

Debugging a Web Dynpro Application



SAP NetWeaver 04



Copyright

© Copyright 2004 SAP AG. All rights reserved.

No part of this publication may be reproduced or transmitted in any form or for any purpose without the express permission of SAP AG. The information contained herein may be changed without prior notice.

Some software products marketed by SAP AG and its distributors contain proprietary software components of other software vendors.

Microsoft, Windows, Outlook, and PowerPoint are registered trademarks of Microsoft Corporation.

IBM, DB2, DB2 Universal Database, OS/2, Parallel Sysplex, MVS/ESA, AIX, S/390, AS/400, OS/390, OS/400, iSeries, pSeries, xSeries, zSeries, z/OS, AFP, Intelligent Miner, WebSphere, Netfinity, Tivoli, and Informix are trademarks or registered trademarks of IBM Corporation in the United States and/or other countries.

Oracle is a registered trademark of Oracle Corporation.

UNIX, X/Open, OSF/1, and Motif are registered trademarks of the Open Group.

Citrix, ICA, Program Neighborhood, MetaFrame, WinFrame, VideoFrame, and MultiWin are trademarks or registered trademarks of Citrix Systems, Inc.

HTML, XML, XHTML and W3C are trademarks or registered trademarks of W3C®, World Wide Web Consortium, Massachusetts Institute of Technology.

Java is a registered trademark of Sun Microsystems, Inc.

JavaScript is a registered trademark of Sun Microsystems, Inc., used under license for technology invented and implemented by Netscape.

MaxDB is a trademark of MySQL AB, Sweden.

SAP, R/3, mySAP, mySAP.com, xApps, xApp, SAP NetWeaver, and other SAP products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of SAP AG in Germany and in several other countries all over the world. All other product and service names mentioned are the trademarks of their respective companies. Data contained in this document serves informational purposes only. National product specifications may vary.

These materials are subject to change without notice. These materials are provided by SAP AG and its affiliated companies ("SAP Group") for informational purposes only, without representation or warranty of any kind, and SAP Group shall not be liable for errors or omissions with respect to the materials. The only warranties for SAP Group products and services are those that are set forth in the express warranty statements accompanying such products and services, if any. Nothing herein should be construed as constituting an additional warranty.

Icons in Body Text

Icon	Meaning
	Caution
	Example
	Note
	Recommendation
	Syntax

Additional icons are used in SAP Library documentation to help you identify different types of information at a glance. For more information, see *Help on Help* → *General Information Classes and Information Classes for Business Information Warehouse* on the first page of any version of *SAP Library*.

Typographic Conventions

Type Style	Description
<i>Example text</i>	Words or characters quoted from the screen. These include field names, screen titles, pushbuttons labels, menu names, menu paths, and menu options. Cross-references to other documentation.
Example text	Emphasized words or phrases in body text, graphic titles, and table titles.
EXAMPLE TEXT	Technical names of system objects. These include report names, program names, transaction codes, table names, and key concepts of a programming language when they are surrounded by body text, for example, SELECT and INCLUDE.
Example text	Output on the screen. This includes file and directory names and their paths, messages, names of variables and parameters, source text, and names of installation, upgrade and database tools.
Example text	Exact user entry. These are words or characters that you enter in the system exactly as they appear in the documentation.
<Example text>	Variable user entry. Angle brackets indicate that you replace these words and characters with appropriate entries to make entries in the system.
EXAMPLE TEXT	Keys on the keyboard, for example, F2 or ENTER.

Debugging a Web Dynpro Application	5
Activating Debugging	5
Starting a Debug Session	8



Debugging a Web Dynpro Application

The SAP NetWeaver Developer Studio provides support when debugging your Web Dynpro applications. By making special settings for the server process of the J2EE Engine and setting breakpoints, you can interrupt the execution of your application and go through it step-by-step.

The following documentation begins by explaining the steps that you need to make in the Developer Studio to activate the debug mode. Next, execute debugging step-by-step using the example of the *Welcome* Web Dynpro application.



The *Welcome* application is part of the standard set of example applications contained in the example directory of the Developer Studio (`.../SAP/JDT/eclipse/examples/WebDynpro_Welcome.zip`). See also: [Creating Your First Web Dynpro Application \[External\]](#).

