browse...** next to the **Project** field. Next, select the **WebDynpro_Quiz** project and confirm with **OK**.
5. Choose **Browse...** next to the **Web Dynpro Application** field. Next, select the **QuizApp** and confirm with **OK**.
6. If the **QuizApp** application to be debugged has not yet been deployed on the server, select the **Create and deploy archive** checkbox.
7. [Optional] To select the server that you want to use for the debugging procedure, choose the **J2EE engine** tab page.
8. The configuration is now complete and you can start the debugger.
9. To start the debugger, choose **Debug**.

The SAP NetWeaver Studio automatically switches to the debug perspective. The Web application is started in an external Browser. If the application appears that it can no longer be executed, you should switch back to the SAP NetWeaver Developer Studio, you will see that the application was stopped at the breakpoint and can now be analyzed.

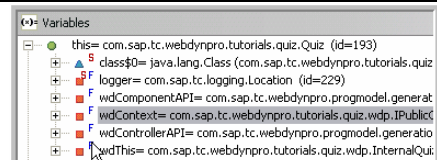
4-1-4 Changing the Web Dynpro Component Context



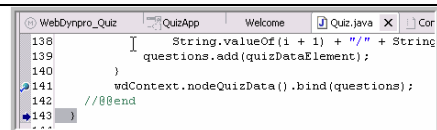
Switch from the browser session to the Debug Perspective.
The program execution should be stopped at your
breakpoint.



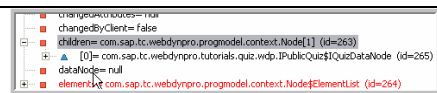
Go to the Variables View in the upper right



- 1) Expand the *this* node
- 2) Expand *wdContext* node



From the editor view, *Step over* your breakpoint. The *wdContext* is now populated.



From the Variables View,
Drilldown to the *children*
Expand this node

```
com.sap.tc.webdynpro.progmodel.context.Node[]
```



Expand the elements node

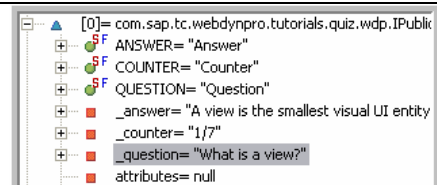
com.sap.tc.webdynpro.progmodel.context.Node\$ElementList

Expand the elements node

```
java.util.ArrayList
```

Expand the elementData node

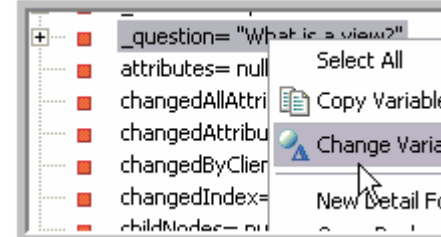
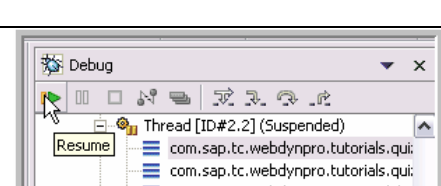
```
java.lang.Object[]
```



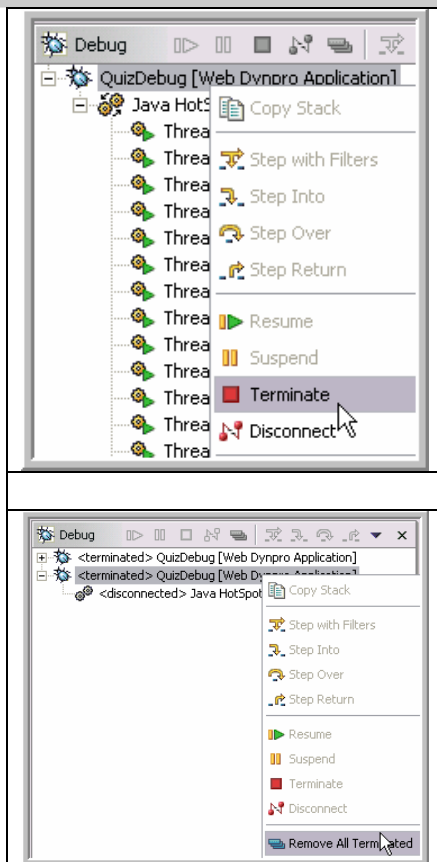
Expand the QuizDataElement Node

com.sap.tc.webdynpro.tutorials.quiz.wdp.IPublicQuiz\$IQuizDataElement

Click on the member variable `_question`

	
	<p>Click on Resume Result - The QuizApp displays the Q&A that you had entered.</p>

4-1-5 Terminate Debugging



Proceed as follows:

...

1. In the *Debug View*, call the context menu for the top node (*QuizDebug[Web Dynpro Application]*). Choose *Terminate*.

2. Call the context menu again and then choose *Remove All Terminated Launches*. The Remove All Terminated Launches tool button clears the Debug view of threads that have been Terminated.

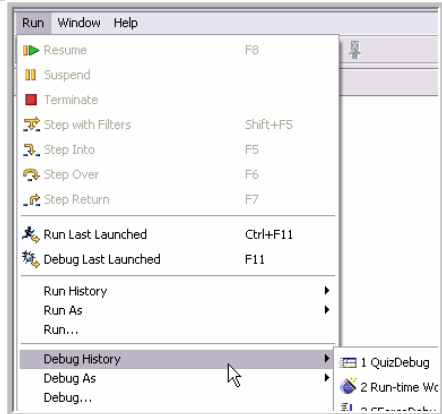
4-2 HotSwap code (Optional Section)

4-2-1 Set a Breakpoint

```
164 //End
165 public void onActionExitPressed(com.sap.tc.webdynpro.progmodel.api.TWCustomEvent wdEvent)
166 {
167     //Begin onActionExitPressed(ServerEvent)
168     wdThis.wdSetQuitInterfaceViewController().wdFirePlugGetUrl("http://www.sdn.sap.com");
169     //End
170 }
```

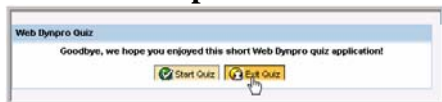
Place a new breakpoint in the Welcome View Implementation of the method `onActionExitPressed`

4-2-2 Resume Debugging

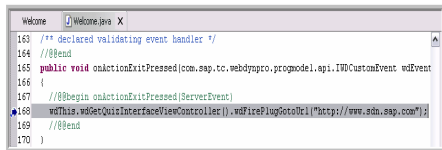


You can restart debugging of the *QuizApp* application by choosing **Run -> Debug History** and select *QuizDebug*.

4-2-3 HotSwap



Run the QuizApp (you may have to resume past the 1st breakpoint). Click on *Exit Quiz* button.



- 1) Toggle over to the Debug perspective. Your application will be suspended at the breakpoint in the `onActionExitPressed` method.
- 2) Change to code to go to a different URL (ex: <http://www.sdn.sap.com>) or fire a different plug. Be creative.
Save and click *Resume*



Result: Your application should reflect your code changes