

Centralize, automate and accelerate software development  
processes without retooling your environment



**Rational** software

**Build and release process automation  
solutions for software development.**

---

**Contents**

---

<b>2</b>	<b><i>Introduction</i></b>
<b>3</b>	<b><i>Addressing some of today's biggest development challenges</i></b>
<b>5</b>	<b><i>IBM Rational Build Forge products</i></b>
<b>6</b>	<b><i>How the system works</i></b>
<b>10</b>	<b><i>The advantages of IBM Rational Build Forge products</i></b>
<b>11</b>	<b><i>Solutions for software development</i></b>
<b>14</b>	<b><i>Improving quality</i></b>
<b>22</b>	<b><i>Enabling auditing and compliance management</i></b>
<b>25</b>	<b><i>Realizing the benefits</i></b>
<b>27</b>	<b><i>Conclusion</i></b>
<b>27</b>	<b><i>For more information</i></b>

## **Introduction**

IBM Rational® Build Forge™ software is a build and release process automation system that can centralize, automate and accelerate your software development processes. When used in conjunction with IBM Rational software's integrated software development suite, Build Forge software can help companies create an automated end-to-end development environment for streamlined software delivery and sustainable compliance management. For companies with disparate software development applications, Rational Build Forge software provides a flexible integration platform that allows teams to leverage their investments in existing tools for improved team handoffs, automation and traceability.

Rational Build Forge software helps distributed teams stay in sync and work together more efficiently. It provides centralized access across multiple applications and hardware platforms to enable global reporting, tracking and optimized use of hardware resources. With Rational Build Forge products, teams can work smarter, not harder. They can be freed from handling repetitive tasks and instead focus on measuring and improving their build and release processes over time.

This paper describes Rational Build Forge software's approach to managing and automating build and release processes and discusses how the solution can improve software development cycles by integrating teams, processes and systems—resulting in significant improvements in development team productivity, product quality and compliance management.

### Addressing some of today's biggest development challenges

IBM Rational Build Forge software offers advantages for bringing together procedures, people and tools into a cohesive and efficient software delivery system. It addresses three key development challenges: application lifecycle management (ALM), tool standardization and compliance.

The challenge of ALM

Getting a product from initial coding all the way into production involves a complex network of people, processes and technologies that need to be integrated for optimal performance and timely software delivery. Typically, development teams face some common challenges:

- *The groups involved in critical phases—development, configuration management, quality assurance (QA), release and customer support—are separated by organizational boundaries, disconnected toolsets or vast geographies.*
- *Each team has its own processes, many of which are manual and rarely documented, resulting in critical build and release knowledge residing in the heads of a few key staff members.*
- *Essential tools such as bug-tracking databases and source code control systems are often disconnected and contain silos of critical information, but making them work with one another can be difficult and time-consuming. As a result, the organization never fully knows what was delivered to the customer—until it breaks in production.*

To cope with ever-increasing demands for high-quality products at more frequent intervals, development teams need a solid foundation of repeatability, reliability and tracking. Teams seeking to establish this mission-critical infrastructure face two alternatives:

- *Purchase an end-to-end ALM solution from a single provider.*
- *Implement an integration framework that complements and connects existing tools and processes.*

IBM Rational Build Forge software helps teams improve development efficiency, regardless of which path is the best fit for their organization.

To standardize or not to standardize?

For teams that want a robust suite of tools with out-of-the-box integration, the IBM Rational Software Development Platform offers a compelling choice. The combination of IBM Rational ClearCase®, IBM Rational ClearQuest®, IBM Rational RequisitePro®, IBM Rational Build Forge and IBM Tivoli® Provisioning Manager software provides comprehensive automation and control from requirements to production. By purchasing a complete solution from a single trusted provider, companies can get robust supported integrations and simplified customer support.

Other teams find that complete standardization is impractical for their environment. Communication and integration throughout the development lifecycle is essential, but mandating consistency across teams can create political turf wars that can put implementations at risk. Even when teams agree to standardize, an event can occur that introduces inconsistency once more, such as the acquisition of another company or outsourcing part of a project to an organization that uses a different set of tools. Ultimately, some development teams need a flexible architecture that allows them to support heterogeneous toolsets and adapt to future development requirements as needed. The IBM Rational Build Forge system offers a powerful option by providing the ability to automate, integrate and report on any tool in your organization's development environment.

IBM Rational Build Forge software leaps the hurdles involved in setting up an integrated development ecosystem because it is designed to work with all of your current tools and to allow you to add new ones to your environment over time. It provides a consistent interface to functionality across operating systems and hardware. With it, you can launch a Microsoft® Windows® program or UNIX® command from the same interface. Or you can start a process that runs simultaneous tasks on several Apple Macintosh, Microsoft Windows and Linux® platforms. This gives development teams freedom of choice to determine the best solution for their unique environment.

#### Enabling auditing and compliance management

More and more organizations are facing the need to prove that their processes comply with government mandates. To pass an audit, development teams must demonstrate that they have repeatable development processes that have sufficient access controls. They must also consistently document why systems changed and who changed them. Without an automated ALM system, teams can spend countless hours aggregating data from each of their applications to provide the complete development view to auditors.