Next step:

[Activating Debugging \[Page 5\]](#)



Activating Debugging

Prerequisites

- You have launched the SAP NetWeaver Developer Studio.
- You are sure that the SAP J2EE Engine has been launched.
- You can only debug in non-productive server nodes (*Productive Use* has the value *NO*).

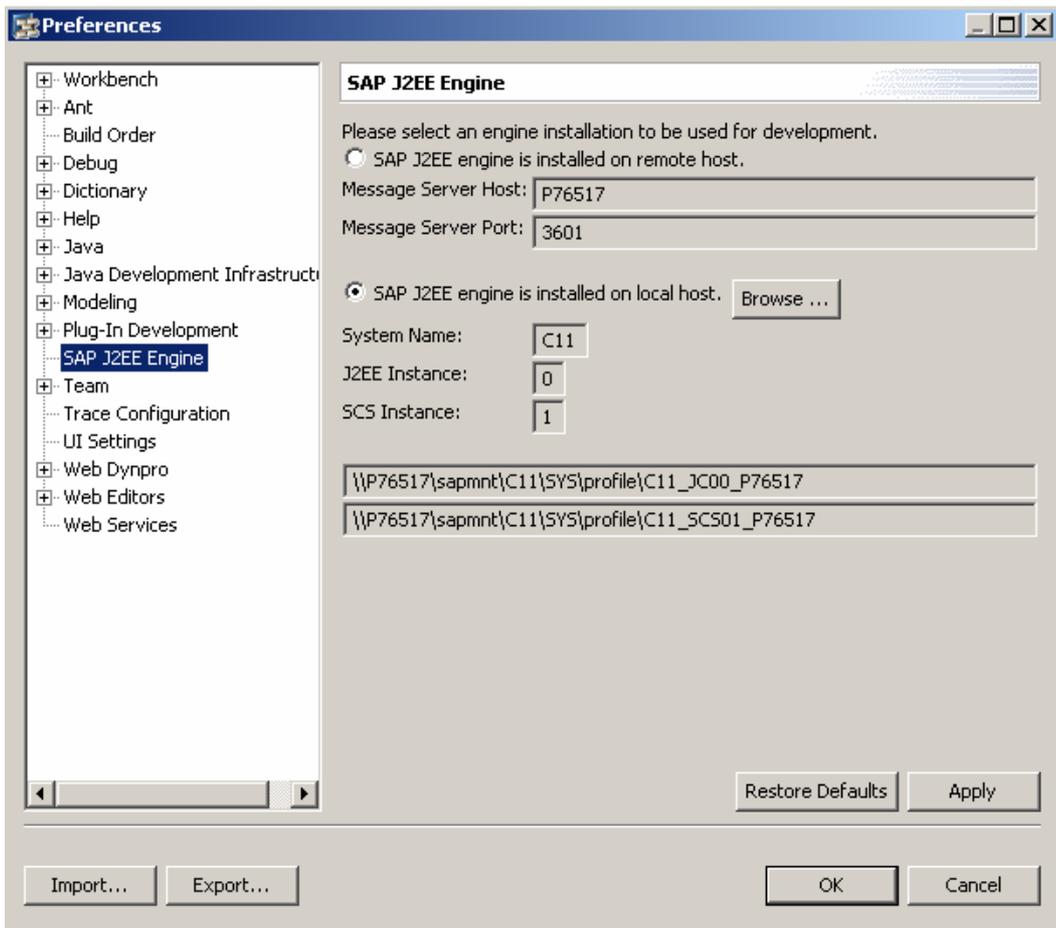
Procedure

Checking the Server Settings in the Developer Studio

1. Choose *Window* → *Preferences*.
2. In the list Preferences, select the node *SAP J2EE Engine*.
3. If necessary, select the appropriate installation option.

Choose the *SAP J2EE engine is installed on remote host* option if you are deploying and debugging the application remotely (LAN scenario). For this case, enter a *Message Server Host* and the *Message Server Port*.

If the SAP J2EE Engine is installed on the same PC as the Developer Studio, choose the *SAP J2EE engine is installed on local host* option. Then choose *Browse...* and assign the relevant system.



