

# Is Your J2EE-Compliant Development Platform Really Up to Code?

Q&A with Rudi Munz, Senior Vice President of Development Platforms, SAP AG



*Rudi Munz,  
Senior Vice President of  
Development Platforms,  
SAP AG*

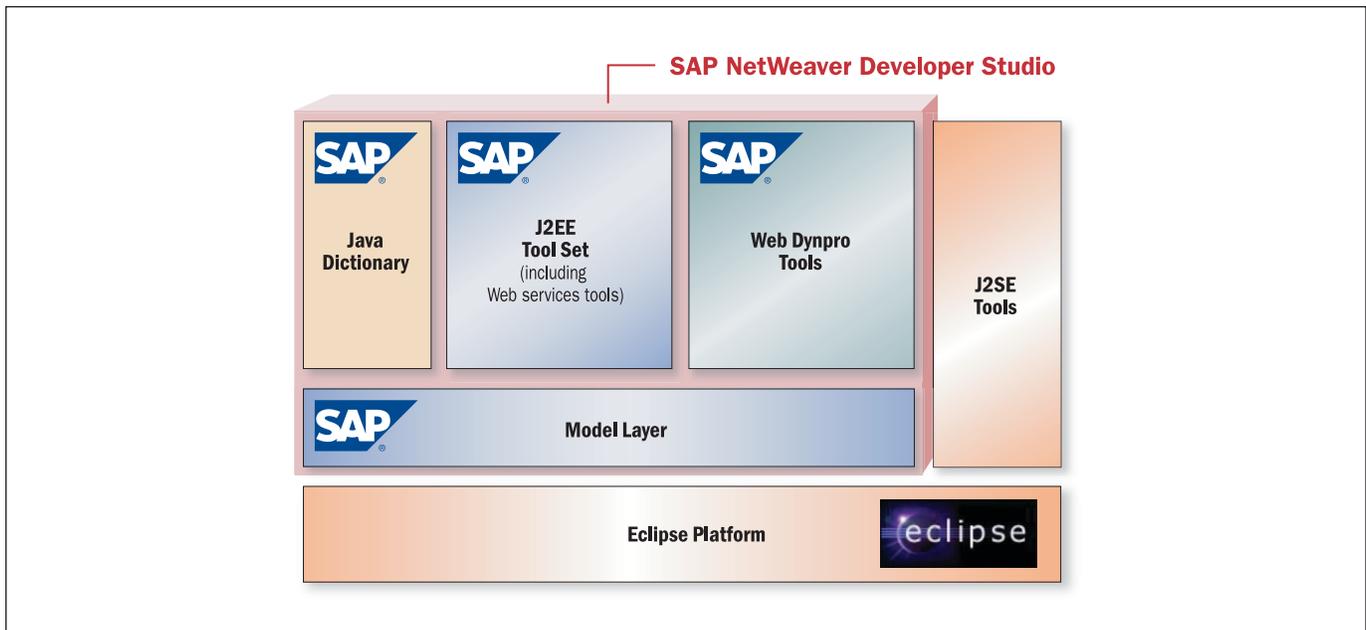
**Q. SAP only recently entered the Java development platform fray with the J2EE-based SAP Web Application Server. Presumably, one J2EE-compliant API looks just like another, so why should SAP Insider readers take notice — particularly if they are already immersed in Java development projects on other platforms?**

One would certainly hope that Java platforms purported to be J2EE compliant look like one another at the API level! The answer to your question lies first in the quality of the implementation and, second, in all that is *not* supported in the J2EE standard — namely the mechanisms needed to support an application product through its fundamental lifecycle paces. How, for example, do you deal with issues of enhancements, modifications, testing, deployment, and functional release upgrades in a comprehensive, consistent, and cost-effective manner? How do you scale your development environment such that it can support a large population of developers working on the same code base, and yet retain control over your development processes? How do you tackle the integration of the individual development components and their dependencies?

You don't find these things — things that ABAP developers take for granted — accounted for in the J2EE standard. The problem confounding Java developers is that no development platform had accounted for them either! This is why you generally see most Java application servers being used as glorified Web servers, encasing existing applications in a new UI. The robustness needed on the platform level to place a complex transactional workload on Java application servers is just not there. This presents SAP with the opportunity to parlay our expertise with the ABAP application server into the creation of the first enterprise-ready Java platform. The result is our Web Application Server, which is part of the SAP NetWeaver technology stack. Java developers gain access to Web Application Server concepts and services from the SAP Java IDE, which we call *SAP NetWeaver Developer Studio*.

