



This article appeared in the Oct • Nov • Dec 2009 issue of *SAP Insider* and appears here with permission from the publisher, Wellesley Information Services (WIS), www.WISpubs.com.



New Tools and Time Savers for Developers

The Latest from SAP NetWeaver Enhancement Packages



Karl Kessler (karl.kessler@sap.com) joined SAP in 1992. He is the Product Manager of the SAP NetWeaver foundation – which includes SAP NetWeaver Application Server, Web Dynpro, ABAP Workbench, and SAP NetWeaver Developer Studio – and is responsible for all rollout activities. Currently, Karl's focus is on how SAP NetWeaver powers SAP ERP and SAP Business Suite software.

With the introduction of enhancement packages, SAP has clearly laid out a strategy that allows customers to better manage innovation in their solution landscapes. That strategy also now has implications for the technology platform with direct benefits for developers, including new productivity tools and useful features that are ready for you to explore.

Enhancement Packages – Now for SAP NetWeaver, Too

By installing enhancement packages, companies can select the features they want to implement to extend their SAP solutions, but remain on a stable core platform running SAP NetWeaver 7.0. The latest enhancement package for SAP ERP 6.0 – enhancement package 4 – includes innovations in solution and industry areas.

But as administrators know, and developers should be aware, SAP NetWeaver also now has its own set of enhancement packages (see sidebar).

IT teams should also note that strict rules apply for enhancement packages in terms of compatibility and aligning SAP ERP 6.0 enhancement packages with those for SAP NetWeaver:

- SAP ERP 6.0 enhancement packages 1, 2, and 3 ran on SAP NetWeaver 7.0, and each required its particular support package level from SAP NetWeaver 7.0.
- Enhancement package 4 for SAP ERP 6.0 requires SAP NetWeaver 7.0 enhancement package 1.
- The next SAP ERP 6.0 enhancement package, enhancement package 5, will require enhancement package 2 of SAP NetWeaver 7.0.

Why Enhancement Packages for SAP NetWeaver?

Many of the most innovative SAP NetWeaver features – including features for developers – are only found in SAP NetWeaver technology “hubs” such as SAP NetWeaver Composition Environment (SAP NetWeaver CE) and SAP NetWeaver Process Integration (SAP NetWeaver PI), which are both based on the SAP NetWeaver 7.1 codeline.

While SAP NetWeaver CE is completely based on Java, SAP NetWeaver PI is based on the double stack (both ABAP and Java). This means that SAP NetWeaver PI introduces many new ABAP productivity features that were intended for all solutions on top of SAP NetWeaver 7.1x and subsequent codelines.

If you're like most of the ABAP community at this time, you are working with SAP ERP 6.0 and thus the SAP NetWeaver 7.0 codelines – an important part of the SAP strategy to ensure customers can remain on a stable core platform.

To address this gap for developers, though, SAP is introducing enhancement package 2 for SAP NetWeaver 7.0, which includes the ABAP and Workbench innovations found in SAP NetWeaver 7.1x that do not violate the strict compatibility rules for enhancement packages, giving ABAP developers in the installed customer base the ability to significantly increase their productivity.

While these details might seem to apply only to administrators, developers should also know that enhancement package 2 for SAP NetWeaver 7.0 will introduce some tool improvements in the ABAP Workbench – improvements that would otherwise only be available with SAP NetWeaver 7.1x. We will examine the most prominent of these advances for developers in this article.

For a preview, you can explore these features now if you have access to the ABAP tools in SAP NetWeaver Process Integration (SAP NetWeaver PI), which runs on SAP NetWeaver 7.1x.

New Code Completion

Code completion is one of those popular features that you've seen if you are familiar with Eclipse (in the Java development environment) or Visual Studio (for Visual Basic or C#). Now code completion in the ABAP Workbench is also available, with SAP NetWeaver 7.0 enhancement package 2.

By simply starting to enter code, you get immediate code help with drop-down menus that supply code suggestions. Not only does it supply code, the tool actually looks into the repository, displays suggested code in a quick window session, and allows you to select the code – all with speed that won't hinder your development efforts (see sidebar).

The ABAP editor in these environments incrementally parses the source text and suggests the completion of the code based on syntactical conventions and the environment that surrounds the program. For example, the editor can suggest proper keywords or the names of the underlying development repository.

Figure 1 shows how the list of available fields pops up if you specify a table or structure. Simply pick the field and the code will be automatically inserted. And with new inspection capabilities, you can look up the methods of a class or the parameters of a method invocation and, rather than navigating to a separate screen, you can immediately insert the complete method call.