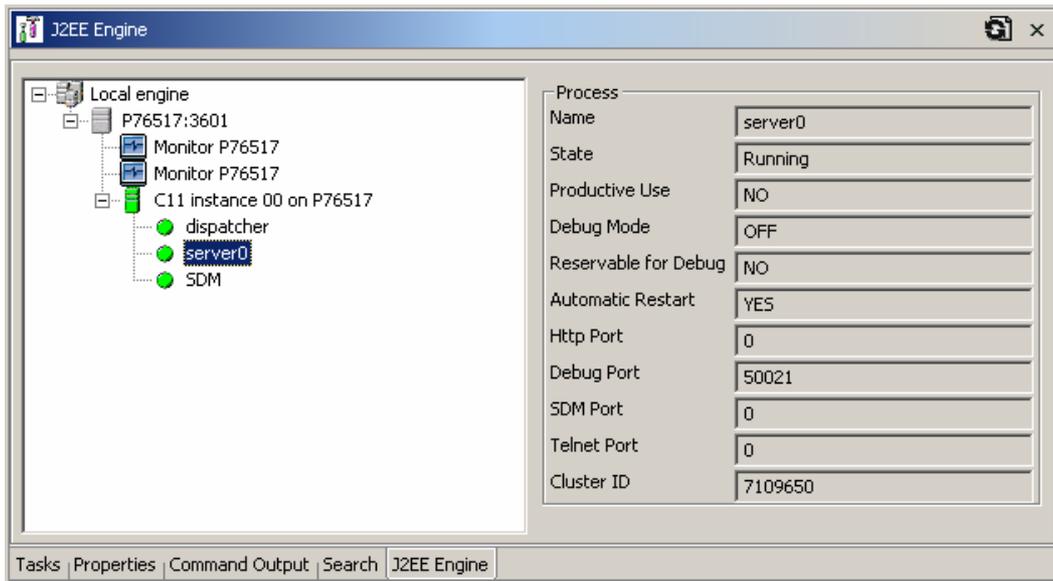
Switching Server Nodes to Debug Mode

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*).



- Right-click the server node and then choose the appropriate function from the context menu.

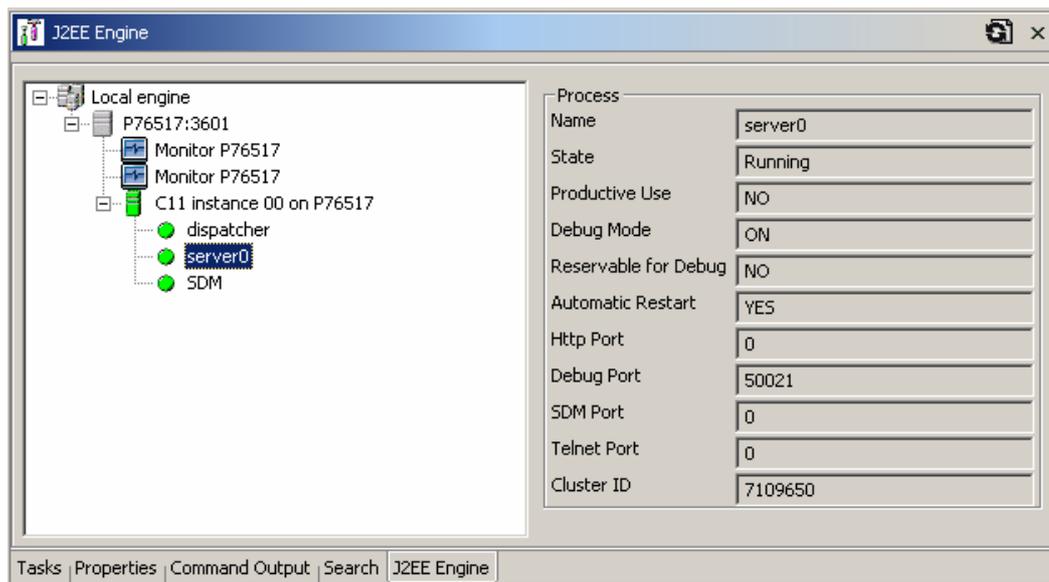
*Enable debugging of process in a **Single Server Configuration***

or

*Reserve process for debugging in a **Cluster Configuration***

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**.



Next step:

[Starting a Debug Session \[Page 8\]](#)



Starting a Debug Session

This step explains how you can interrupt processing of a Web Dynpro application by adding a breakpoint. This then enables you to analyze the status of the application at runtime.

