Debugging of Web Dynpro for Java Applications in CE7.1

Applies to:

Web Dynpro for Java applications for the SAP NetWeaver Composition Environment 7.1

For more information, visit the User Interface Technology homepage.

Summary

This document explains how to start debugging Web Dynpro Applications in CE7.1.It also explains about Context Debugging which was introduced in Composition Environment 7.1

Author: Sudhir Gorantla

Company: HCL Technologies Limited

Created on: 03 November 2008

Author Bio

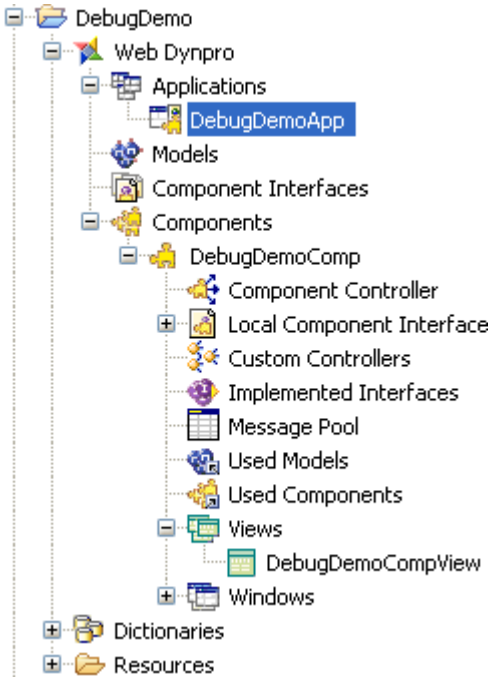
Sudhir Gorantla is an Application Developer working in HCL Technologies Limited on SAP Netweaver Technology (CAF, Web Dynpro for JAVA and GP).

