

# Introducing SAP NetWeaver Developer Studio, Designed to Handle Enterprise-Scale Java Development Projects

by Karl Kessler, SAP AG

To develop Java projects with enterprise-scale functionality, you don't just need powerful tools and services from your development environment. An integrated development environment (IDE) must also meet the challenges and tasks faced by large development teams working from the same code base — from writing code, to maintaining and updating programs, to synchronizing development activities and organizing central design and testing.

To meet these needs, SAP set out to provide development tools for Java that are not just J2EE-compliant, but that also offer the robust features on par with what ABAP developers have come to expect from SAP's tools in the ABAP Workbench. Now, as of SAP Web Application Server 6.30, Java developers have SAP NetWeaver Developer Studio, which offers large development teams full access to all components of a Java project, tools based on a common framework that makes them easy to use, and support for combining standard J2EE development tasks with SAP's visionary Web Dynpro UI technology.

## Filling the Gaps in Traditional J2EE Development Tools

SAP Web Application Server is the application platform for an integrated, highly scalable business solution

environment, and as of Release 6.30, it provides a development and runtime environment for ABAP applications and for Java applications based on the J2EE standard.

When it comes to designing applications and interfaces, the ABAP design-time environment is well established, having evolved over the past decade, and provides a repository-based approach that supports development by large teams. The ABAP Workbench is the central starting point for all ABAP development activities in a distributed development landscape.

In comparison, Java development is typically geared to the single developer working at a local frontend, and the J2EE standard simply leaves integration issues up to the individual tool vendor. The J2EE standard focuses primarily on the *runtime* behavior of enterprise-scale applications written in Java, and takes for granted the existence of J2EE-compliant development tools. So it's rare to find a Java IDE that offers solutions for the typical tasks involved in developing medium- and large-size projects:

✓ **Repository access:** Most Java design-time tools are file based. These tools typically lack features that developers of large projects with big development teams have become

accustomed to, such as version control and sophisticated impact analysis.

✓ **Support for central data types:** The need to define data types and database objects in a central location is rarely addressed in today's J2EE tools. Most tool vendors simply include a database and corresponding database tools. But, as a result, Java applications are most likely to be database dependent.

✓ **Advanced user interface features:** J2EE concepts for user interface abstraction (e.g., modeling and design of interfaces) such as JSP and servlets are browser dependent — unless you invest a serious amount of code discipline in your UI projects! In many cases, key functions such as input validation or value help are missing.

✓ **Distributed development:** Large projects must first be tested in a development landscape before they can be deployed to a productive environment. Pure J2EE deployment leaves open many issues, such as version control, dependency analysis, and component support. These issues require the help of a change management system for keeping track of all deployed development components.

To address these concerns, SAP has included a robust, integrated Java IDE in SAP Web Application Server

## What's Web Dynpro?

If you're currently using Java Server Pages (JSP) as the interface into backend systems, you've likely run into concerns with browser dependencies. SAP's alternative to browser-dependent UI programming techniques is Web Dynpro.

Web Dynpro is a model-driven approach to designing user interfaces that allows you to develop, view, and update your Web UI in a graphical way. The Web Dynpro approach structures an application into distinct layers that separate the interface from the backend business logic to form a highly reusable UI. In fact, the layers are so distinct that code-specific information is added in a very late phase of the development process. Web Dynpro also supplies much of the supporting code in the background automatically, so that you can ensure J2EE compliance and a complete, accurate UI.<sup>1</sup>

Web Dynpro for Java is included with Web AS 6.30 and integrated into the SAP NetWeaver Developer Studio, so you can create Web Dynpro Java UIs with this current release. For more information on Web Dynpro, see [www.sap.com/technology](http://www.sap.com/technology).

<sup>1</sup> See my article "Your 'Easy Way In' to Web Dynpro Development" in the April-June 2003 issue of *SAP Insider* and the article by Peter Tillert in this issue ([www.SAPinsider.com](http://www.SAPinsider.com)).