When connected with the other tools in your development ecosystem, IBM Rational Build Forge software can provide a detailed, self-documenting record of your development processes so you can be ready for an audit without lengthy preparation time. The software captures critical data about exactly what was released, what changed (as well as who changed it and why), and what tests were performed, providing a detailed bill of materials (BOM) that may be used as evidence for compliance and audit checks.

And when used in conjunction with the full IBM Rational suite of software, teams can control, automate and track their development compliance practices from initial coding to production – complete with all the necessary approvals, workflows and audit trails.

#### **IBM Rational Build Forge products**

The Rational Build Forge product line comprises three applications:

- *IBM Rational Build Forge Standard Edition software provides comprehensive build and release process automation, distributed server access and process tracking.*
- *IBM Rational Build Forge Enterprise Edition software includes advanced capabilities such as process threading, server configuration discovery and dynamic server pooling.*
- *IBM Rational Build Forge Adaptor Toolkit software provides out-of-the-box integrations with third-party software development applications along with a flexible application programming interface (API). These capabilities allow clients to create seamless links between their third-party software configuration management (SCM) systems, proprietary tools, and build and release environments for better integration, information sharing and tracking of source code, defects, test results and more.*

### How the system works

At its core, Rational Build Forge software is an automation and process management solution designed to make build and release processes more efficient. Think of Rational Build Forge software as a central repository of processes that can be run, tracked, scheduled and distributed to various computers.

This section describes the scope of automation that Rational Build Forge products provide, and it explains how system components work together.

#### Defining and running processes

Within IBM Rational Build Forge software, you can define a process as a series of tasks called a “project.” Each task within a project contains a set of command lines and environment variables that can be passed to a computer on your network for execution. Once projects are defined, the system can run them at scheduled times or when an authorized user launches them. After a project starts, the system runs the tasks on one or several servers, and records the results in its database. You can program the system so that some tasks launch other projects.

Once a project is finished, the Build Forge software notifies appropriate members of your organization as to the success or failure of the project. A project’s activities can involve traditional build activities, such as compilation of source code, but the Build Forge system is capable of automating and managing many more diverse tasks. Projects can involve complex development processes, extending from source check-out all the way through the build, test and deployment phases. A typical project might perform all of the following tasks:

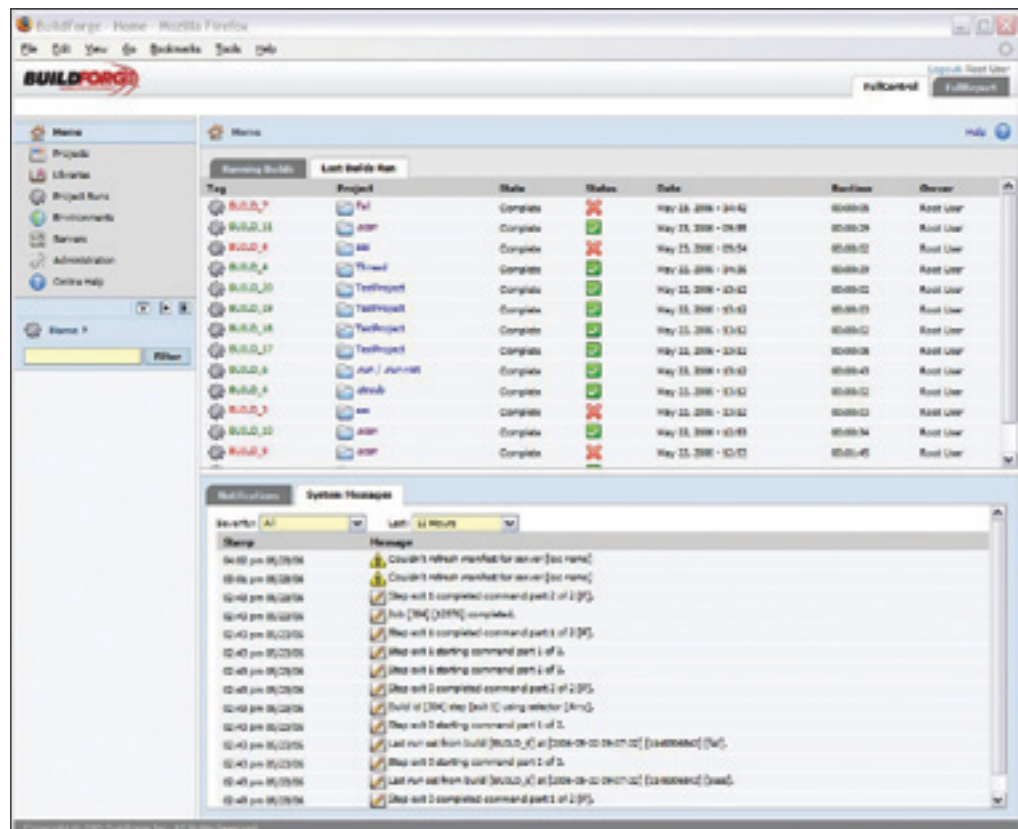
1. *Check out a set of source code files.*
2. *Compile the code and report on progress along the way.*
3. *Run automated unit tests against successful compiles.*
4. *Create an installer.*
5. *Publish the installer to a download site, and notify teams that the installer is available.*
6. *Run the installer to create an installed executable.*
7. *Run automated tests against the executable.*
8. *Report the results of the tests.*
9. *Launch a subordinate project to update standard libraries.*
10. *Promote executables and other files to QA for further testing.*
11. *Deploy finished releases to production environments, such as Web servers or CD manufacturing.*

Many other types of projects and activities are possible. With IBM Rational Build Forge software you can:

- Create Portable Document Format (PDF) files from documentation source files, copy them to a download location on your Web site, update the Web page listing the files, and update the Web site search index.
- Allow individual developers to launch builds of individual components and run a master build daily to compile the entire product, recalling all of the individual builds.
- Run the same process on 5 or 20 different computers, serially or in parallel.
- Allow a project team to run its own builds, while restricting Web server updates to authorized teams.