With code completion, you can avoid a lot of typical ABAP Workbench navigation commands, and you also skip the extra step of having to copy and paste code on your own – a big time saver.

A New Two-Way Class Editor

Every ABAP developer has used the Class Builder to edit global classes. Like its cousin the function builder, the traditional Class Builder relied on a form-based

Code Completion: Overcoming the Speed Barrier

Of course, it is crucial that all the code completion steps occur in real time – anything else disrupts developer productivity. For ABAP development, speed has been a long-standing challenge in developing the code completion feature.

The problem has been that – unlike a Java compiler, which can quickly provide code completion since it resides on the developer's machine and can access its required archives from a local folder – the ABAP compiler and class libraries are on the server side. In addition, ABAP programs typically depend on data and type declarations in the ABAP dictionary, which also resides on the server.

To support fast code completion in ABAP, the server-side environment is now partly being transferred to the developer's front-end systems.

transaction with multiple steps: First, you specify all the ingredients of a class (constants, attributes, methods, and events) in tabular form with drill-down dialogs for all subordinate items (such as the parameter of a method).

Then, the Class Builder would typically automatically generate the source containing all the attributes and signature information, just as you would manually type them in as a local class in your ABAP program or report. Finally, from these tables you switch to the ABAP editor to type in the code.

Now, developers can avoid this multi-step process. With the new source code-based Class Builder (see **Figure 2** on the next page), you have a true two-way editing tool that keeps track of all changes and applies them implicitly to your code, so you can

✓ NOTE!

If you're like many organizations, you are running SAP NetWeaver 7.0 for your SAP ERP 6.0 solution while at the same time running SAP NetWeaver 7.1x, the basis for SAP NetWeaver Process Integration (SAP NetWeaver PI). If your organization uses SAP NetWeaver PI, you already have SAP NetWeaver 7.1.

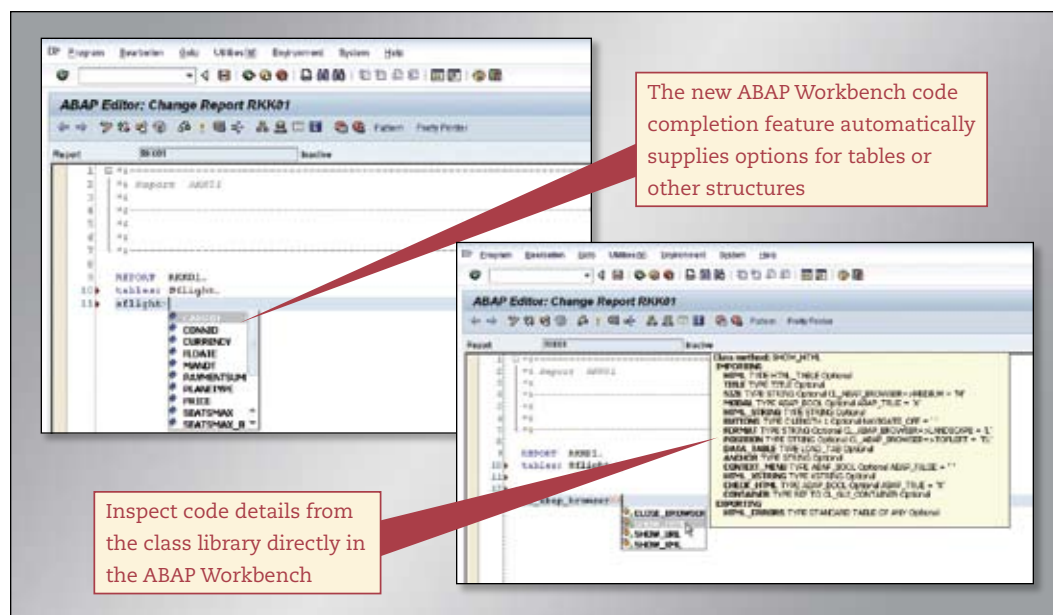


FIGURE 1 ◀ The code completion feature in the ABAP Workbench, which includes the ability to immediately view code details, is available with SAP NetWeaver 7.0 enhancement package 2 or SAP NetWeaver 7.1x

NOTE!

If you are working with SAP NetWeaver 7.0 enhancement package 1, you'll see something akin to code completion in the ABAP editor. The compiler suggests keywords and supports syntax color-coding based on ABAP standards, but it is not capable of presenting the technical names directly from the repository.

seamlessly move from the tables and back to the ABAP editor. (If you prefer a more form-based environment, the new Class Builder can still support that as well.)