Table of Contents

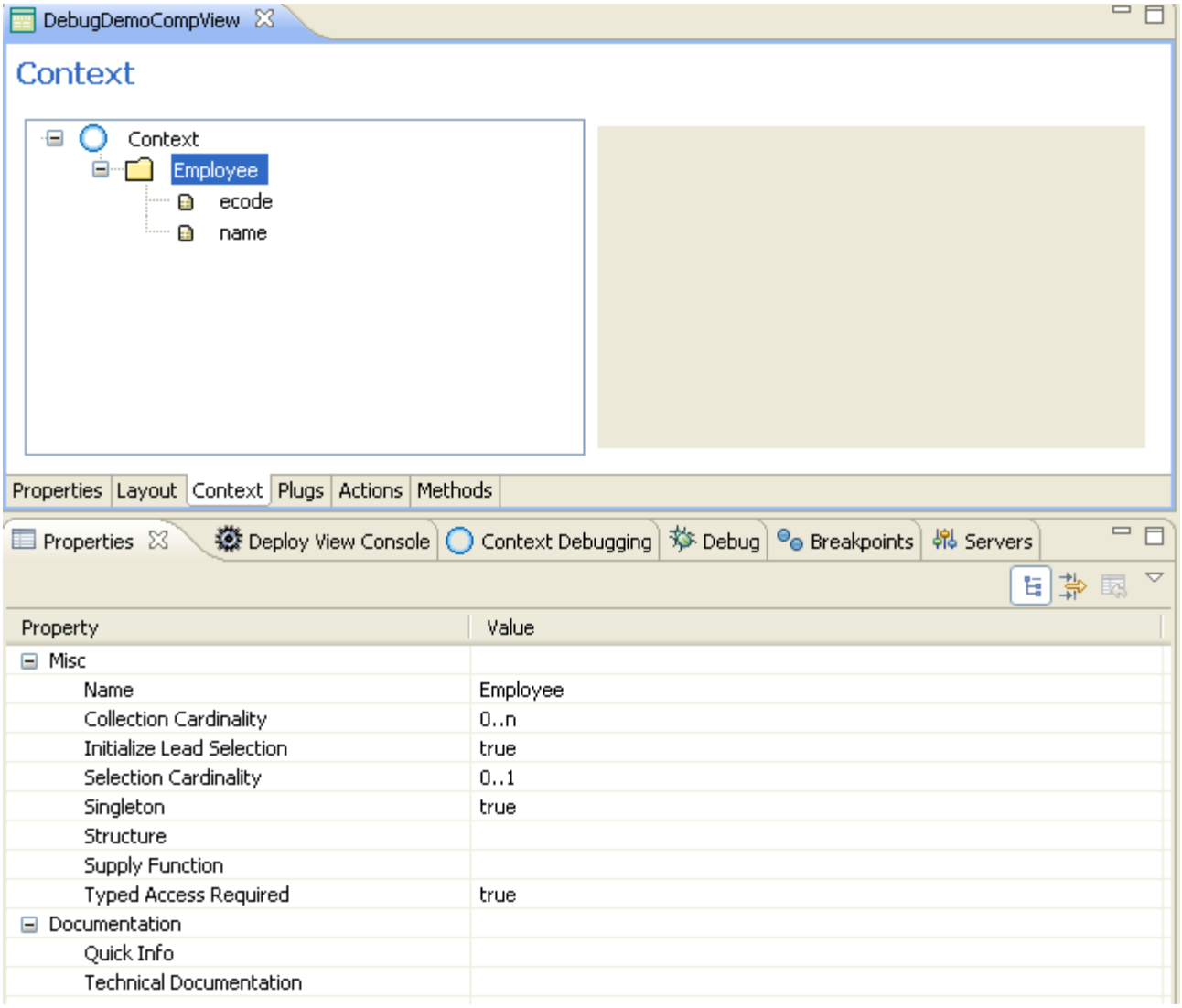
Procedure

Create a Project

Create a webdynpro Project with name “DebugDemo”.Create an application with name “DebugDemoApp” which uses the Component “DebugDemoComp”.The project structure is shown in the below figure



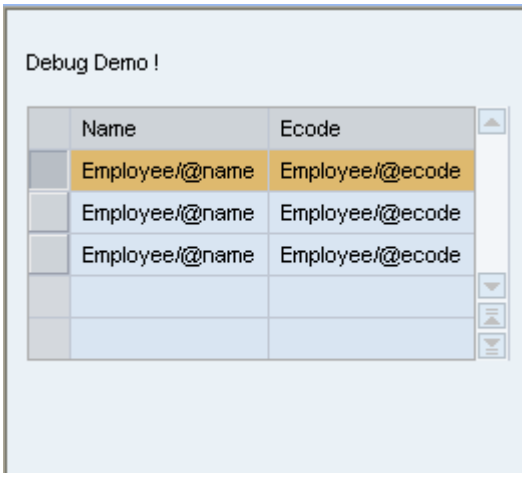
In the Context of “DebugDemoCompView” controller create the below shown node structure



In the wdDoInit() method of "DebugDemoCompView" controller write the following code

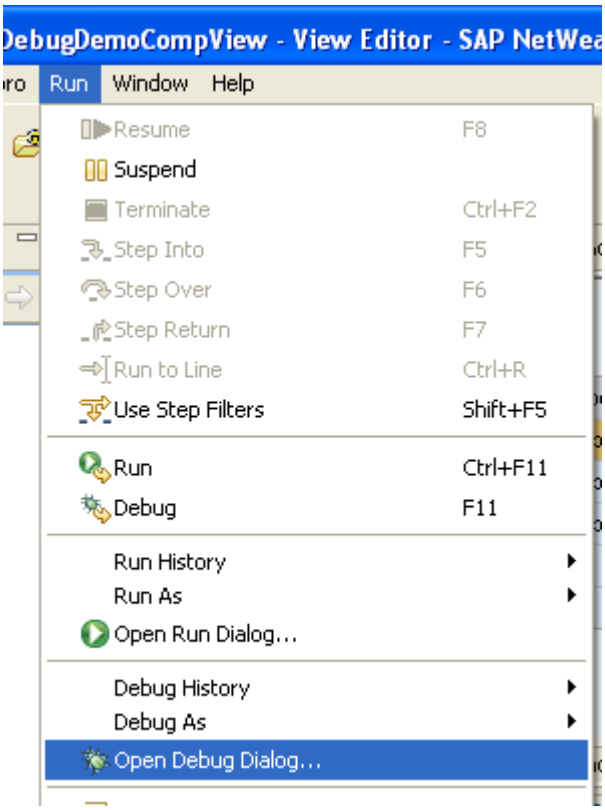
```
public void wdDoInit()
{
  //@@begin wdDoInit()
      for (int i = 0; i < 10; i++) {
           IEmployeeElement element = wdContext.nodeEmployee()
                   .createEmployeeElement();
           element.setEcode("Ecode-"+i);
           element.setName("Name-"+i);
           wdContext.nodeEmployee().addElement(element);
  }
  //@@end
}
```

Create the below shown layout in the "DebugDemoCompView"

