

Hello,

Thanks for joining this podcast on A quick tour of productivity-enhancing capabilities in Rational Application Developer for WebSphere Software.

My name is Adeel Omer and I'm the Solutions Marketing Manager for Rational Architecture, Design and Construction.

This podcast is meant to serve as a companion piece to a more in-depth examination of the topic of productivity-enhancing features available to users of Rational Application Developer for WebSphere Software. As you all might be aware, in the 2008 Evans Data Corp. IDE Users' Choice survey, software developers ranked IBM's Rational Application Developer higher than Integrated Development Environments (IDEs) offered by many of its competitors. To view an archive of this webcast, please visit the Global Rational Users Group web site at rational-ug.org and look under "Events" --> "Webcasts" and then view the Community Webcast Archives on the left navigation bar.

So with that said, let's take a quick trip around some of the cool features that can help you really cut down on the amount of time it takes to write, refactor, improve, debug, test, and deploy your code with Rational Application Developer.

The first category of capabilities I'd like to highlight are some of the changes around the Rational Application Developer infrastructure. A lot of these were the result of direct feedback from users such as yourself and help to improve install times and footprints, performance, etc. Rational Application Developer is installed via the IBM Installation Manager, which simplifies, and optimizes your install to minimize your install footprint by only installing only the options you need and also shell share with other Rational products to help developers more easily manage their desktop

For enterprise scenarios, your custom install image and configurations can be provisioned through response files. Furthermore, help documentation, tutorials, samples, etc. can be stored in a variety of places: you can choose to access them from the web/internet every time, you can choose to have everything installed on your local system, or, you can actually have these hosted on a shared server within your organization or company, so you don't get the footprint on your personal machine, but you can still access this info without having to go to the world wide web.

As far as some tips around performance are concerned, make sure you close unused/unneeded projects on your workspace. You can simply right-click on a project on your workspace and choose to close it if it's not being used. Create multiple workspaces - i.e. don't have just a single, big workspace with everything. Not only is it easier to organize, but it provides better performance. And finally, make sure you close unnecessary editors on your workbench that aren't being used. There are more tips around project migration on the webcast mentioned earlier in this podcast.

Let's move over to Java Enterprise Edition, or JEE Application delivery. Rational Application Developer has been the leader in providing enhanced features for automated development of J2EE and now JEE applications using the latest standards. For JEE application development, Rational Application Developer provides Content assist and as you type validation which allows you to not have to memorize the syntax, or if you do get something wrong, immediately find out about it with instant validation. The validation can be tuned to check for a customizable set of criteria, all of which can be controlled by the user. Couple that with Quickfixes for code and project configurations and you're much better off with fixing your bugs faster than if you were to look for the information manually on how to fix the issue. When changes need to be made to the project, as is always the case, Advanced refactoring options to allow you to modify and maintain code in an iterative manner as these changes and references get propagated throughout your workspace, with less human interaction and less human error. Another neat feature is Annotations support; with Java providing this capability, RAD provides an annotations view that

allows you to quickly search for, add, and modify annotation properties. This again, saves a lot of time as you don't need to create specialty files such as deployment descriptors, and you still get the content assist and error-checks on Annotations just like you do with the rest of your code.

JPA or Java Persistence API is a fantastic new technology allowing you to work with data driven applications. The JPA wizards in RAD can help create entity classes from existing databases, create new tables from existing entity classes or create mappings between existing entity beans and database tables. You can go from nothing but a database, to a complete data-driven application that allows you to access and update the data using a simple web interface without manually having to write a single line of code! The visual construction capabilities using palettes do all the coding on the UI and generate all the necessary code as you simply point to the DB and create your queries and define relationships between the tables.

Now, as your application grows, it may not be practical nor desirable to keep your entire application in source form in your workspace. Large workspaces can include more than a hundred projects. It is very unlikely that every developer will need to change every one of those projects every day. A better approach is to keep only the projects or modules that you intend to changing in source form and keep the other projects or modules that you are using in binary form. Doing this will greatly reduce the memory footprint of IBM® Rational® Application Developer and greatly speed up many common development tasks. You will still be able to view the source code in the binary projects and do most of the productivity operations, such as Open Declaration (F3), that many of you have come to depend on often. You can learn about how to do this from an excellent article linked from the webcast.

Other neat features on the webcast go into more detail around JEE including EJB 3.0 support with class diagram visualization, the embedded WebSphere Application Server test servers and how these can be tuned to your liking in terms of whether you'd like the server behavior optimized for testing and development, applications deployed automatically, and if the server should remain running after you shut down the workbench.

Moving on to enhanced productivity for Rich Internet Application and Web 2.0 development. RAD provides support for the Dojo AJAX toolkit and you can build pages visually using the Dojo Widgets from a palette for drag-and-drop development and access to services. When it comes to developing JavaScript, it's really great to see the same level of developer support in terms of content assist, validation and refactoring. Normally IDEs tend to provide much more support for Java code versus JavaScript, but you really get the same amount of help from the IDE in the case of Rational Application Developer; there's even a JavaScript debugger, in this case FireBug is integrated with RAD and it allows you to step through your JavaScript code in RAD as you step through your web page.

You can find out more tips on features such as JSF 3rd party component library support and building your own custom components for JSF, on the webcast.

Now a quick note on how RAD helps you ensure quality during the development process. You get a number of different features. One of these is Line Level Code Coverage. You can create eclipse-based or HTML BIRT reports on coverage of code within your app at a configurable level, going all the way to every line of code. You can apply filtering to include/exclude packages, classes, methods as you wish and invoke code coverage via Ant scripts to make this a separately running process with your other regular activities around build, etc.

RAD also provides monitoring tools that help you to identify performance and memory usage problems in your Java and Web applications, regardless of the application server on which you are running.

The profiling tool enables you to collect and display data related to the runtime behavior of your Java program. It can help you identify problems related to application performance by showing which operations take the most time, are repeated most often, or result in memory leaks.

All this reporting really helps you find out how your application is running and where there might

be bottlenecks for performance, etc. so you can refactor the code accordingly or take appropriate actions.

The webcast talks about other capabilities around quality, such as probes, which are pieces of Java code you can inject into your code and invoke at certain times to gather application information.

And finally on today's podcast, I'd really like to talk about a feature that's received a great welcome from the community, and that feature is something we call "Collaborative Debug". Using Rational Team Concert integration with Rational Application Developer, you can not only manage your work items from your IDE, but also kick off debug sessions on a server, instead of your local workstation. The advantage of this approach is that when you get stuck at a certain point and need the help of a subject matter expert, or the code owner, you can "park" the debug session on the server where it stays frozen in that state. You can then transfer the session to another team member, or just let them know via email or any other method that you've left a debug session parked on the server with a particular name. This other person, can then simply look at the team server and instantly start from the point where you left off. Breakpoints can automatically be imported and this way, we just got someone to help us on their free time without getting them to spend any effort on reproducing the problem. The debug session can be parked and handed off as many times as you want until the problem is resolved. With globally distributed development teams, this feature has received a great welcome from the community. If you want to see it in action, please check out the webcast.

Again, to get to the webcast, please visit the Global Rational Users Group web site at rational-ug.org and look under "Events" --> "Webcasts" and then view the Community Webcast Archives on the left navigation bar. I hope you picked up a few pointers on how to use Rational Application Developer's capabilities to your benefit and that you'll check out the webcast to see some of these features in action. Remember, you can always download a free trial of Rational Application Developer to get a first hand experience playing around with these functions.

Thanks for listening to this podcast on the Rational Application Developer's productivity-enhancing features.