A New, Integrated UML Viewer 7.6 7.2)

UML (Unified Modeling Language) is an accepted standard for modeling object-oriented systems in a graphical way – everything from use case diagrams to state transition diagrams and much more. While earlier ABAP Workbench tools did support interfaces to other tools using UML conventions, this is now built right into the Workbench; there's no need to initiate program-to-program communication.

The integrated UML tool is based on JNet, a graphics package used in other areas of SAP software, such as the Workflow Editor. (JNet is a Java application that is loaded into SAP GUI.)

Figure 3 shows the UML viewer inside SAP GUI's main window. A complete ABAP package is displayed with all its classes, their ingredients, and their relationships:

- Classes are depicted as rectangles.
- All attributes, methods, and events are displayed within the class. The visibility of an attribute or method (based on whether it is public, protected,

or private) is shown as a prefix (+, -, #) which provides insight into the class's design.

- If a class has an association to another class, this is depicted with a light line and arrow.
- A dashed arrow indicates a relationship based on an interface implementation, while a solid arrow indicates an inheritance relationship.

The class CL_GUI_CONTROL, which is a generic class for GUI controls (think of push buttons, radio buttons or input fields), is shown at the bottom of Figure 3. This class has an association to a GUI container whose name is "PARENT." CL_GUI_CONTROL inherits from CL_GUI_OBJECT, indicated by the solid arrow from the descendant class to the ancestor class.

Like other graphical tools, the UML viewer allows you to zoom in and out of the presentation view, auto-arrange the items, and use a navigational window pane to quickly traverse large graphs. You can analyze a single program to check inner dependencies, or you can look at a complete ABAP package.

For those projects with complex class interactions and behavior, the UML viewer is a powerful tool for inspecting, analyzing, and troubleshooting the architecture of your systems, and for understanding the structure of your networks of classes, successors, ancestors, descendants, and interface implementations.

New Debugging Capabilities

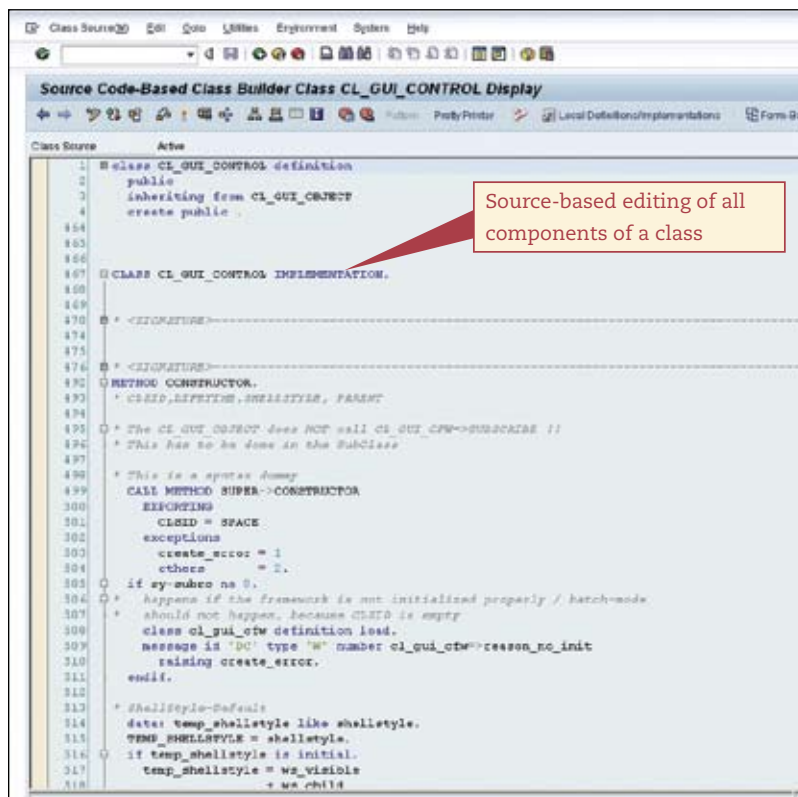
With SAP NetWeaver 7.0 enhancement package 2 (and SAP NetWeaver 7.1x) also comes enhancements to the ABAP Debugger. A two-task approach – separating the debugger screen from the application that you're debugging (the "debuggee") – has been state of the art for the debugging tool for some time, for faster speed and high availability. The two-task debugger is based on a debugger API that allows you to control the state of the debugger.

