

Applies To:

SAP NetWeaver Developer Studio 6.40 and J2EE Engine 6.40.

By: Dobrinka Stefanova

Company: SAP AG

Date: 18 July 2005

J2EE Toolset: How Do I Love Thee? Let Me Count the Ways...

Dobrinka Stefanova explains why the SAP NetWeaver Developer Studio is our favorite environment for J2EE development.

Maybe you've noticed that application developers are a most diverse group and can be peculiar, picky, particular or just a plain nuisance? We certainly have! But beside the numerous things one can say of developers, let me tell you that a professional in need is a professional indeed. That's why this article introduces the J2EE toolset in the SAP NetWeaver Developer Studio - an integrated development environment (IDE) for the most demanding of developers that will facilitate your daily work, beginners and experts alike.

Among the vast number of application developers, you will find that each and every one has different requirements in terms of an IDE. Some prefer IDEs with rich functionality, others would use only the basic functionality, and others still would prefer not to deal with any kind of IDE whatsoever, and to use basic text editors instead.

The SAP NetWeaver Developer Studio is based on the Eclipse platform. Now, if you are an experienced Java developer, bells will already be ringing. Eclipse has proven to be a wonderful Java development environment for its stability, speed, user-friendliness, agility and universality. SAP has joined the growing number of companies developing plug-ins for Eclipse and provides a convenient toolset for diverse J2EE application development. SAP plug-ins are tightly integrated with the Eclipse GUI environment. The J2EE toolset is fully J2EE 1.3 compliant and covers the full development lifecycle of your applications.

The SAP NetWeaver Developer Studio can surprise you with the multitude and variety of features it provides, and at the same time with its speed and simplicity of use. With the SAP NetWeaver Developer Studio, not only can you make the most of J2EE development, you can also do it faster and better. Proficient developers will discover that many of their boring tasks are automated and only a mouse-click away. Beginners will discover that, with the J2EE toolset, creating J2EE components does not require in-depth knowledge of the J2EE specification.

So, regardless of whether you are a complete novice or an expert in J2EE development, read on. The SAP NetWeaver Developer Studio has it all.

Full Development Lifecycle

The SAP NetWeaver Developer Studio provides more than just an environment for writing code. It is an environment supporting the full application development lifecycle – from the design phase to the testing and debugging phase.

A typical scenario for developing J2EE applications through the SAP NetWeaver Developer Studio may look as shown in the figure below:

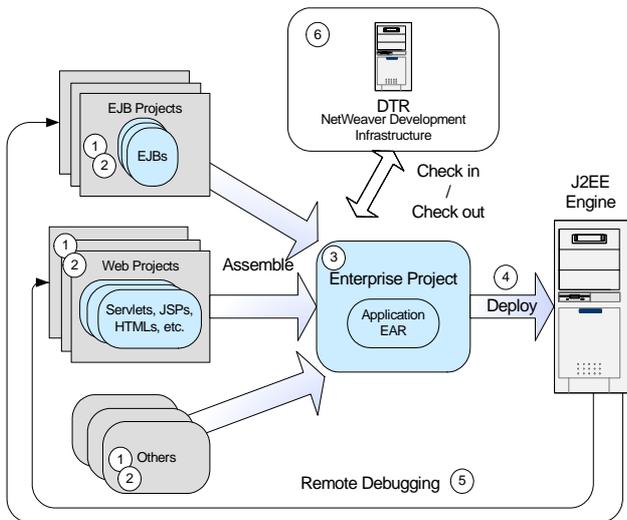


Figure 1: J2EE Application Lifecycle

1) Application design

With the SAP NetWeaver Developer Studio, you can model J2EE projects using UML diagrams. The SAP NetWeaver Developer Studio automatically creates the physical components corresponding to the diagram elements.

2) Development of application components

Development consists of several sub-phases:

- Creating the corresponding J2EE projects
The development of the different application components is disassembled into different J2EE projects. The types of projects are explained later in this article.
- Generating the source code using the wizards, editors, templates and code assistance
The J2EE toolset provides numerous UI components that allow you to avoid working directly with the source code.
- Editing the deployment descriptors using the corresponding multipage editors.

