The **Build Forge Management Console** enables 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.

<0:19>
Let's look at how to create a project within **Build Forge**.

<0:22>
Projects hold the series of steps necessary to execute a process. For example, a project could contain a series of automated tests, the entire build to release process for a product, a simple process for a Web site update, or any combination of all three. By capturing and documenting these vital processes as they evolve, **Build Forge** makes it possible for teams to share their workload, execute and reproduce builds and distribute releases in a repeatable fashion. By providing a consistent management layer to all your production build and release processes, you now have complete visibility to your build operation.

<1:03>
A project is defined as a series of steps. Each of these steps is a set of commands and environment variables which can be distributed to a computer on your network for execution.

<1:14>
This project is comprised of multiple steps involved in the building of an application. Tasks include retrieving the source code, compiling the application, building the installers, testing and so on. Steps can run scripts, invoke other applications, move or copy files, export projects for backup and more.

<1:36>
This enables you to capture, document and standardize your processes so they can be easily reused and replicated. It ensures process consistency and significantly reduces the time to set up new projects.

<1:49>
One of the **Build Forge** key differentiators is its flexibility to utilize your own tools, scripts and processes and to leverage the positive aspects of your current build approach.

<2:01>
There are many advantages to taking an existing build script and breaking it down into smaller more modular steps.

<2:09>
Build Forge includes the following support:

<2:12>
Threading: which provides the ability to run parts of the build in parallel to reduce the total build time.

<2:19>
Pooling: which groups similar build resources into server pools to provide load balancing and fault tolerance capabilities. This enables you to get the best value from your computing resources.

<2:32>
Notifications: which allow rapid response to any events requiring action.

<2:38>
The ability to reuse the steps across multiple projects makes your software development organization more productive and more agile.

<2:47>
Projects can be run manually or as continuous integration builds that are started whenever source code is changed.
Tests can be run in one or more computers with the results recorded in the **Build Forge** database. Progress is monitored via the management console.

<3:05>
We will now manually execute a project.

<3:08>
The first screen shows the environment variables for this run and they can be changed as needed.

<3:17>
The Steps tab allows us to examine and modify the steps for this project as required.

<3:23>
For example, you may want to build the executable and only run it through unit testing. Simply deselect the unneeded steps and run the project. This provides a fine granular control over the build process.

<3:37>

Next, we execute the build job. The build dashboard is displayed and shows all the different builds running in the system.

<3:46>
For any build, you can drill down and see the current run status.

<3:50>
Since the management console is Web based, team members have a real-time view of all activities including fine-grained controls to pause and resume builds. The results from any step can be viewed.

<4:04>
In this example, we drill down into the build main application step, and see exactly what instructions were run.

<4:15>
We can also see how the environment was set up on the build system.

<4:20>
With **Build Forge**, managers and developers have easy access to the status of all the builds across all build systems. This information can significantly increase productivity in all phases of the project.

<4:32>
Build Forge makes it possible to centralize, automate and accelerate your software development while leveraging your investment in existing tools.

<4:41>
This results in faster, higher quality development to build release cycles, enables global reporting, centralized tracking and distributed access to your hardware resources.