How the system's components work

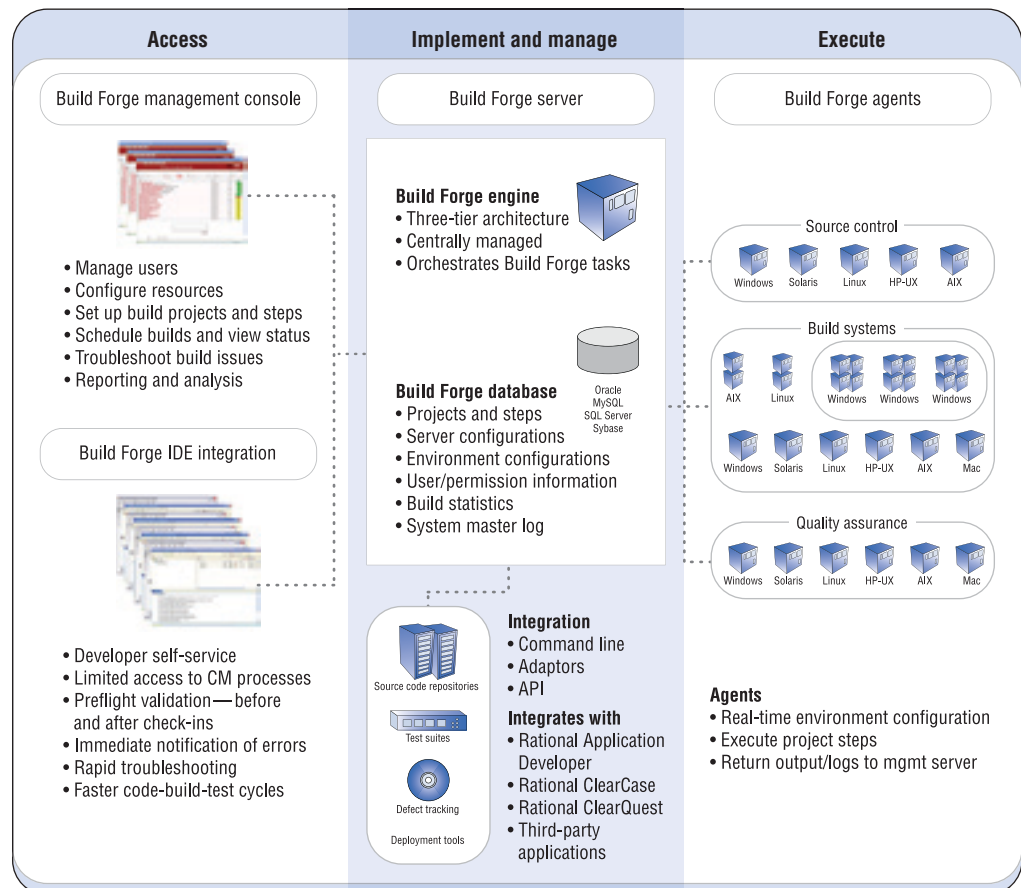
The IBM Rational Build Forge product family started with a simple concept—to provide development teams with a reliable build and release process management system without mandating the use of specific tools. That idea has emerged into a flexible and robust automation infrastructure that is used by distributed development teams around the world. Regardless of the target machine's location or operating system, the Rational Build Forge solution allows you to run arbitrary command lines on multiple servers from a Web-based interface.



The Web-based management console provides a real-time view of all activities — perfectly suited for geographically distributed development teams.

The system contains the following components:

- *The management console provides a user interface to the system. It is a Web-based PHP application that runs on an Apache HTTP server. Through it, you can organize commands into projects and manage the server resources and environment variables that those commands need.*
- *The process engine uses information entered via the management console and stored in the database to communicate with agents, execute project tasks and perform notifications. It acts on instructions from the management console.*
- *The database stores comprehensive project information and tracks each run of a project. User and system actions are stored in it so that auditing and reporting data can be extracted and analyzed.*
- *Individual computers run agents compiled for their operating systems, allowing the computers to respond to instructions from the system. Agents dynamically construct appropriate environments for projects, drawing information from the database so that each command executes in an environment that's fully configured for its needs.*



IBM Rational Build Forge system components



Some examples

With these components in place, the Build Forge system acts as a framework that ties your development resources together. Consider the following development scenarios that can be automated and tracked with Rational Build Forge software:

- *A configuration manager (CM) uses the management console to schedule a daily run of a process that compiles the most recent development version of your flagship product for multiple operating systems. The system knows each step that is required to complete the project, and it creates the appropriate environment at run time based on your specifications. After running some common steps, the project forks into two subprocesses, one for the Red Hat Linux operating system and one for the Microsoft Windows operating system, each of which checks out its own source code and starts compiling the product in parallel.*
- *Throughout the entire process, the configuration management team has a real-time view of the project's progress. If one of these processes fails, the subprocess notifies the developer who was responsible for the work. Once the project is successfully completed, the finished executables are posted to an internal Web server for testing, and the QA team is automatically notified.*
- *A developer launches a test build of the product he or she is working on. The developer is unaware that the machine he or she normally uses is down. The system automatically redirects the process to a different server and notes the actual server used. The developer receives an e-mail message when the process is complete, noting the location of the most recent build artifacts. Neither the developer nor the CM has to respond immediately to the outage.*
- *A webmaster creates a process that collects updated Web page files from a staging server and copies them to appropriate locations on the public server several times a day. Through a single automated process, the webmaster is able to update the files, restart the Web server and archive the old versions of the files to another machine. Now when the webmaster is not available, other team members can execute the same process with minimal training required.*
- *Using many machines, a project team needs to perform a stress test on a product. Working with the configuration management team, the project team schedules processes to run at night using servers belonging to many departments. The project team receives restricted access to the servers for the scheduled tests, so the departments can safely share their resources without compromising the server's environment. Any files installed or generated during the tests may be automatically cleaned up by the system, restoring the server to its original condition.*

### The advantages of IBM Rational Build Forge software

IBM Rational Build Forge software's unique approach to automation offers clear advantages over other methods. These advantages are the result of:

- *Anchoring the Build Forge software on the true deliverable of any software project—the executable.*
- *Providing the Build Forge product with the ability to report what really happens during development.*

Built on the key reliable artifact of development—the executable

The effort required to write code tends to overshadow the process that produces a final product from that code. The processes involved with building software have typically been seen as necessary evils that occur just before the product is ready to ship, and they are handled by understaffed configuration management or build teams within the organization. Yet, the executable is the only real deliverable that the product development team produces. Consider these examples:

- *Developers can write code for months, only to find that an improperly built executable fails to function correctly for customers.*
- *Your source control system tells you that a defect is fixed, but the branch never makes it into the final release. As valuable as the SCM system is for tracking code, it can't show you what really made it into the product when you are trying to track whether a particular issue was resolved.*
- *Your bug-tracking database relies on human input when it insists that a problem is fixed. The executables that your team produces become the definitive product of record. To be accurate, you need a way to manage your development that derives its data from the executables. When you use Rational Build Forge software to create your products, the system can report on the bug fixes that went into the executable, and it can reveal the code that was used to make the bug fixes. It ties your development process together, and it allows you to track backward from problems to the code that produced them. As a result, build and release management processes can be managed and coordinated closely with other phases of the application delivery process to minimize development mishaps.*

#### Reporting what really happened

Rational Build Forge software bases its reports on detailed records obtained from specific process runs, whether that process is a software compilation, an automated test or a Web site update. This differs from other document-based development systems that require manual input as their information source. Consider the benefits of an integrated development environment using Rational Build Forge software:

- *Your source-control system can tell you what code was checked in, but Rational Build Forge software can tell you what code was actually included in the release.*
- *Your defect-tracking system can tell you which fixes were intended to be in a release; Rational Build Forge software can tell you which fixes actually made it in, and the software can run automated tests to validate the fixes.*
- *When a customer calls with a problem in a specific software release, you can refer to Rational Build Forge software to determine what was included in that release.*
- *When you encounter a problem on your Web site, Rational Build Forge software can tell you exactly what changed and help you roll back to the earlier version if needed.*

#### **Solutions for software development**

Whether you want to speed up your development process, enhance the quality of your products, enable distributed teams or fulfill auditing and compliance requirements, Rational Build Forge software can help.

#### Accelerating your processes

Rational Build Forge software excels at speeding up your development process with features that address typical bottlenecks, including:

- *Ineffective handoffs of a project from one functional group to another*
- *Slow, sequential processing speeds*
- *System downtime*
- *Underutilized hardware*
- *Opaque batch processes*

This section describes how the features of the Rational Build Forge system address these problems.

#### Faster handoffs and fewer fumbles

Rational Build Forge software's self-documenting features help ensure better group coordination, whether people work across a building or across an ocean from one another. Teams need not worry about passing the appropriate information to the next team downstream in the cycle; they can concern themselves with handling their portion of the work appropriately and encapsulating the information in the system, knowing that the system will store it and pass it along. The following features facilitate this process:

- *BOM: The system organizes a description of the contents of the process run, including checkpoints to show file changes, and stores them in a package.*
- *Logs: The system logs activity and command output, and makes those logs available across your organization according to the security settings you specify for your projects.*
- *Notes: The system allows users to enter notes relevant to projects and tasks within projects.*

These features combine to provide a clear picture of exactly what happened during a development process. The system tracks who did what on which machines, what the results were and how long the process took. Such tracking features help to efficiently comply with auditing rules. For more on this topic, see the "Enabling auditing and compliance management" section of this paper.