3) Assembling the separate components into a single Enterprise project, and building the application archive (EAR).

4) Deploying the application on the J2EE Engine directly from the SAP NetWeaver Developer Studio.

5) Testing and debugging the application's behavior.

This can be done via a direct debugging connection from the SAP NetWeaver Developer Studio to the J2EE Engine.

6) Checking the components into the Design Time Repository (DTR)

At any time during the development you can check your current work into the DTR of the NetWeaver Development Infrastructure, and check out the latest versions from your team mates.

The J2EE Perspective, Views and Projects

The J2EE toolset supports the application lifecycle with dedicated UI components.

It provides a dedicated perspective consisting of a number of views for J2EE application development. Like the standard Java perspective of Eclipse, the J2EE perspective shows the J2EE projects that are available in the workspace, but focuses on their J2EE-related elements: deployment descriptors, Enterprise JavaBeans (EJB)

components, servlets, JavaServer Pages (JSP), HTMLs, and so on. There is also a separate view showing the state of the J2EE Engine cluster (see Figure 2).

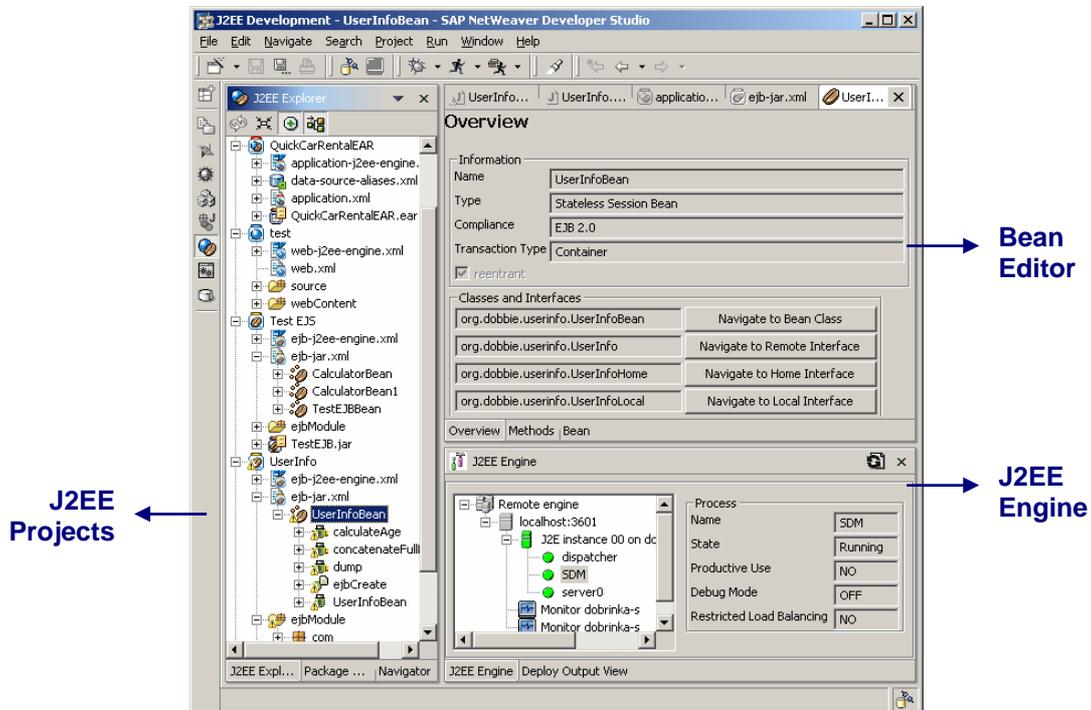


Figure 2: J2EE Perspective

The creation of J2EE applications is dissembled into working with a set of projects:

- Web Module project**
 Dedicated for the development of Web components (servlets, JSPs, HTMLs and related). A Web project provides miscellaneous wizards and editors, and makes it easy to build the Web archive (WAR).
- EJB Module project**
 Used for the creation of enterprise beans. An EJB project provides facilities for the creation of EJB components, editing of the deployment descriptors and easy building of the EJB archive (JAR).
- Enterprise Application project**
 Creates ready-to-run applications. It wraps the Web and EJB archives packed in the project into an Enterprise application archive (EAR), and provides editors for the application deployment descriptors. The SAP NetWeaver Developer Studio allows you to deploy the finished applications to the J2EE Engine directly from the IDE.
- Library project**
 Wraps deployable libraries for the J2EE Engine. You can also deploy them directly on the target J2EE Engine.

Saving Time

OK, that's it for the beauty of the environment. Now let's talk about how it really helps. The J2EE toolset provides a whole lot of wizards, editors and options that facilitate the J2EE development process.

The toolset provides wizards for generating base J2EE components: enterprise beans, JSPs, servlets, filters and listeners. These wizards generate the skeleton Java source of the J2EE component, thus saving your time. This includes the generation of necessary interfaces and their default implementations, default methods, import statements, and so on.

There is also an editor (and wizard page) for EJB business methods (see Figure 3). It automatically generates the method body (in **all** local and remote interfaces, unless you specifically remove it from some of them) according to the method name, arguments and return type that are supplied. With the EJB wizard you can create the business methods initially. If you decide to create or change them later at development time, you can use the EJB editor.