Now you can create both frontend and backend applications with a common Java development platform (see **Figure 1** on page 2). SAP NetWeaver Developer Studio includes a component model for Java development, complete with a component hierarchy and management of component dependencies. It includes a design-time repository, automatic builds of changed and dependent components, and automated deployment to the local or a central J2EE server. To increase your programming productivity, we offer a declarative UI programming model (Web Dynpro) as well as a persistency layer, which abstracts from the database engine and provides complete DBMS portability. This is complemented by support for Web service creation and integration, support for automated testing, monitoring capabilities, and a whole lot more.

Frankly, achieving J2EE compliance is relatively easy — that's why there's no shortage of J2EE platforms out there. Delivering a J2EE platform that can reliably support the



**Figure 1** Key Components of the SAP NetWeaver Developer Studio and the Eclipse Platform

full depth and breadth of an application’s life cycle at the enterprise level is not. With all the experience inside SAP, we are years ahead of the others in this regard.

So I’m going to answer your question with a question: Given a level playing field at the API level, why *wouldn’t* you opt for the platform that is actually used by SAP application development and can bear the most demanding SAP applications?

**Q. The norm, for now, is to use Java applications as a facade for background applications that do the transactional work. What’s wrong with maintaining the status quo?**

At heart, I think most of us are staunch supporters of the if-it’s-not-broken-don’t-fix-it ideology. If using Java as a facade serves your needs, I’m not going to suggest you suddenly shift gears. I am suggesting, however, that you make your next Java project an SAP Web Application Server-based endeavor. You will realize immediate productivity gains. You will ensure that the quality and reliability of your application is “up to code” for your production environment. You will accelerate the speed with which you can develop applications. You’ll have the entire life cycle of that Java project covered. And on all counts, you can expect significantly better TCO. Over time, as your applications mature, the TCO factor will become more and more pronounced.

Upon closer inspection, I think most will see a development landscape that is in need of repair. Today’s Java

landscapes are typically riddled with productivity and TCO inefficiencies.

The long-term prognosis for Java platforms that cannot support the full range of application development life cycle issues isn’t good. IT organizations need to accelerate development times, increase their output, and maintain high quality standards. You can’t achieve these goals and contend with varying application server types and administration tools.

I know this from firsthand association with a very large development organization — at SAP we have upwards of 6,000 application developers. A big part of our annual development budget is devoted to these individuals, and it is imperative that high volumes of high-quality code be produced and be serviceable for years to come in production environments. What are we doing to secure our competitive stance? We’re endorsing Java as a development platform in addition to our proven ABAP platform.

Our newer solutions — solutions such as SAP Enterprise Portal and Internet Sales, which are built to service really large end-user communities — are based on the SAP Web Application Server and built with the SAP NetWeaver Developer Studio.

So we’re not just a provider of the Web Application Server and its Java IDE. We are also a big, demanding consumer of this technology. We are, perhaps, the most grueling reference account for it! As the largest business application software firm in the world, we wouldn’t take a

gamble on our Java platform and its development tools. That's why we use the SAP Web Application Server and its Java IDE.

**Q. Why is SAP embracing Java in this way? And how would you position Java relative to ABAP?**

We want to offer our customers more than an ABAP monoculture. The time is now right to open up our ecosystem to the large Java community and benefit from all the Java momentum that's out there. Conversely, the time is right for the Java community and technology stack to benefit from what we have to offer.

Both stacks are here to stay and will nicely integrate and interoperate with each other. There are no plans to abandon ABAP or migrate existing ABAP applications to Java.

Still, there are some signals ABAP developers need to pick up on. First, it's time to acquaint yourself with Java. Start to acquire some skills in this area. Don't get wedged in an ABAP silo. You don't want to be tending exclusively to SAP applications while the young university graduates do the high-profile, cutting-edge, frontend work. There is no reason to have the Web site folks and SAP application folks in separate camps. SAP Web Application Server offers you a common, flexible platform for both. You certainly don't want to run the risk of becoming tomorrow's equivalent of a "mainframer." Only limited future career opportunities go along with that moniker. So strike a balance. A balance is good for you personally, and good for your organization. There is a lot to be gained by knowledge transfer and collaboration between both software stacks.

In addition to that, Web services and consequently our Enterprise Service Architecture abstract from whatever software stack is being used. So the times are gone when integration of applications was much easier if they were written within the same software stack. In fact, this enables the coexistence of software stacks and of software components written in different languages.

**Q. Does it go without saying that the Java development environment of the SAP Web Application Server offers the best way to integrate new Java-based applications with existing SAP solutions?**

SAP NetWeaver basically is an integration platform, but it's not just for integration with your SAP landscape — it's also designed for all your other integration needs. So with the next integration project inside your company, give SAP NetWeaver a try. ■