#### Increased speed and uptime

When you use Rational Build Forge software, you can take advantage of increased speed and uptime from concurrent processing and server pooling. These features allow you to:

- *Group several machines into a server pool, and then start several process runs. As soon as one server is fully occupied, the system redirects processes to other servers in the same pool.*
- *Use a single process to launch concurrent tasks on several machines, enabling quicker results.*

- *Launch processes on different machines to build different versions of an application—for several different operating systems, for example—in their appropriate native environments.*
- *Hand a process over to another member of the server pool if the default machine specified for the process goes down when the Rational Build Forge software attempts to run it, enabling the process to run uninterrupted.*
- *Run the same task on every member of a pool at once. For example, if you want to deploy a code update to all of your pooled Linux servers, a single task can do it, and the updates occur in parallel.*

#### More speed with less hardware

When you implement IBM Rational Build Forge software, you make it possible for your departments to share their hardware resources in a controlled fashion. Without this centralization, a department often dedicates hardware to a single project, resulting in idle, underutilized hardware over time. But with Rational Build Forge software in place, the following scenarios become possible:

- *A department that needs to occasionally run some processes on a Macintosh system can be provided with limited access to one—without needing to physically access or modify the system. Access can be restricted to specific tasks and cleanup can be automated. If the use becomes too frequent, access can be revoked.*
- *One department's servers can be set up as backups for another's, so that if key servers in a department are down, the system automatically uses the secondary machines until the preferred machines come back online.*
- *A group of machines can be designated as a pool of available servers, ready for any process that needs them, regardless of the department or project involved.*
- *A machine can be set up to offer only a portion of its processing power to other departments.*

These scenarios can translate into increased processing speed and lower costs for your organization.

Process clarity and analysis for continual improvement

Because Rational Build Forge software can track all of the activities you perform from coding to production, it contains a wealth of information to help teams improve their effectiveness. Rational Build Forge software generates a collection of prebuilt reports that you can use to analyze your processes and continually improve them. In contrast to using an extensive batch file or script that provides little visibility to your processes, with Rational Build Forge software you can:

- *Immediately pinpoint errors without searching through extensive log data.*
- *Review individual processes to see how long they typically take, looking for anomalies or trends.*
- *Track down where problems originate—across departments or within them.*
- *Track server utilization over time.*
- *Identify areas of the code base that are changing most frequently, and pinpoint error hotspots to improve project planning.*
- *Generate your own reports from the Rational Build Forge database using the system's documented schema.*

#### **Improving quality**

Speed alone is not the whole story. When you implement Rational Build Forge software in your organization, you may experience a variety of quality improvements that result from having a consistent build and release automation system, standardized development processes and better team communication.

Repeatable processes

IBM Rational Build Forge software brings consistency to your build and release processes, running them the same way every time. When you drive your activity from Rational Build Forge software, the system can repeat a project run using values fed to the system from a previous build, or it can display values that you can customize at run time. The Build Forge solution is ideally suited for iterative development and for replicating past processes to test them or diagnose problems. You can even create shared libraries of processes that can be reused from project to project. This provides valuable quality controls through standardization and reduces the time required for new-project setup.

#### Knowledge retention

Turnover is a fact of life in all organizations, but Rational Build Forge software can help you survive it. The system's centralized knowledge base automatically stores process information within the company's infrastructure, protecting information investments and dramatically lowering the learning curve for each stored process. The Rational Build Forge knowledge base allows team members to:

- *See the current definition of a process.*
- *See the results of previous runs of a process.*
- *See the changes in a process definition over time.*
- *View notes that explain why a process was changed.*

The system creates a repository of process knowledge and automatically archives it so that processes are executed in a standardized, consistent and repeatable fashion.

#### Reducing human error with automation

Rational Build Forge software gives you the tools you need to implement a comprehensive build and release automation system. After you design a process, you can easily automate it, and then share the automated process with the rest of your organization.

The system supports automation by allowing you to:

- *Schedule regular process runs.*
- *Drive activity based on the success or failure of a run, or a portion of a run.*
- *Drive activity based on changes to source code.*
- *Deliver automatic reports based on the success or failure of a process.*
- *Provide an automatically generated BOM with each run, so that you know its components, why it was run and who made changes.*

With these features in place, each team becomes more productive.

- *The development team gets more frequent code builds, providing them with immediate feedback on the success or failure of their coding efforts to keep them on the right track.*
- *The QA team gets details on the availability and contents of each release and can start from the results of automated tests to evaluate a release's problems. Through the BOM, the QA team knows what changed and what tests should be run. With automated tests, many routine problems can be eliminated from consideration.*
- *The configuration management team spends less time managing day-to-day operations, leaving more time for optimizing development processes.*
- *The IT team can use IBM Rational Build Forge reports to help assess current system needs and estimate future demand.*

Centralized control and distributed access

Communication among teams is often a major impediment to timely releases. Without an effective build and release automation system, the configuration management team's ability to respond to demands for new builds, new tests and new projects can gate the development process. In addition, developers who are disconnected from the build process often lose productivity while waiting for results. Similarly, if a project is nonrepeatable and undocumented, information about varying product versions is difficult to determine, and reproducing a customer's problem can become the most time-consuming part of fixing it.

Rational Build Forge software supports distributed development teams by providing centralized control and secure, role-appropriate access at the same time. You can bring processes from many departments into one central reporting system; then you can spread access to the processes (and the information they generate) across your organization as widely as desired. A centralized configuration management group can create and test a process, and then grant the ability to use that process to appropriate groups of people who may not all belong to the same department. And individuals not part of the formal configuration management team can create their own processes and share them company wide. At the same time, you retain control over who can run an automated process, view a report or define a new process.