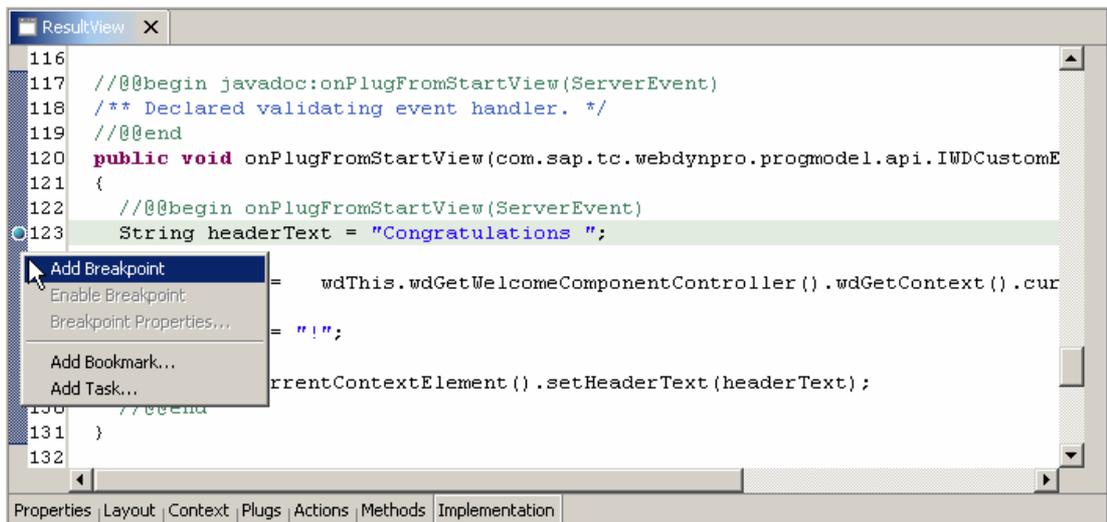
Prerequisites

- Debugging is activated – that is, the server process has been stopped and restarted in debugging mode.
- You have imported the *Welcome* application to the Workspace of the Developer Studio and have opened it in the *Web Dynpro Explorer*.

Procedure

Setting a Breakpoint

1. Open the implementation page of the *ResultView* from the *Welcome* application. To do so, in the *Web Dynpro Explorer*, double-click *Welcome* → *Web Dynpro* → *Web Dynpro Components* → *Welcome Component* → *Views* → *ResultView* and then choose the Implementation tab page.
2. Display the source code and navigate to the beginning of the *onPlugFromStartView* method. Right-click the left bar of the editor frame above the appropriate line (see below) to open the context menu and choose *Add Breakpoint*.



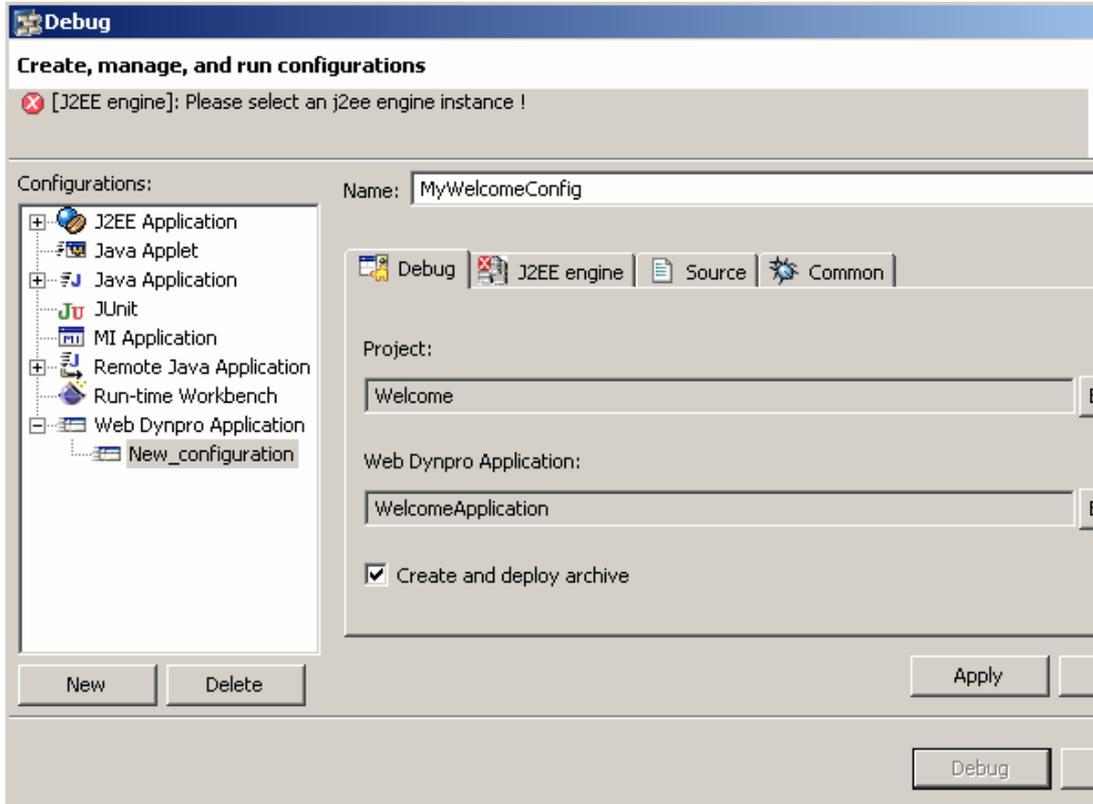
The breakpoint lines are highlighted with a blue dot.