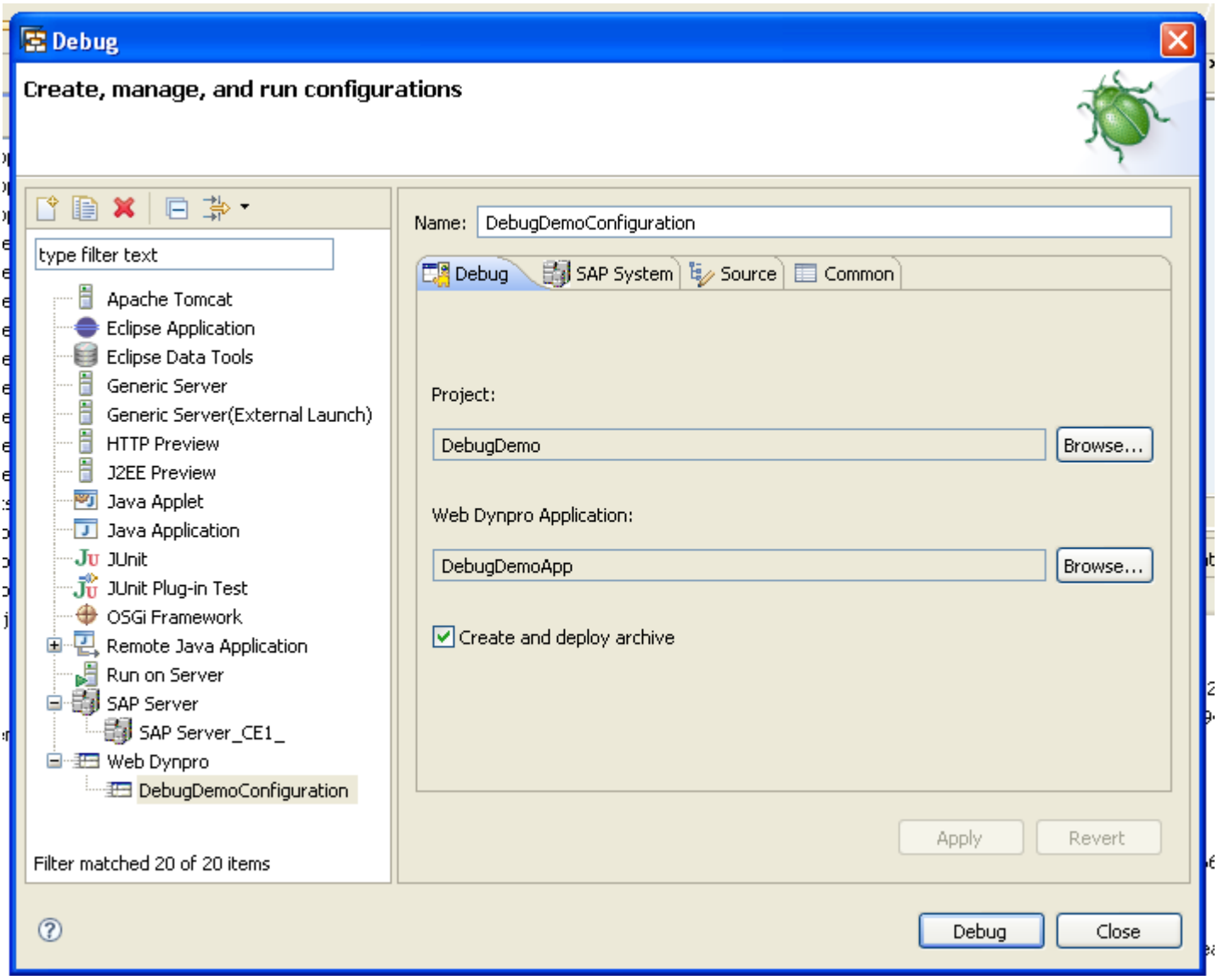


Launch Debugging

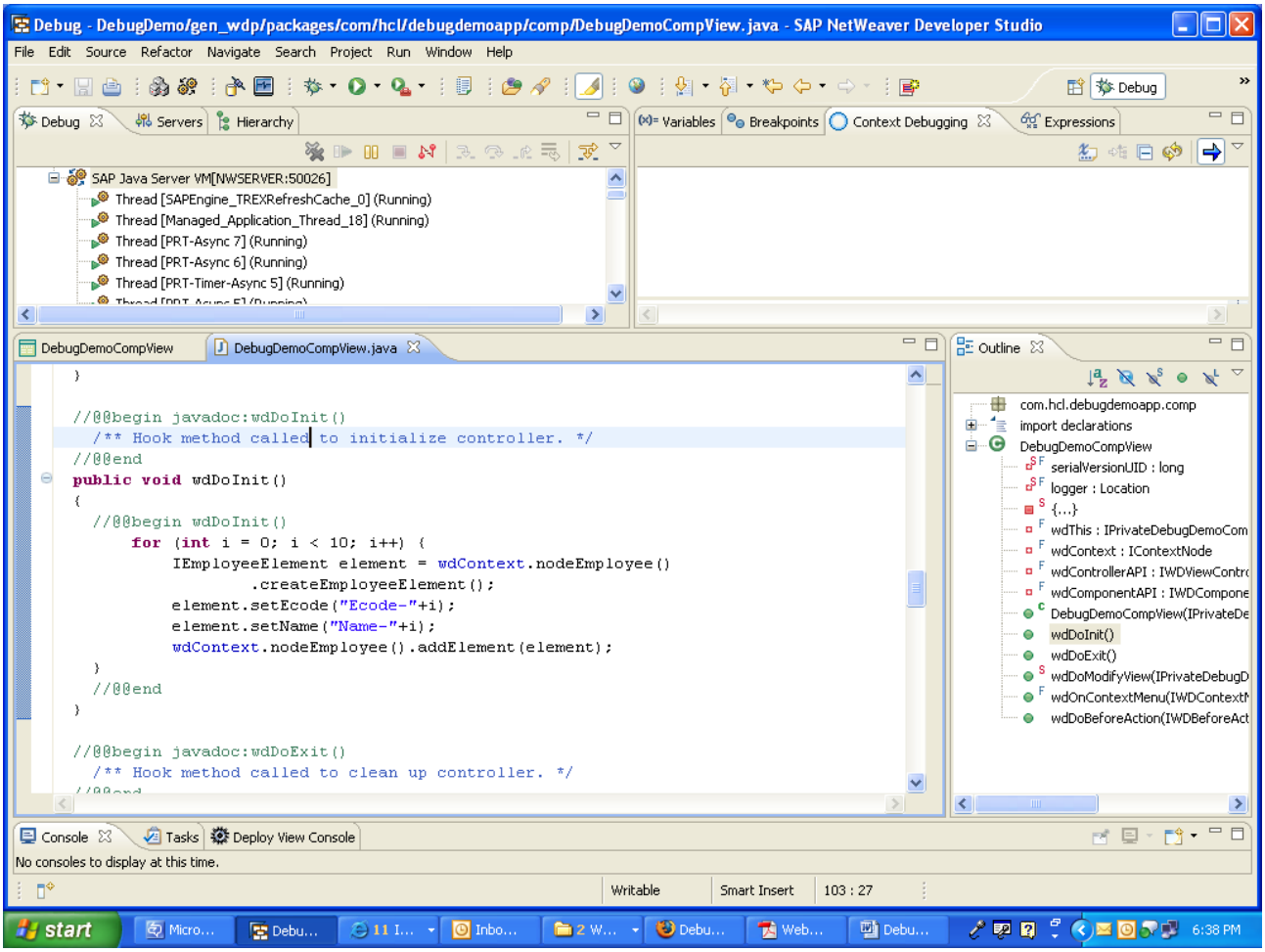
Navigate to Run -> Open Debug Dialog



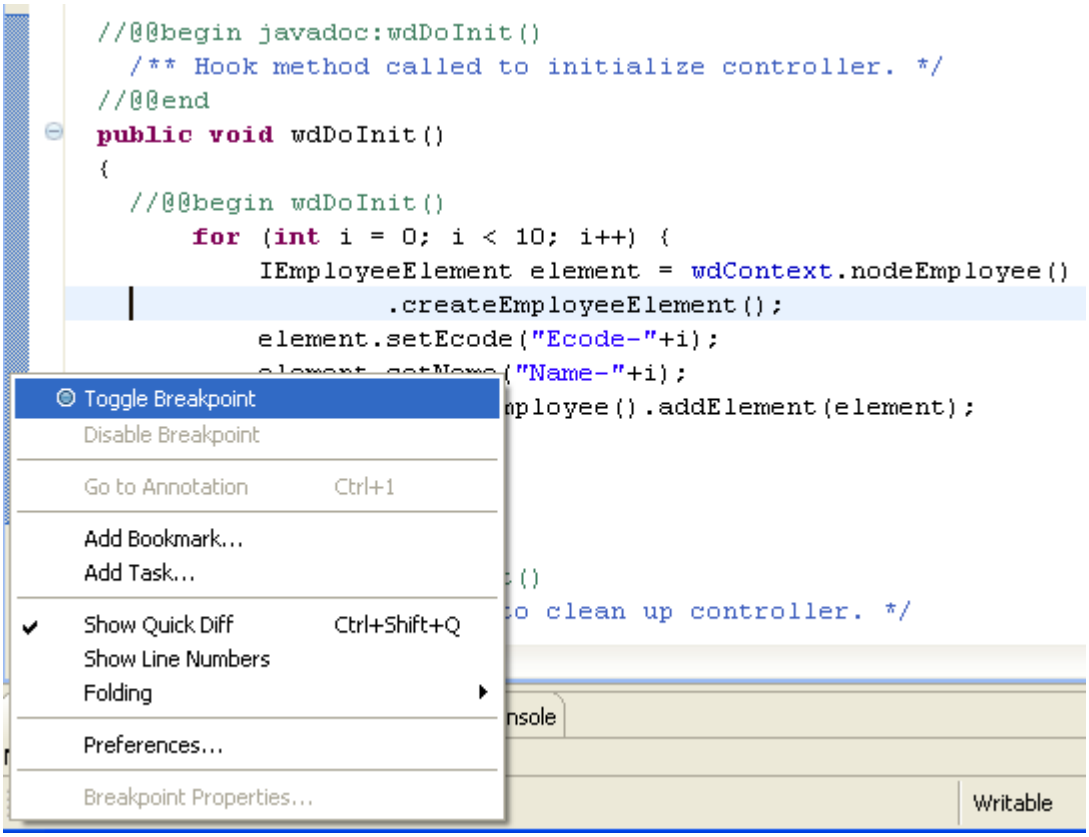
Double click on the “Web Dynpro” which is present at the