Using Rational Build Forge software, a developer can launch a build project, provide instant feedback and test a new feature or a fix, but he or she can only initiate those processes that your organization has decided he or she should have access to. This kind of self-service makes developers more productive and reduces the time your CMs spend responding to requests. You can extend this idea to any process that you automate.

Providing agility through more frequent test cycles

Though automated testing can never replace the insight of a QA team, Rational Build Forge software makes it possible to incorporate efficient automated testing into your overall build and release processes, and to share those test results across your organization. Moreover, the self-documenting nature of the Rational Build Forge system helps to make each test cycle more productive, as each build contains information about what happened in it and what its executable contains. Any iterative development process—such as Agile Development and other methodologies—demands ongoing code-build-test activity to provide frequent feedback to the development and testing teams. Rational Build Forge software provides powerful automation, scheduling and source control integration that enables teams to implement continuous integration.

Through IBM Rational Build Forge software, a tester can start to evaluate a product with many details already determined.

- *Problems that can be detected by automated tests can be fed back to development before the QA team sees the release, so that manual testing begins only after automated tests have passed.*
- *Previously encountered errors can be quickly retested, speeding and improving regression processes so that developers and testers can focus on fixing new problems with less concern about introducing new ones.*
- *Stress testing and loading can be performed via automation.*
- *Rational Build Forge software provides infrastructure for kicking off the tests on all the target machines and collecting the results.*

- *Rational Build Forge custom filters can preprocess test results, enhancing reporting with detailed lists of warnings or failures.*
- *The BOM report details the bug fixes that a build contains, so each can be tested or matched against automated test results.*
- *Rational Build Forge software can be integrated with your source control system and bug-tracking database.*

Possible uses for IBM Rational Build Forge software for a test group include:

- *Perform smoke tests. After every build, launch a subset of basic tests and report on their success or failure to determine whether the build still works after changes.*
- *Automate installations of completed builds on multiple target platforms. You can use Rational Build Forge software to configure your test environment and notify testers that machines are ready for test scripts to be run.*
- *Determine which tests to run. Projects in the Rational Build Forge system are tagged with different classes. You can use the project class to decide which tests to launch.*
- *Skip the testing when a developer just wants a quick scratch build.*
- *For a scheduled build, run a smoke test to ensure that nothing major is broken.*
- *For a production build, run a full test suite—perhaps including scaling or load testing—that may require a significant part of your system resources.*

These features help make it possible to build more and more efficient testing into your processes, thereby building higher quality into your product releases.

Designed for integration

At heart, IBM Rational Build Forge software is a solution for tying applications together to accomplish larger development tasks. With Rational Build Forge software, you can easily mix applications, operating system commands and batch files or shell scripts within your projects.

With its unique ability to encapsulate command lines, Rational Build Forge software creates the necessary environment for each command on the target computer system before actually sending the command. And you can define the environment for a command to integrate it with the other pieces of software you want to use.

Beyond its basic function of tying together applications, Rational Build Forge software can also integrate with other solutions.

- *You can integrate your existing user data via the Lightweight Directory Access Protocol (LDAP) standard, so that you can avoid creating an additional set of user logins for the new information system.*
- *IBM Rational Build Forge software offers out-of-the-box adaptors for major source code control, defect tracking and test automation systems to launch processes and update information driven by development activities.*
- *An API allows you to drive IBM Rational Build Forge functionality from other applications within your information system, giving you fine-grained control over your processes.*

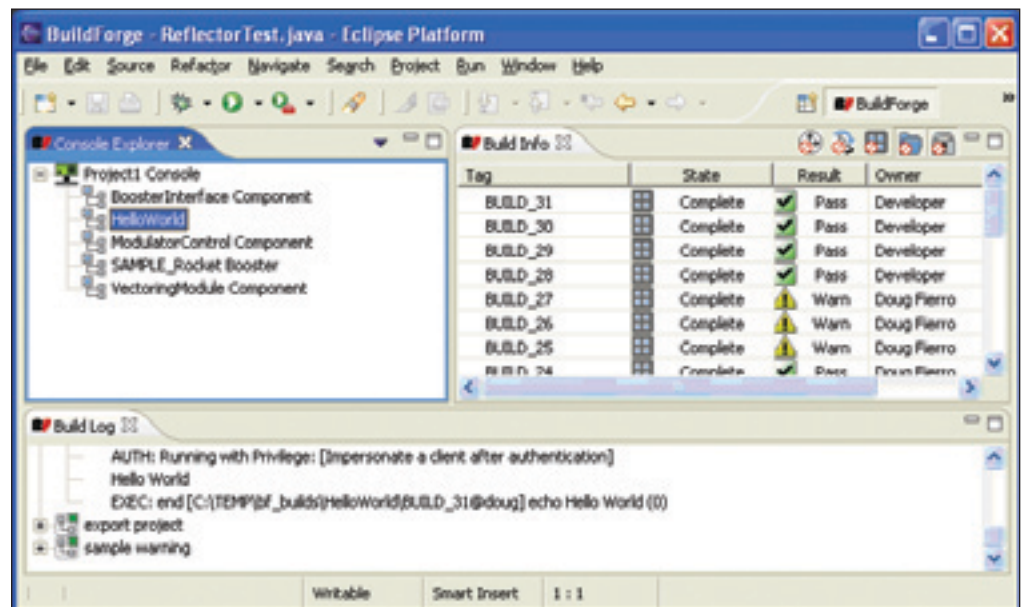
With these features, Rational Build Forge software remains an integral part of your development environment and a well-behaved member of your corporate information system.