6.30. SAP NetWeaver Developer Studio becomes the main entry point for all development activities, offering developers a window onto a distributed design- and run-time landscape for J2EE-compliant applications, combining the flexibility of using the local file system with the reliability and scalability that a centralized infrastructure can bring.<sup>2</sup>

## Architecture of SAP NetWeaver Developer Studio

When SAP decided to deliver its own development environment, it wasn't long before the question came up of whether to embed the SAP tools in a pre-existing IDE. Most third-party IDEs in the market lacked the openness and extensibility that SAP sought, so SAP chose to build its tools on top of

<sup>2</sup> To support this distributed development environment, the SAP NetWeaver Developer Studio consists of client-side components, available in 6.30, and will also offer server-side components in its infrastructure in 6.40 — providing a "best-of-both" approach detailed in Wolf Hengevoss's article in this issue of *SAP Insider* ([www.SAPinsider.com](http://www.SAPinsider.com)).

Eclipse — an open-source, Java-based development platform. The Eclipse platform was designed specifically for development of custom IDEs, and comes with its own J2SE (Java 2 Platform, Standard Edition) environment for standalone Java applications.

In Eclipse, every extension is called a plug-in. Each plug-in adds functionality to a given tool at predefined extension points. What's more, a plug-in can include its own set of extension points, so you can even add plug-ins to a plug-in. All development tools in the SAP NetWeaver Developer Studio — including Web Dynpro, J2EE, Web services<sup>3</sup> (which are built on top of the J2EE perspective), and the Java Dictionary — are in fact Eclipse plug-ins.<sup>4</sup>

<sup>3</sup> See Martin Hugar's article on Web services in this issue of *SAP Insider*.

<sup>4</sup> For a good look at the SAP NetWeaver Developer Studio and the Eclipse platform, see Rudi Munz's article in this issue of *SAP Insider* ([www.SAPinsider.com](http://www.SAPinsider.com)).

If Eclipse has any limitations from SAP's perspective, they are in modeling the application, presenting logical hierarchies, or diagramming the application. (This is especially crucial when it comes to Web Dynpro-based design and technology, where so much supporting Java code is automatically generated — code that the typical Java developer probably doesn't want to see!)

So SAP and Borland jointly developed a modeling framework based on Eclipse. This framework abstracts from the physical source files to provide developers with visual models of a program's presentation logic. Both Web Dynpro modeling tools and graphical J2EE tools are based on the same common software layer.<sup>5</sup>

In fact, all SAP's Java development tools are built on top of this modeling framework, resulting in a highly sophisticated, consistent graphical framework for all kinds of development activities.

## Quick Tour of the Three Major Tools of SAP NetWeaver Developer Studio

For a first impression of how the SAP NetWeaver Developer Studio operates, here is a brief introduction to some of the tool sets and perspectives it offers.

### Java Dictionary

We'll start with the Java Dictionary. The Java Dictionary offers tools and wizards to define central data types (e.g., enumeration types for value help) for use in Java projects that will run on Web Dynpro. You can also create database objects, such as indexes and

<sup>5</sup> Borland also provides extensions to SAP NetWeaver Developer Studio for models based on the UML standard, the SAP NetWeaver Developer Studio Together Edition.

tables (see **Figure 1**), add them to the Java Dictionary, and later deploy them to your database. The dictionary definitions are stored in a database-independent fashion, so you can deploy them to all database platforms officially supported by SAP.<sup>6</sup>

### J2EE Tool Set

SAP NetWeaver Developer Studio offers editors and wizards for creating all the typical components (e.g., JSP pages and servlets, session beans, entity beans) of a J2EE-compliant application.<sup>7</sup> The J2EE tool set lets you easily create the Web components of your J2EE applications by means of JSP and servlet editors, which even offer such fancy features as syntax-based coloring of code and intellisense (which automatically completes method names, etc.). The J2EE tools can validate JSP syntax against the corresponding DTD available over the Internet, and business logic can be implemented through wizards for Enterprise JavaBeans. J2EE 1.3-compliant session beans, entity beans, and message-driven beans are all supported.

### Web Dynpro Tools

The heart of the Web Dynpro approach is the Application Modeler, which depicts the various components of your Web UI and their relationships. Navigational controls are shown as lines that connect Web Dynpro views. These views in turn contain Web Dynpro controls (input fields, push buttons, tables, etc.). Web Dynpro technology allows you to quickly assemble views, even as layouts or data sources change. By relying on a compo-

nent-based approach, you will find a high degree of reuse in your Web Dynpro applications.

Additional features of the SAP NetWeaver Developer Studio include:

#### ✔ **SAP Enterprise Connector: The BAPI Browser and Proxy Generator**

When you develop your first Java application, you will most likely wish to connect to an existing SAP database. To call BAPIs from a Java application, SAP has delivered the Java Connector (JCo). However, JCo is not strongly typed; to avoid runtime errors, you must check that all data type information is valid in the parameters of any JCo call.

