The Build Forge Management Console is a Web interface that is ideal for distributed teams enabling secure and controlled access to the system any time, anywhere.

It gives a real-time view of all activities including fine-grained, role-based control to start, pause, cancel and resume builds on multiple production machines.

Let us look at how to create a project within Build Forge. Within Build Forge, processes -- or as we call them, projects -- hold a series of steps necessary to execute a process.

A project could contain a series of automated tests, contain the entire build or release process for a product, or be a simple process which does a simple Web site update, or all at the same time.

By capturing and documenting these vital processes as they evolve, Build Forge makes it possible for teams to share their workload and execute and reproduce builds and distribute releases in a repeatable fashion.

The benefits are by providing a consistent management layer for all your production build and release processes within Build Forge you have complete visibility to your build management operation.

Within the Build Forge system,you define a project as a series of steps and each of these steps within a project is an individual task that contains a set of commands and environment variables which can be distributed to a computer on your network for execution.

As you can see, this project is comprised of multiple steps involved in the building of the application. Tasks such as retrieving your source code, compiling the application, building the installers, running testing and so on.

Steps can run scripts, invoke other applications, move or copy files, export projects for backup and many more things. This enables you to capture, document and standardize your processes that then can be easily reused and replicated ensuring process consistency and significantly reducing new project set up.

One of the Build Forge key differentiators is its flexibility to utilize your own tools and scripts and processes and leverage the positive aspects of your current build approaches.

There are many advantages to taking an existing monolithic build script and breaking it down into smaller more modular steps in Build Forge including the following.

Threading. This gives you the ability to run parts of the build in parallel and reducing your total build times. We can also support the notion of pooling, which allows us to group similar build resources that can be placed in server pools that provide a load balancing and fault tolerance capabilities.

This will enable you to get the most value of your computing resources. Many customers have significantly reduced the build times by leveraging this technology.

Other capabilities we have, the abilities to notify. Notifications allow us to configure step level notifications enabling rapid response to any events requiring action.

And with the ability to reuse these common steps and environments across multiple projects, for example, you could have a generic step which checked out your source code and used your environment variables to then decide, you know, what source code I'm going to access at this time.

And all these capabilities really combine to provide greater build repeatability reducing your build times as well as reducing the time in finding and solving errors -- making software development organizations more productive and more agile.

Once projects are defined the system can run them either manually by unselected options at scheduled intervals or from continuous integration automatically starting builds when source code is changed.

After a project starts, the system runs the test in one or more or several computers, recording the results in its database and monitoring the progress within the system.

Here we're going to do just a manual execution of a project, and when we do that we're presented with a screen which provides us with the environment variables. And these are the environment variables which are used by the process for the run.

For example, we could have a generic compiler set up for our Java versions or here I could select a specific Java version that I wanted to use.

Also I can look at the steps that are part of the project and decide which steps do I need to run and which steps do I not want to run.

For example, If I only wanted to build the executable and run it through unit testing, however, I did not want to promote it to quality assurance and doing post build

activities, I could simply deselect those steps and then when I run the project, Build Forge is going to run the project all the way up to the J Unit test and then stop that project. So a really nice fine granular control over the process.

So, with that step set, then I can go ahead and execute the job. And when I go to execute a job, here we go to what our build dashboard -- this is where I can see all the different builds running in the system.

And for any build running in the system I have the ability to drill down on that build and see the current status of that build and how it's running.

Since the management console is Web based, this gives team members a real-time view of all activities including fine-grained controls to pause and resume builds. And I can look at any step as the build is running to see, you know, what the results were from that step.

For example, here we're going to look and drill down into the build main application step, and then we can see exactly what instructions were run for that step as well as we can look at what environment was set up on the system that we ran on.

So here I'm able to look at all the different environment variables, the Build Forge environment variables, as well as the system environment variables on that system that I ran on.

So with the Build Forge Management Console, managers and developers have simple access to the status of potentially many builds across many systems. This information can increase the productivity significantly in all phases of the project.

So while we have the ability to look at current projects when their running, I can also look at projects that have run in the past, getting a complete view of, you know, all my different projects that are running in the system, you know, what their current status is, as well as which projects are currently running.

Or I have locked any projects where I want to save the results of that and for other purposes, I can have that saved and so the system will not automatically clean up that build.

So Build Forge makes it possible to centralize, automate and accelerate your software development while leveraging your investment in existing tools.

The benefit provided is faster, higher quality development to build release cycles enabling global reporting, centralized tracking and distributed access to your hardware resources.

[END OF SEGMENT]