Out-of-the-box value: IBM Rational software development suite

Both IBM Rational Build Forge Standard Edition and IBM Rational Build Forge Enterprise Edition software come equipped with tight out-of-the-box integrations with the IBM Rational Software Development Platform solution to provide immediate customer value. Using the products together, companies can manage, automate and track their entire development lifecycle from requirements to production. Integrations include:

- *IBM Rational ClearCase software. Build Forge software links directly to Rational ClearCase Versioned Object Bases (VOBs) in Unified Change Management and base Rational ClearCase environments for continuous integration and detailed source code tracking for each build or release. Through bidirectional information sharing, Rational Build Forge build projects and build artifacts can be stored as part of a Rational ClearCase baseline.*

- *IBM Rational ClearQuest software. Rational Build Forge software automatically detects Rational ClearQuest activities for a specific build session based on the view being used. Upon a successful build exit status, Rational Build Forge software can automatically advance the state of the associated defect to a resolved status, and reference the build record from the Rational ClearQuest application. Rational Build Forge software also creates a deployment unit that may be handed off to deployment engines such as IBM Tivoli Provisioning Manager software.*
- *IBM Rational Application Developer software. Rational Build Forge software breaks down the walls between development and configuration management by providing developers with limited access to production build processes from within their Rational Application Developer integrated development environment (IDE). This allows developers to run preflight builds and get immediate status in advance of a group build for higher-quality and more timely releases.*



The system communicates directly to the IBM Rational Build Forge console from within the IDE.

#### Linking third-party applications

The IBM Rational Build Forge Adaptor Toolkit product allows users of third-party software, such as source code managers, defect-tracking managers and test automation tools, to create seamless links between their software configuration management and build environments for increased efficiency and tracking of source code, defect and test changes. Third-party adaptors are available for CVS, Perforce SCM, Borland StarTeam, Microsoft Visual SourceSafe, Subversion and Bugzilla environments . You can modify these adaptors or create your own integrations with homegrown or third-party software development tools.

The Rational Build Forge Adaptor Toolkit solution provides many integration options to automate and optimize your development processes, including:

- *Monitor source code changes. IBM Rational Build Forge adaptors provide continuous monitoring of the third-party source repository, and they execute builds automatically when a change occurs. The system directly monitors source changes and gathers metrics to provide detailed repository state tracking for each build. Using the Rational Build Forge management console, you can see where changes occurred—from submission comments to actual file differences—on a per-build basis.*
- *Automate defect tracking and tests. Rational Build Forge defect-tracking adaptors help you track defects and report the state of a given defect to a BOM, such as “verified” or “closed.” The defects are automatically reported to help you know which defects were fixed within a specific build.*
- *Automate test case runs and capture test results to the BOM with provided test adapters.*

Adaptors allow you to correlate source code changes, defects and tests with specific builds for a detailed understanding of the build components. IBM Rational Build Forge software captures key build statistics—including changes made, build time and date—and stores this information in a central location for quick access.

#### **Enabling auditing and compliance management**

As discussed earlier in this paper, compliance is a topic on the minds of most development teams. Even if your team is not responsible for the company's financial systems, your group delivers products or services that are critical to the company's revenue streams, and you should be prepared to be audited.

In the aftermath of the Sarbanes-Oxley Act, many companies have decided that standardized, repeatable and documented development processes are an essential business best practice. The key is to make the collection of this audit data as effortless as possible. IBM Rational Build Forge software captures critical data and provides traceability from initial coding to production that can be used as evidence for compliance and auditing purposes.

When you use Rational Build Forge software to run a build and release process, the system automatically tracks your process from start to finish. The more processes you build into the Rational Build Forge system, the clearer the view of your development environment.

The system retains version information on all of your processes. For each product iteration, you can see what changed, who changed it and why it was changed. And the system provides a BOM with each completed process run.

#### Automatically tracking your processes

Even the simplest of scripts becomes a valuable compliance tool when you run it through Rational Build Forge software, because the system stores data about every project run. When run within Rational Build Forge software, a batch file or shell script that copies files from a source location to a Web server quickly becomes much more than a script.

Consider these examples:

- *The system reports success or failure via e-mail as well as on a Web-based dashboard.*
- *Attempted commands and resulting output or error messages are stored in the logs.*
- *The system can schedule the script for repeated runs, guaranteeing that the standard process is executed the same way, every time, and occurs as often as needed.*
- *The system can find available server resources to run the script, rather than relying on the availability of a single machine.*
- *The system logs who started the script.*
- *Other user actions, such as changes to the script, are logged.*

Additionally, Rational Build Forge user notes help document the reason for each change that occurs.

When integrated with your existing tools, Rational Build Forge software can transform information silos into useful compliance data that is available at your fingertips.

#### Versioning your processes

When you add a process into Rational Build Forge software, many pieces of information about that project are captured: the steps in the project, the environment variables needed by the project, and the server (and/or server pool) that the project should run on. The project record can be updated at any time as you change your process; the system retains the current version of the process as the default set of instructions for that process.