To overcome this limitation, BAPI Browser and Proxy Generator let you browse through BAPI lists that include data type information to help you generate a “type-safe” Java proxy for

use in your Java projects (see **Figure 2** on page 4). The BAPI Browser uses your SAP Logon files to connect right to existing SAP systems.

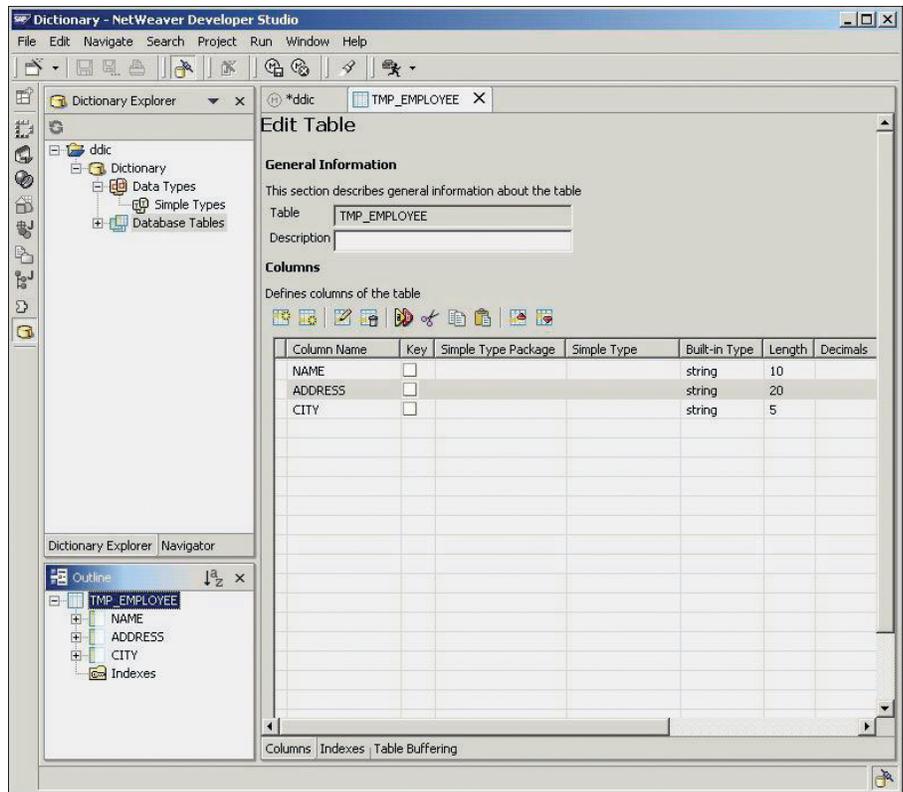
#### ✔ **Support for Open SQL**

SAP offers tool support for database-independent access to tables. This is done by means of an SQLJ precompiler — a precompiler that translates embedded *SQL* commands into plain *Java* code. Later, during debugging, the original *SQL* code is presented directly to the developer so that you can debug your application rather than the generated code.

#### ✔ **Resources for Team Development**

SAP NetWeaver Developer Studio, with local access via Eclipse, is also tightly integrated in SAP’s development infrastructure, which consists of:

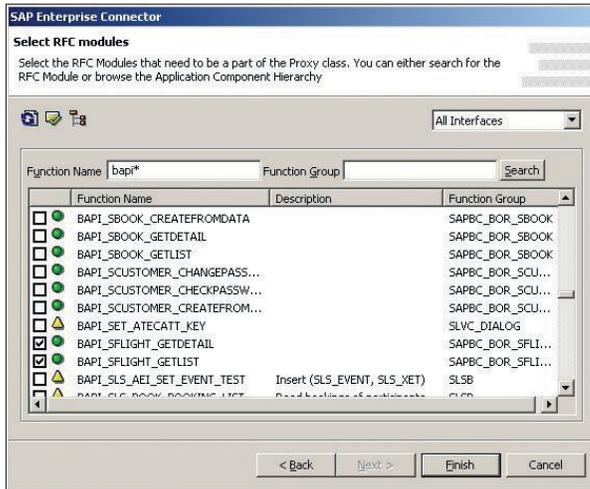
- A design-time repository for versioning Java source files



**Figure 1** Java Dictionary

<sup>6</sup> In this release, SAP’s Software Deployment Manager (SDM) handles deployment. In future releases, it will be carried out automatically by the Change Management Service (CMS). For more on these components, see Wolf Hengevoss’s article in this issue of *SAP Insider* ([www.SAPinsider.com](http://www.SAPinsider.com)).