bottom of the dialog box to create New_configuration.Give the name as “DebugDemoConfiguration” and choose the project “DebugDemo”.The application “DebugDemoApp” appears automatically in the Web Dynpro Application.



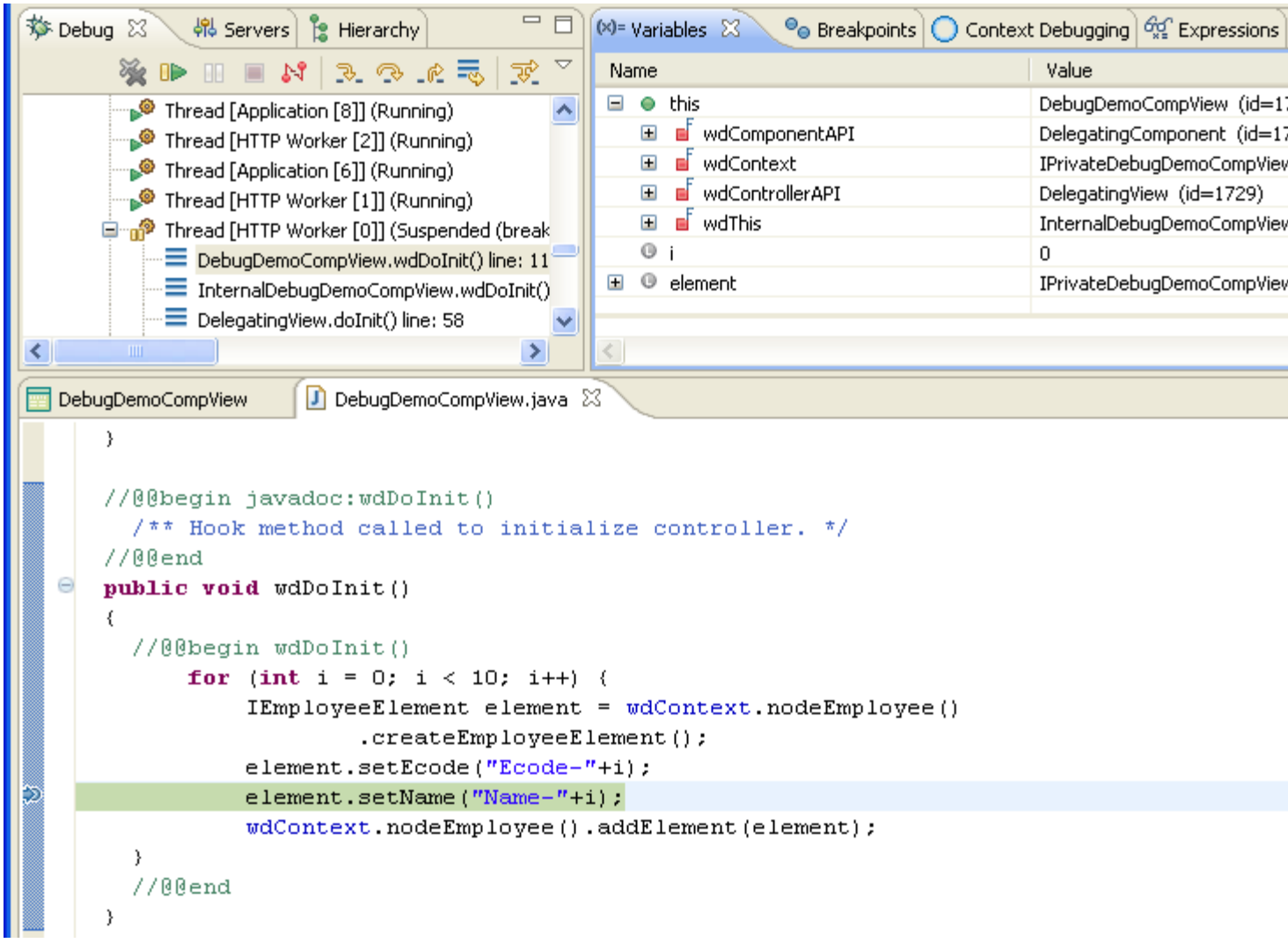
Click on “Apply” and then on “Debug”. Now go to Window->Open Perspective -> Other -> Debug. Also navigate to Window -> Show View and open the View’s “Variables”,” Break Points” and “Expressions” .Now NWDS should look like this