Each time you run the project, the system stores a copy of run-time information. You can review the record of the project run to see exactly what steps were performed on a particular occasion, even if the process definition has changed. Further, the system can re-create an earlier run, following the instructions from that run and ignoring intervening changes.

Thus, any process added into Rational Build Forge software immediately becomes repeatable and traceable. If changes to a process result in quality problems, you can return to the last good state. Although the system automatically stores process records within its own database, you can archive the process instructions in your source control system. This allows you to store the instructions for a process along with the files used to create the product. You can use Rational Build Forge software to automate the archiving process as well.

Self-documenting systems via the bill of materials (BOM)

IBM Rational Build Forge software includes a configurable BOM. For every process that it runs, the Build Forge system automatically produces a BOM that provides a concise package of information about a process run that can be used to review the results of the process. The BOM also serves to document the process for various consumers of the product.

The system automatically includes certain items of interest about every build in its BOM.

- *Using provided commands, you can add tasks to a process that cause additional information to be written to the BOM. For example, you can store information about the files in the process working directory; then you can show how the process changed those files at various checkpoints.*
- *The BOM acts as a ticket passed from one group to the next in the process. Without a BOM, a build is a mystery package; with it, a team can quickly evaluate what a new build means to them.*
- *The BOM can even be used as a working outline for the technical publications team as they start to document a release.*



The BOM acts to encapsulate the process and its end-to-end results in one convenient package, suitable for auditing or verifying compliance.

### **Realizing the benefits**

Because of its flexible architecture, reuse of existing scripts and out-of-the-box integrations, IBM Rational Build Forge software can be implemented quickly and readily evaluated in your environment. To get initial benefits, development teams can begin with basic build and release automation of a single project; then they can fine-tune the project over time with additional optimizations. Once initial success has been achieved, the implementation can be extended to include additional projects and teams.

### **Evaluating the system**

The best way to find out whether IBM Rational Build Forge software is for you is to download, install and test the solution yourself. While the system is designed to operate on multiple machines working in parallel, you can test most of its functionality on a single machine after running a pair of installation programs.

The IBM Rational Build Forge Implementation Guide includes instructions for setting up an evaluation system, and it features a tutorial for learning how to use the solution. Try out the Rational Build Forge solution on representative projects from your own development environment—from simple to complex build and release tasks. That way, you can validate that the product works in your environment. Pilot implementations can typically be completed in two to four weeks.

Product evaluations require prequalification. If you are interested in conducting a product trial or learning more about IBM Rational Build Forge software, contact your local IBM Rational account representative or your IBM Rational Build Forge account executive. Or e-mail [sales@buildforge.com](mailto:sales@buildforge.com).

Implementing the system: phased deployment

When you implement the Rational Build Forge system, you won't need to change your processes overnight. It's neither necessary nor desirable to implement it all at once. Instead, you can start by adding a single process to the system. As soon as you are happy with the results, you can select key processes to add, and then prioritize others.

In a large organization, a single department might adopt the system, and then spread it to others after demonstrating its worth. The system can grow organically as users see its value, rather than being reluctantly adopted because its use was mandated by top executives.

Each process that you add to the system becomes a repository of controlled knowledge about your organization that you can hand to the people who need it, while restricting it from those who don't need it.

### **Conclusion**

IBM Rational Build Forge software provides a complete build and release process management and automation system that can enable development teams to be more productive and rapidly deliver high-quality products to market. Customers using the Rational Build Forge system begin by adding key processes into it; after they analyze the return on their investment, they are motivated to add more processes to the system. The system's ability to coexist with legacy tools makes adoption less invasive than retooling your entire environment. As your teams use the system, they find that they retain control over processes and machines, but the safeguards built into the system enable them to disseminate information and tools further than ever before. The system makes it possible to integrate teams and tools of varying origins; yet it also allows users to store their information in a central repository so that the scattered components of a complex development process can be brought into sharp focus.

Let IBM Rational Build Forge software help you take your development to the next level.

### **For more information**

If you are interested in conducting a product trial or learning more about IBM Rational Build Forge software, contact your local IBM Rational account representative or your IBM Rational Build Forge account executive. Or e-mail [sales@buildforge.com](mailto:sales@buildforge.com).



© Copyright IBM Corporation 2006

IBM Corporation  
Software Group  
Route 100  
Somers, NY 10589  
U.S.A.

Produced in the United States of America  
06-06  
All Rights Reserved

Build Forge, ClearCase, ClearQuest, IBM, the IBM logo, the On Demand Business logo, Rational, RequisitePro and Tivoli are trademarks of International Business Machines Corporation in the United States, other countries or both.

Linux is a registered trademark of Linus Torvalds in the United States, other countries or both.

Microsoft and Windows are trademarks of Microsoft Corporation in the United States, other countries or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Other company, product and service names may be trademarks or service marks of others.

The information contained in this documentation is provided for informational purposes only. While efforts were made to verify the completeness and accuracy of the information contained in this documentation, it is provided "as is" without warranty of any kind, express or implied. In addition, this information is based on IBM's current product plans and strategy, which are subject to change by IBM without notice. IBM shall not be responsible for any damages arising out of the use of, or otherwise related to, this documentation or any other documentation. Nothing contained in this documentation is intended to, nor shall have the effect of, creating any warranties or representations from IBM (or its suppliers or licensors), or altering the terms and conditions of the applicable license agreement governing the use of IBM software.