

*** Demo Transcript ***

Hello and welcome to the “Creating a Software Assembly Line for Automated Software Delivery” demonstration.

In this demonstration we will review typical software delivery challenges that companies face today due to increased demands, then we will discuss how IBM Rational Build Forge can automate your software delivery processes, and finally provide a quick demonstration of Rational Build Forge. Let’s get started.

There is a lot of pressure today to get quality software out the door, quickly! Companies are challenged by coping with missed deadlines, trying to maintain quality while adding new features, outdated processes that slow them down, manually tracking data for compliance mandates and dealing with connecting and coordinating remote teams... Automating software delivery processes can help to eliminate many of these software delivery challenges.

So, let’s discuss Rational Build Forge and how it addresses these typical software delivery challenges.

Rational Build Forge allows you to automate software assembly – to create an assembly line that streamlines, automates, accelerates and executes your software assembly processes providing repeatability, reliability, and tracking. Rational Build Forge provides a framework to automate the *entire* software assembly process end-to-end. And Rational Build Forge was designed to integrate with existing scripts and tools, so there is no need to replace those pre-existing assets. Automation is really just one piece of what Rational Build Forge does; it also provides value-added capabilities such as build acceleration and automatic notifications. So the result is faster development cycles, higher product quality, increased staff productivity, and a shorter time-to-market.

Many teams create scripts or may look at some open source tools to try to eliminate their manual error-prone assembly or delivery processes but these solutions often fall short because they don’t address issues such as diverse environments, secure access, a full and documented audit trail and reporting.

Now let’s take a quick tour of Rational Build Forge

You can access Rational Build Forge through integrations with leading IDEs such as the Eclipse framework and Microsoft Visual Studio which provides seamless access to software assembly processes without leaving your IDE. The management console is the heart of Rational Build Forge – this is where you define projects, manage user access and review execution results and so on...

We are now looking at the Rational Build Forge management console. As you can see, the management console is a web interface that is ideal for distributed teams as it provides real-time, consolidated and controlled access to the system anytime, anywhere.

We see a list of projects. A project contains a series of execution steps. We are currently looking at the details screen where you can define project attributes.

Let's look further at an existing project to see the steps. These steps can be very diverse tasks such as executing automated tests, running installers, updating database or web sites or compiling an application. Rational Build Forge also supports more environments for production machines than any other assembly tool as well as offers an optional adaptor toolkit to integrate to other 3rd party tools such as source code control, defect tracking and testing tools.

Let's start this project. Once you execute a project, you can now see each step running. At any point, you can pause, cancel, and resume steps on multiple production machines as well as view the complete status of the processes.

Rational Build Forge automatically captures information on the execution. So we can look at our results and examine why we have a failure such as the one showing on the screen. Rational Build Forge produces a bill of materials which contains information about the steps in the assembly processes and detailed information such as:

- Environment variables used
- Server configuration data
- Any automated tests performed
- Command line captures
- Output by the target machine
- And any messages produced

When you have this level of audit and result information, you simplify compliance management and speed up your problem resolution but most importantly it provides you with the ability to automatically reproduce your assembly process at any time.

At any point, you gain further visibility into the assembly processes through intuitive out-of-the-box reports, RSS data feeds (which provide real-time notification and monitoring) or with advanced reporting using Rational Build Forge Quick Reports which allow you to quickly create, modify and run ad-hoc custom or out-of-the-box reports (helping you to uncover bottleneck, determine trends and pinpoint areas needing performance tuning). The best thing is that no scripting is required!

So what are the actual business benefits of Rational Build Forge? They are VERY impressive. For example, actual improvements to team productivity can be as much as 90% and the reduction of development costs can be up to 65%. Rational Build Forge has a direct impact on productivity, quality, and reduction of costs.

So in conclusion, Rational Build Forge customers are realizing actual business benefit in terms of:

- Reduced production costs due to automation and efficiency
- Improved product quality

- Better distributed and offshore team collaboration
- Faster time to market
- And sustainable development compliance & governance

This concludes the demonstration on “Creating a Software Assembly Line for Automated Software Delivery”.

For additional information on Rational Build Forge, please refer to the following links.
Thank you for attending this demonstration.