<sup>7</sup> For details on the structure of J2EE-compliant Java projects, see my article “Faster and Easier J2EE-Compliant Design with New Tools from SAP NetWeaver Developer Studio” in the July 2003 *SAP Insider* ([www.SAPinsider.com](http://www.SAPinsider.com)).



**Figure 2** BAPI Browser

perspective. From the Development Configurations screen, shown in **Figure 3**, you can browse configurations consisting of software components that, in turn, are made of development components. Development components are stored centrally in DTR to ensure consistency and reusability, but they can also be viewed by a developer locally via Eclipse.

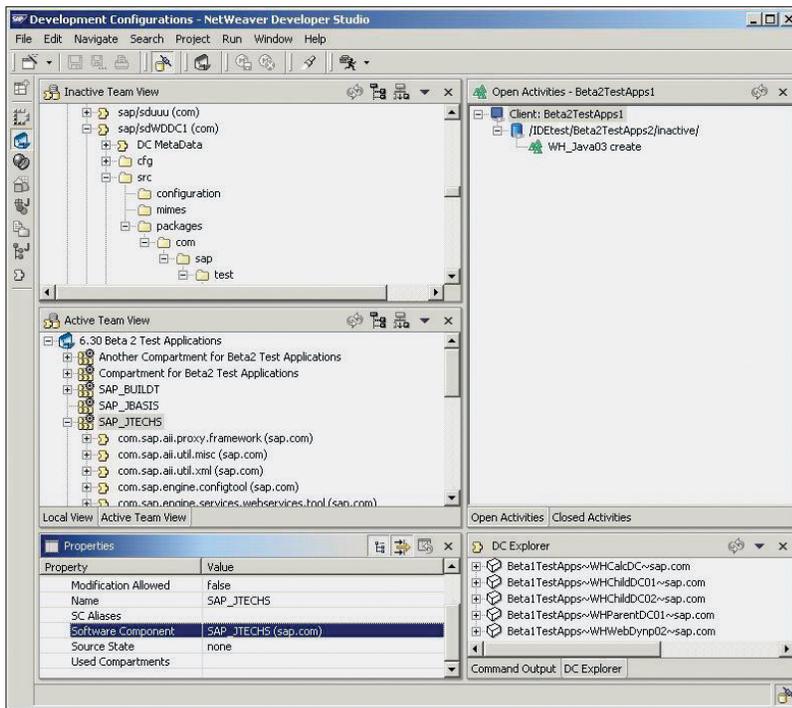
For more on the components that support the Developer Studio, see Wolf Hengevoss's article in this issue of *SAP Insider*.

### Summary

SAP's approach toward an integrated development environment offers Java developers an environment comparable to that of their ABAP colleagues. It features a central repository for development objects, and services that support large teams of developers and enterprise-scale Java projects. At the heart of SAP's innovations in this area is the SAP NetWeaver Developer Studio, which offers full developer access to the project, just as the Workbench does for the ABAP environment.

What's more, all the Developer Studio tools are based on a common framework, making it easy to become familiar with their features and to combine SAP's Web Dynpro technology with standard J2EE development processes.

For more on SAP NetWeaver Developer Studio, visit [www.sap.com/netweaver](http://www.sap.com/netweaver). ■



**Figure 3** Development Configurations Screen

- A component build service that produces runtime objects such as Java archives
- A change management service to distribute and deploy the Java archives in a distributed system landscape

In addition, the Developer Studio allows developers to easily browse

through designtime and software components to view the up-to-date status of development efforts. First, from the NetWeaver Developer Studio, the development team can view the contents of the Design Time Repository (DTR) — the repository for development objects (Java classes, XML sources, etc.) — through a DTR

Karl Kessler joined SAP in 1992. He is the Product Manager of the ABAP Workbench and the SAP NetWeaver Developer Studio and is responsible for all rollout activities. You can reach him via email at [karl.kessler@sap.com](mailto:karl.kessler@sap.com).