Right-click on the left bar of the editor frame besides the appropriate line (see below) to open the context menu and choose Toogle *Breakpoint*. Alternatively you can double click on the left bar of the editor frame at the appropriate line to set the brake point.

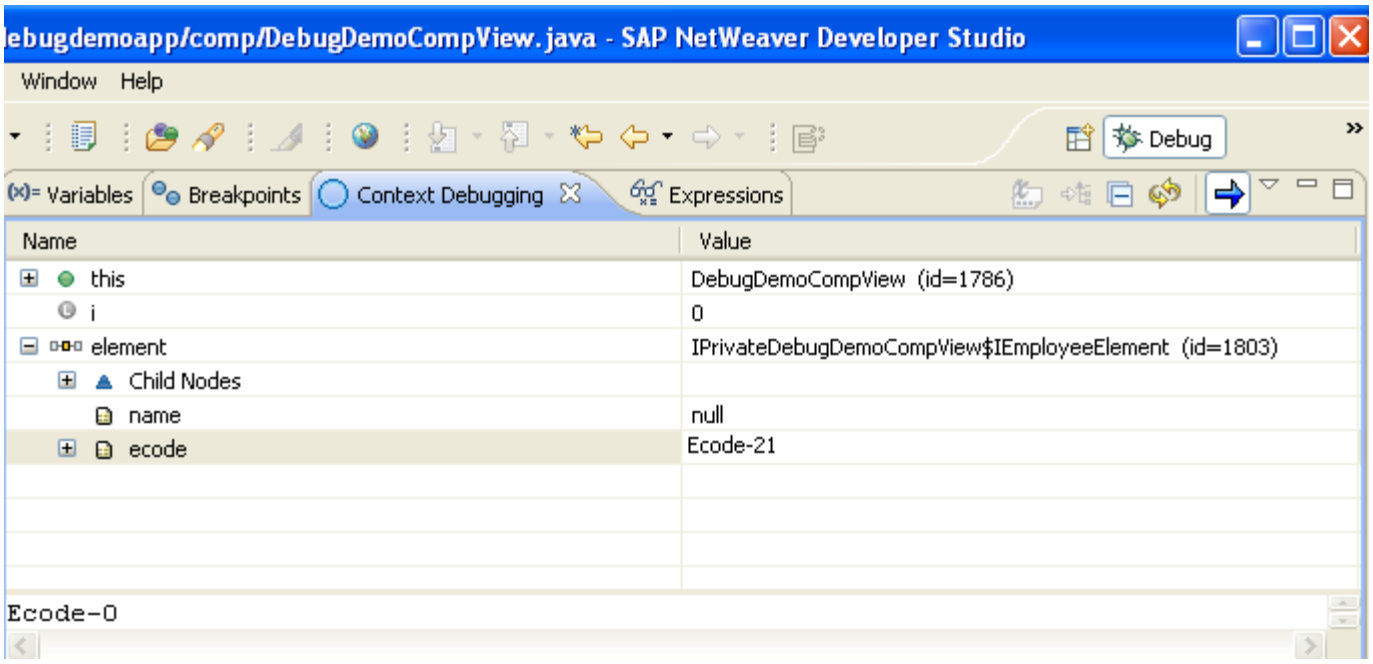


Now run the application. The execution will stop at the break point .Now observe already opened different views “Variables”,”Breakpoints”,”Context Debugging”.