Defining a Debug Configuration and Starting the Debug Mode

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

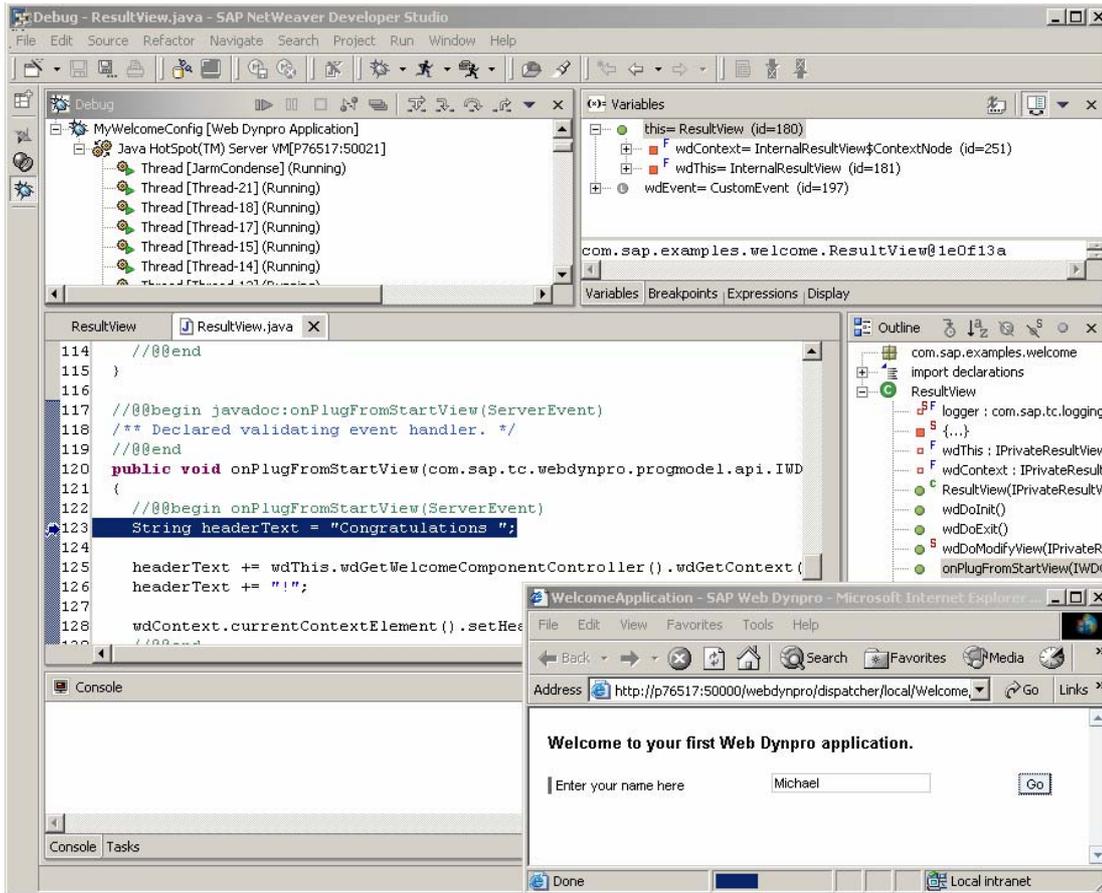
1. Choose *Run* → *Debug...* in the main menu.
2. In the list of possible configurations, select *Web Dynpro Application* and then choose *New*.