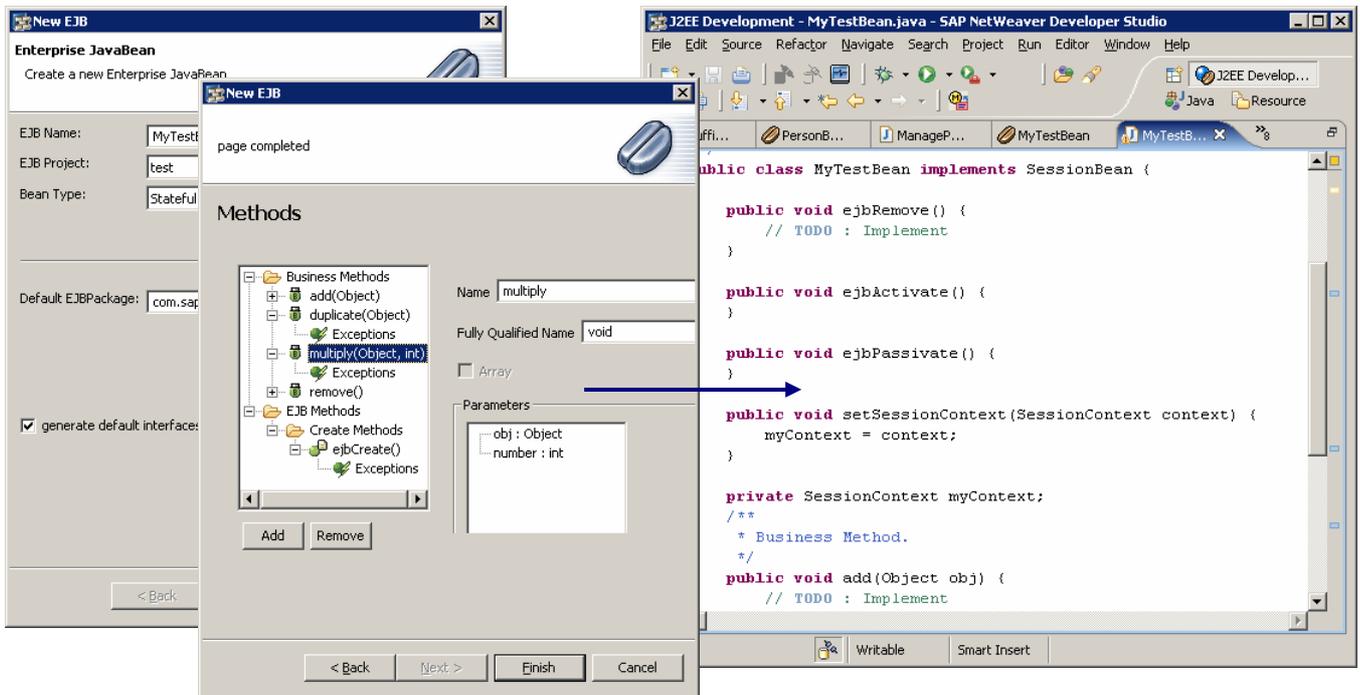


Figure 3: EJB Generation Wizard

For entity beans with container-managed persistence, the J2EE toolset is tightly integrated with the Java Dictionary . This includes:

- Easy Object/Relational mapping
- Validation of the column and table names to which the cmp- and cmr-fields are mapped

For both standard and SAP-specific deployment descriptors (*application.xml*, *application-j2ee-engine.xml*, *web.xml*, and so on), the J2EE toolset provides multipage editors for the specifics of each descriptor. You can use the multipage editors to edit the XML content of descriptors without working directly with XML tags. The multipage editors ensure the generated XML source is valid according to the DTD.



Figure 4: application.xml Multipage Editor

The J2EE perspective provides “smart” integration among different UI components. When you add/remove/change a component from one element, the changes are reflected in all editors and views concerned with the change. For example, when you add a new business method to an enterprise bean in the bean editor, the method is also added to the source code and deployment descriptor (*ejb-jar.xml*). Or, when you model a project visually via its UML diagram (in the *Diagram* view), all changes to the diagram are automatically propagated to the physical components of the project, and vice-versa. This feature is very useful, as it automates the development process to a maximum extent.

The SAP NetWeaver Developer Studio also provides other automation features, such as automatic generation of setter and getter methods for available fields, easy conversion of local variables to fields and vice versa, organizing imports, and so on.

Integration with the J2EE Engine

The J2EE toolset saves you from having to use external tools for deploying applications on the J2EE Engine.

From the *J2EE Engine* view, you can observe the status of the J2EE Engine cluster, such as available processes, hosts, ports, etc. You can connect to a J2EE Engine that is available on the local host or on a remote one.