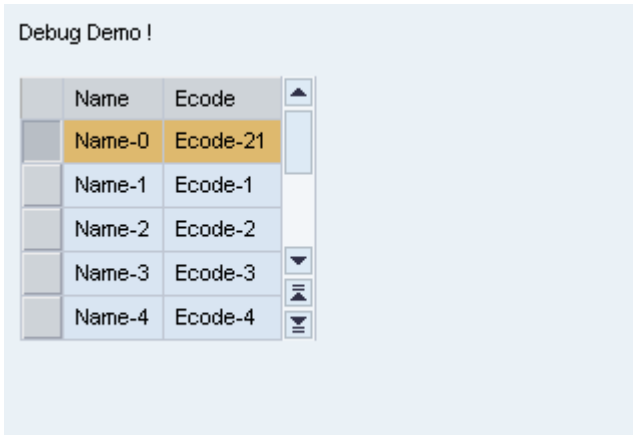


Context Debugging

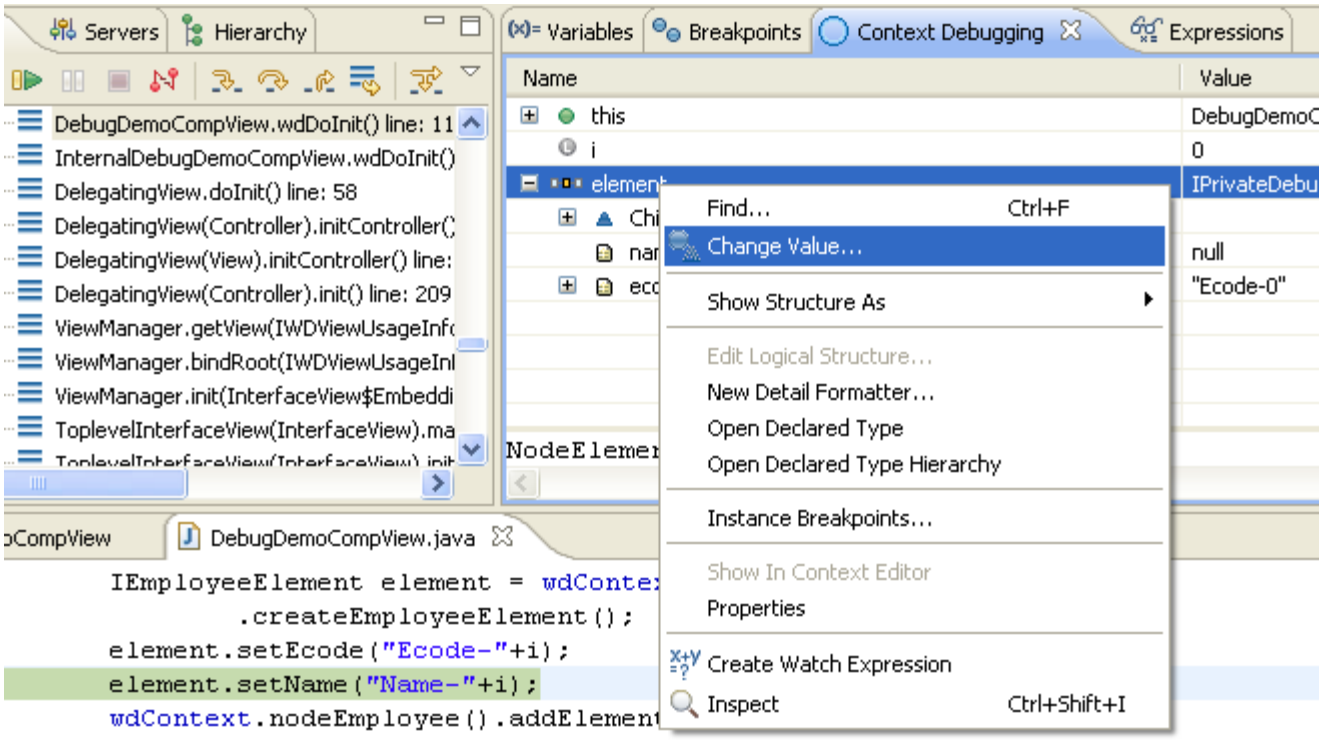
Observe the “Context Debugging” View. The context values can modified at run time. To do this, expand “element” and change the Ecode value from “Ecode-0” to “Ecode-21” and finish the execution.