3. Under *Name*, enter **MyWelcomeConfig** as the name of the configuration.
4. Choose *Browse...* next to the *Project* field. Next, select the *Welcome* project and confirm with *OK*.
5. Choose *Browse...* next to the *Web Dynpro Application* field. Next, select the *WelcomeApplication* and confirm with *OK*.
6. If the *Welcome* application to be debugged has not yet been deployed on the server, select the *Create and deploy archive* checkbox.



7. 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. When you enter a name, for example **michael** and choose *Go*, it appears that it can no longer be executed. However, if you change back to the SAP NetWeaver Developer Studio, you will see that the application was stopped at the breakpoint and can now be analyzed.



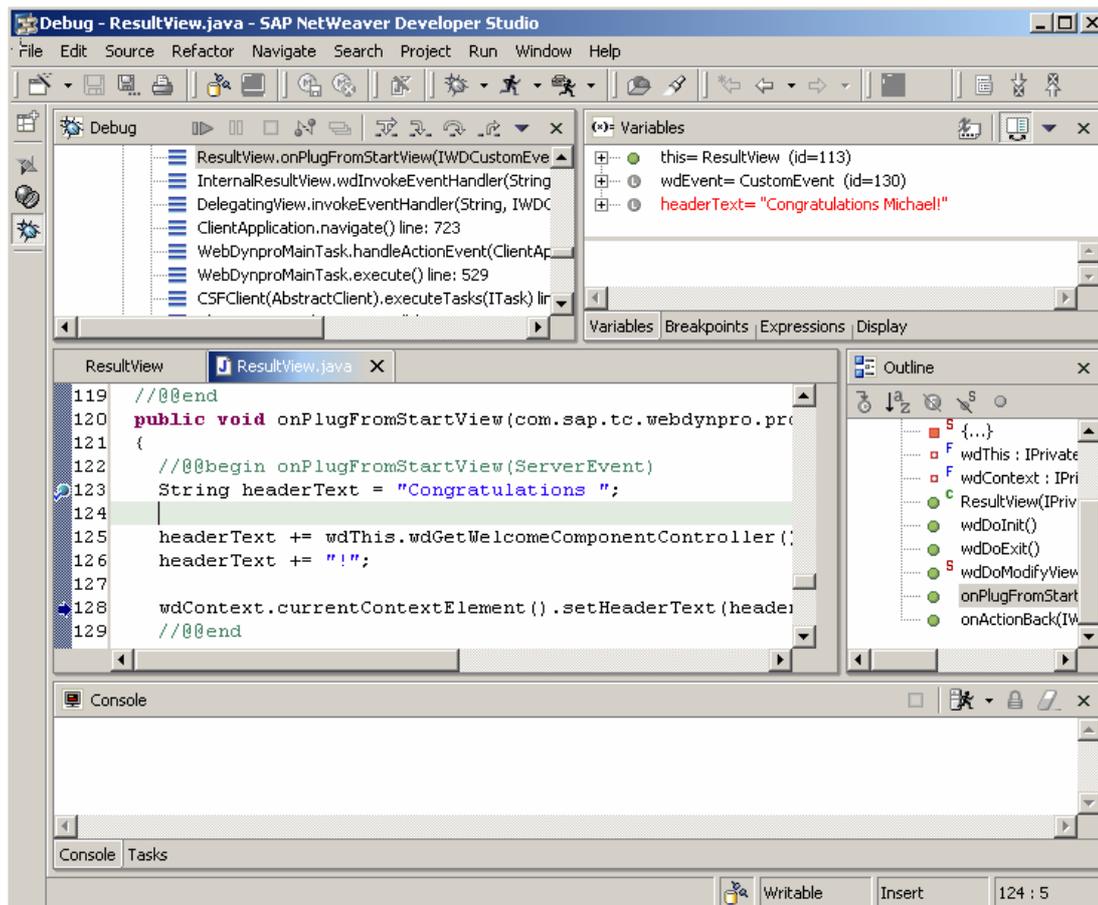
Analyzing the Example Application

There are various different options for further processing on a step-by-step basis:

Key	Description
F5: <i>Step Into</i>	Jumps to the next statement
F6: <i>Step Over</i>	The next command is executed without jumping to the current statement
F7: <i>Step Return</i>	If you previously chose F5, you can choose F7 to cancel the debugging of the current command
F8: <i>Resume</i>	The application exits debug mode and continues with execution

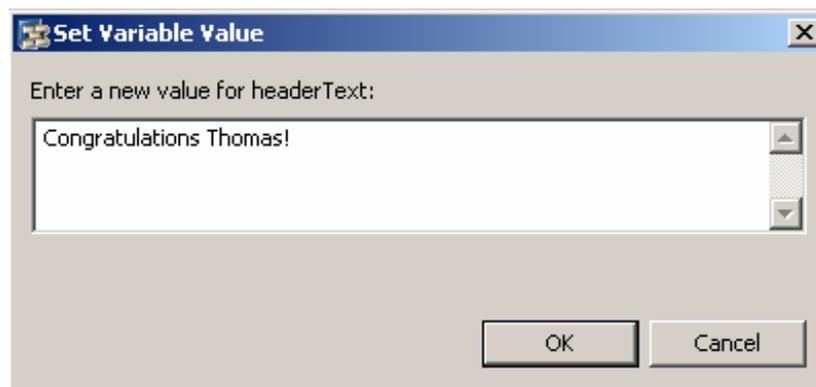
1. Use F6 to execute all statements before the line `'wdContext.currentContextElement().setHeaderText(headerText)'` within the `onPlugFromStartView` method and observe the variable contents in the *Variables* window of the debug perspective.

The local `headerText` variable changes your value in accordance with the executed program lines.

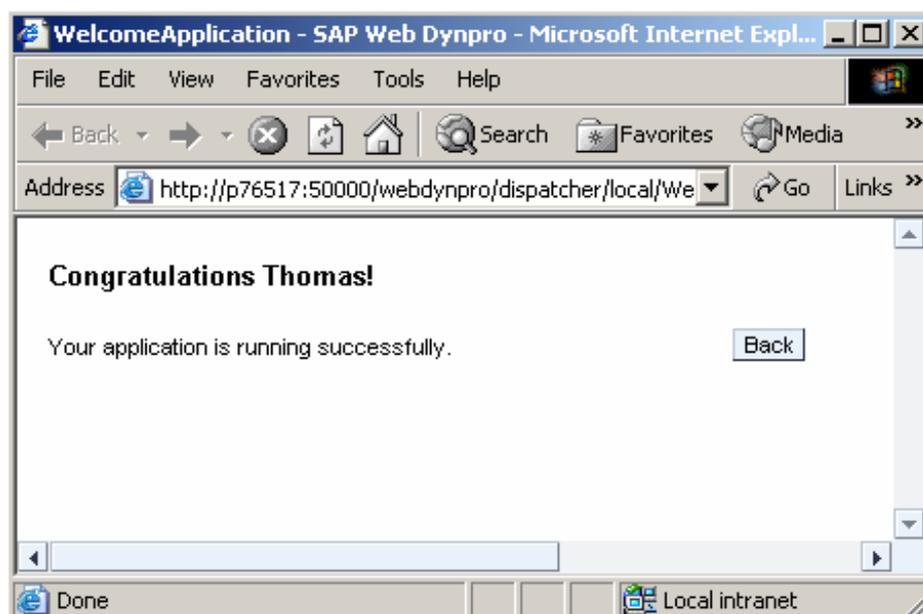


The debugging mode enables you to dynamically change the variable contents.

2. Double-click the *headerText* variable in the *Variables* window of the debug perspective.
3. Replace the name **Michael** in the input field with **Thomas** and choose **OK**.



4. Choose **F8** to continue the execution of the application. The system then displays the following *ResultView* that displays the changed value of the variables.



Terminating Debugging



If you want to exit debugging, you must terminate the threads in the SAP NetWeaver Developer Studio.

Proceed as follows:

1. In the *Debug View*, call the context menu for the top node (*MyWelcomeConfig[Web Dynpro Application]*).
2. In the context menu that appears, choose *Disconnect*.