This API is now available for programmatic access inside the debugger itself (see Figure 4). This new approach to debugger scripting allows you to automate certain tasks in a debugger session.

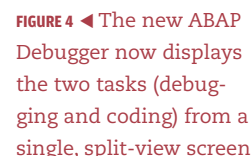
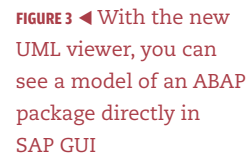
Debugger scripting allows you to call small ABAP programs that do the analysis and then control the debugger. Output such as the sequence of executed statements can be written to a trace file and examined later.

Scripting is basically as simple as writing ABAP code. A script wizard, available directly from the

FIGURE 2 ▼ The new Class Builder offers a built-in editor to change all components of a class in a source-based fashion



Layer-aware debugging helps you skip software layers (such as Web Dynpro) that you don't want to see when stepping through your code. You can also now specify the visibility of layers by using debugger



SAP NetWeaver 7.0 continues to provide a stable core for the enhancement packages you choose for business solutions.

profiles. (Think of these profiles as combinations of object sets that define the development objects that you are interested in debugging.) By setting a debugger profile, you can set the “debug scope,” so to speak. By applying a debugger profile, you can control the extent of your debugging efforts (including commands such as STEP INTO) based on the current profile.

New Language Features

Enhancement package 2 for SAP NetWeaver 7.0 also delivers new language features: one coming directly from a need in the banking and insurance industries, and another supporting faster access to table data.

“Decfloat” Data Type for More Precise Decimals

A new data type, decimal float (or decfloat), has been introduced to mediate between the restricted range of traditional packed decimals and the inaccuracy of floating point for commercial calculations.

Decimal float, as its name suggests, offers a large range and gives you the greater flexibility and precision for working with decimals that is required by business applications. With enhancement package 2, this new data type is now consistently integrated into the ABAP language and properly reflected in all important concepts such as internal tables, SQL access, and data conversion rules.

Secondary Indices to Improve Performance

Every ABAP programmer is familiar with the concept of internal tables. You might choose to locate data in large internal tables by means of sorted or hashed primary keys. However, in many situations, users need to access the data fast – and from different angles. If you’re working with large data volumes, table performance can be a challenge.

Consider, for example, the SAP demo flight database, which uses flight identification numbers as a primary key, allowing a user to search for flights using departure/arrival information. If the flights are stored in an internal table, the application must scan the full internal table for the result.

Now, new internal tables support secondary indices that search the secondary index based on criteria given by the user to provide much faster user access to the data. This would allow you to include a secondary index in your table based on departure/arrival times.

As on the database side, there is a certain trade-off for the developer, since the management of

secondary indices require additional memory. This can mean that the process of inserting a record into the internal table becomes slower because these secondary indices also need to be maintained as part of this process. But if you manage large data sets and you do a lot of loop processing, your internal table operations can be accelerated significantly.

Summary

This overview is by no means complete. Enhancement package 2 for SAP NetWeaver 7.0 offers many new capabilities, including new features in shared objects handling and resumable exceptions, and improvements in many ABAP trace and test transactions. For more details, see the ABAP pages on the SAP Developer Network (www.sdn.sap.com → *SAP NetWeaver Capabilities* → *Development and Composition* → *ABAP*) as well as sessions from SAP TechEd 2009 (www.sdn.sap.com/irj/scn/sapteched).

Most importantly for IT teams generally, it’s critical to stress that SAP NetWeaver 7.0 continues to provide a stable core for the enhancement packages you choose for business solutions. And with enhancement packages for SAP NetWeaver, you can also look for innovation for developers, enabling increased developer productivity from both the language and tools. ■

AdditionalResources...

...from **SAP** insider



- + “Get Ready to Implement SAP Enhancement Packages for SAP ERP 6.0” by Christian Oehler and Thomas Weiss (*SAP Professional Journal*, Volume 11, Update 1, www.SAPpro.com)
- + The **Administration and Infrastructure 2010** conference in Orlando, Florida, March 23-26, 2010 (www.sapinsideradmin2010.com)
- + “Avoiding Pitfalls in a Rapid SAP ERP 6.0 Upgrade” by Ranjan Baghel (*SAP NetWeaver Magazine*, Summer 2009, www.NetWeaverMagazine.com)
- + “New ABAP Editor and Debugger Boost Developers’ Productivity,” an Under Development column by Karl Kessler (*SAP Insider*, April-June 2006, sapinsider.wispubs.com)