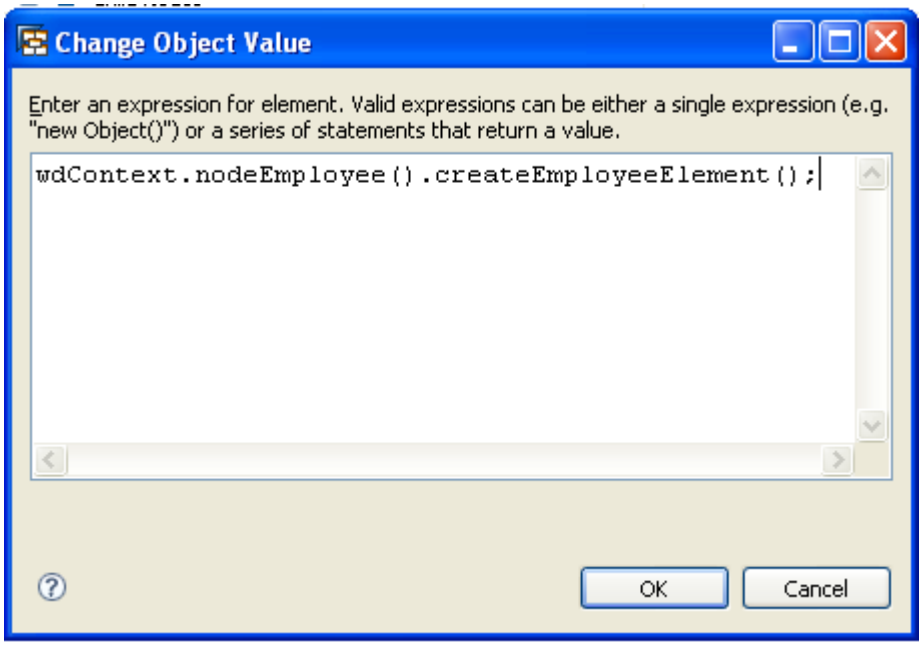
The out put will look like this. The first element's Ecode has been changed from "Ecode-0" to "Ecode-21" at runtime.



And also we can create a completely new Employee Element at runtime and assign it to variable "element". To do this, set the break point at the line "element.setName("Name-"+i)" and debug the application. Now open "Context Debugging" view, select "element" and choose "Change Value" as shown below.



Enter the following code in the Change Object Value Dialog box

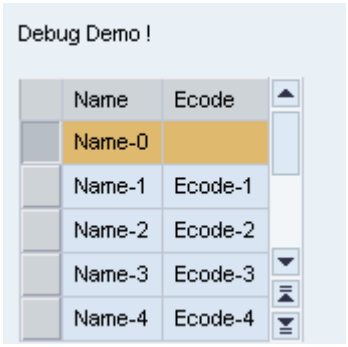


Now your code is changed at runtime as shown below for the first iteration of the loop

```
//@@begin javadoc:wdDoInit()
  /** Hook method called to initialize controller. */
//@@end
public void wdDoInit()
{
  //@@begin wdDoInit()
      for (int i = 0; i < 10; i++) {
          IEmployeeElement element = wdContext.nodeEmployee()
                  .createEmployeeElement();
          element.setEcode("Ecode-"+i);
          element=wdContext.nodeEmployee().createEmployeeElement();
          element.setName("Name-"+i);
          wdContext.nodeEmployee().addElement(element);
  }
  //@@end
}
```

After setting the Ecode to the variable “element”, we are creating a new “EmployeeElement” and assigning it to variable “element”. Therefore in the result the first Employee Element doesn’t have Ecode.

The output is as shown below



Related Content

Highlights of Web Dynpro for Java in SAP Netweaver Composition Environment 7.1

https://www.sdn.sap.com/irj/sdn/go/portal/prtroot/docs/library/uuid/b0e03d24-a21d-2a10-9ea6-f13e6e7ec9e7

For more information, visit the User Interface Technology homepage.

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.