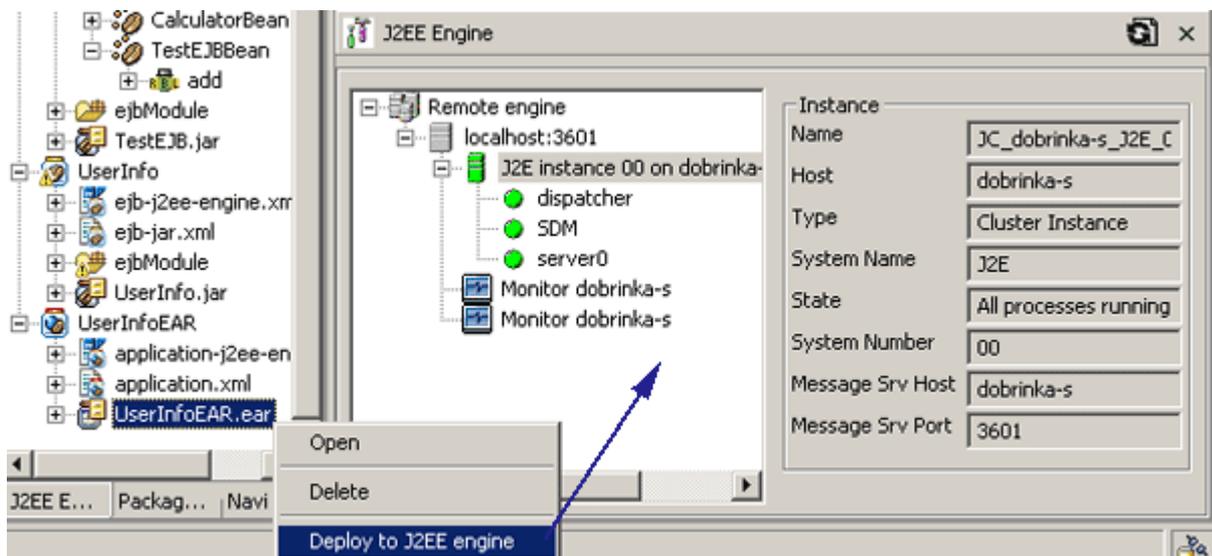


Figure 5: Deploying J2EE Applications to the J2EE Engine

You can also perform basic administration of server processes in the cluster, such as restart, enable/disable debugging, and so on. This feature is especially useful for remote debugging of J2EE applications. It allows you to restart a J2EE Engine process in debug mode, which is a prerequisite for remote debugging.

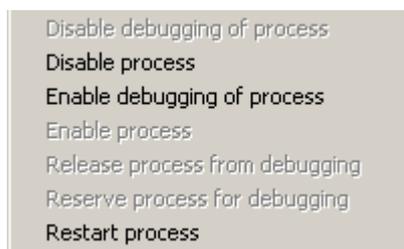


Figure 6: Commands over the Cluster Processes

Remote Debugging

The J2EE toolset allows you to debug applications on the fly, that is, while they are running on the target J2EE Engine. This includes the debugging of their servlet, JSP and EJB components. This feature enhances the default debugging available in the Eclipse IDE.

First, you need to add the desired breakpoints to the Java and JSP source code. Then you can build and deploy the application to the target J2EE Engine. Then you can start debugging from the Developer Studio. You can monitor the threads that are running, variable values, method execution, etc.

Easy Integration with Non-J2EE Components

J2EE applications can consist of various components, some of which are not part of the J2EE specification. These can be, for example:

- Web services
- Web Dynpro applications
- Portal components

You can easily import these components from the corresponding project type in the IDE, or from external JAR or PAR files.

Integration with the NetWeaver Development Infrastructure

When you work on large projects, you usually work in teams, being responsible for only specific parts of the development. In such cases, you need to be able to pass your newest work to your team mates and at the same time get their latest versions to compile and run the whole project. In other words, you need a version control system. The SAP NetWeaver Developer Studio works as a front end to such a system – the NetWeaver Development Infrastructure (NWDI).

The SAP NetWeaver Developer Studio provides a special perspective for team-shared versioned application development (the *J2EE DC Explorer*) and special project types for development components: *Web Module DC*, *EJB Module DC* and *Enterprise Application DC*. You can work with the Design Time Repository (DTR) directly from the Developer Studio without having to use separate tools for checking applications in and out.

Support for SAP-Specific Components and Deployment Properties

The SAP J2EE toolset is not only J2EE-oriented, it is also J2EE Engine oriented. It supports the development of all components that are supported by the J2EE Engine, including those beyond the scope of the J2EE specification. These are, for example:

- SAP-specific deployment properties (contained in *web-j2ee-engine.xml*, *ejb-j2ee-engine.xml* and *application-j2ee-engine.xml* files).
There are multipage editors that enable you to edit the contents of these descriptors without having to write XML code.
- J2EE Engine libraries
There is a special project dedicated for deployable libraries. You can deploy the libraries that are created directly to the J2EE Engine.

Summary

The SAP NetWeaver Developer Studio offers a convenient user interface and rich functionality for developing J2EE applications. It benefits from the wonderful features of the Eclipse platform, and adds specific features for the needs of J2EE application developers. It is worth trying out.

The NetWeaver Developer Studio is distributed as part of the SAP NetWeaver Application Server. You can download an evaluation version from

http://www.sdn.sap.com/sdn/downloaditem.sdn?res=/html/software_download.htm

(Accessible after you log on to the SAP Developer Network)

Disclaimer & 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.

Author Bio



Dobrinka Stefanova is an information developer at SAP. She is responsible for the user documentation of the J2EE Toolset, integrated in the SAP NetWeaver Developer Studio.