IBM Rational® Build Forge® enables high-performance builds and agile software development. Its adaptive build-and-release management framework helps development teams standardize repetitive tasks, manage compliance mandates, and share information.

This presentation will cover an overview of the available editions of Build Forge, key capabilities of the system, architecture, and the key roles and steps in creating a project in Build Forge.

IBM Rational Build Forge editions

Key features	Standard Edition	Express Edition	Enterprise Edition
Support for remote sites	X	X	X
Sophisticated log file filtering and searching	X	X	X
Integration with external tools and technologies	X	X	X
Web console for user access and administration	X	X	X
Detailed audit logging and bill-of-materials report	X	X	X
Reports	X	X	X
Scheduled, as-needed and continuous integration builds	X	X	X
Integrated development environment (IDE) integrations	X	X	X
Application programming interface (API)			X
Distributed multiplatform builds, build acceleration	X	X	X
Robust permissions and security model	X	X	X
Out of the box integration with other IBM Rational products	X	X	X
Choice of database repository	X	X	X
Add-on advanced reporting: IBM Rational Build Forge Quick Report	X		X
Automated process for build and release management	X	X	X
Dynamic server management			X

Overview for V7.0

3

© 2008 IBM Corporation

The standard edition provides a consistent management layer for your production build and release processes.

The express edition is an enterprise-grade build and release framework for small and midsize businesses. This version allows for easy installation and quick adoption.

The enterprise edition offers full build management with the features shown here including an API, dynamic server management, and scalability to large teams. It automates and accelerates build and release processes through server pooling and fault tolerance for Agile development and streamlined software delivery.

IBM

Build Forge: System overview

| Development | Source control | Product build | Quality assurance | Package | Release |

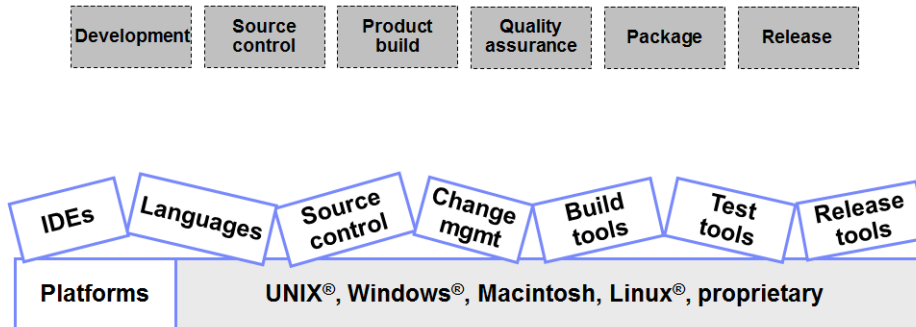Overview for V7.0

© 2008 IBM Corporation

4

Developing and delivering software is a complex set of tasks involving a network of people, processes, and technologies that need to be integrated.

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.

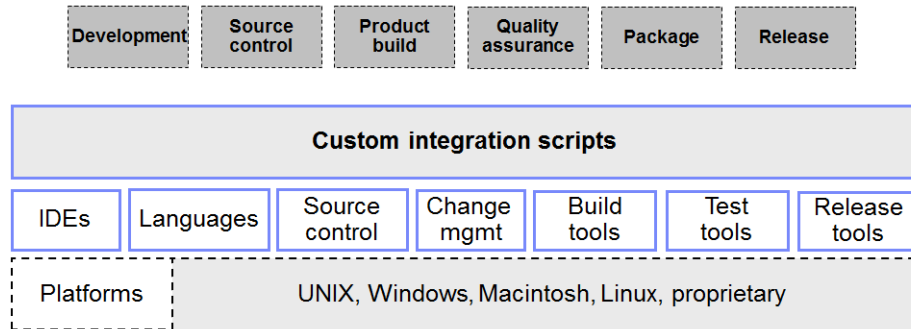Build Forge helps development teams bring a product from initial coding all the way into production.

Build Forge: System overview

| Development | Source control | Product build | Quality assurance | Package | Release |

IDEs · Languages · Source control · Change mgmt · Build tools · Test tools · Release tools

Platforms: UNIX®, Windows®, Macintosh, Linux®, proprietary

Each team has its own processes, which are often manual and rarely documented. 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 each other is difficult and time-consuming.
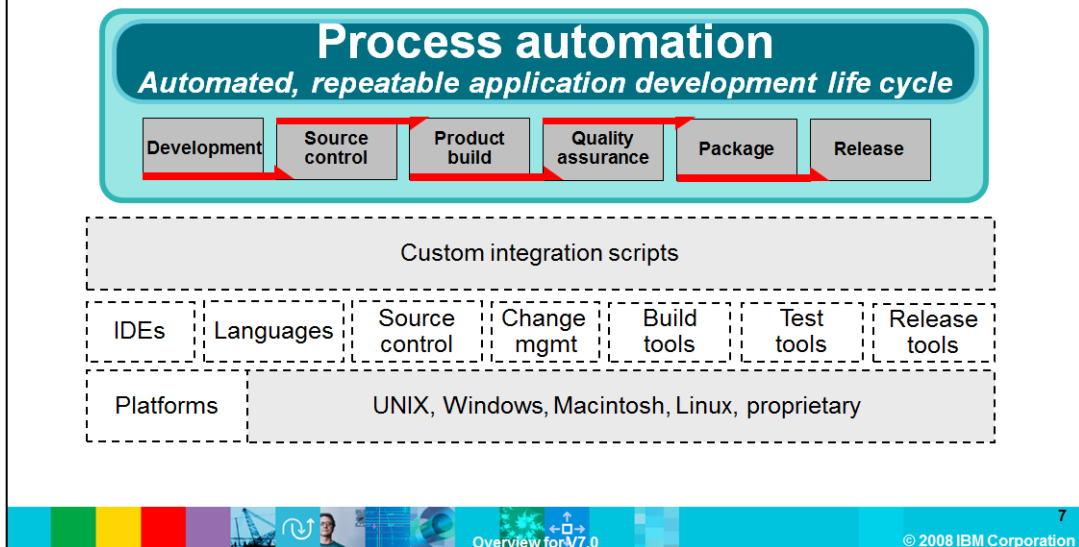
However, the ever-increasing demands for high-quality products at more frequent intervals create a need for a solid foundation of repeatability, reliability, and tracking.

Build Forge: System overview

Development	Source control	Product build	Quality assurance	Package	Release

Custom integration scripts

IDEs	Languages	Source control	Change mgmt	Build tools	Test tools	Release tools

Platforms	UNIX, Windows, Macintosh, Linux, proprietary

Overview for V7.0

6

© 2008 IBM Corporation

Many teams have scripted custom integration wrappers to share information among some of their tools to remedy these issues. However, these scripts typically do not provide the ability to view and control builds, and generally there are multiple versions to support multiple platform product lines. Plus, you create a non-core product that requires time and key resources to maintain and enhance.

Build Forge: System overview

Process automation
Automated, repeatable application development life cycle

| Development | Source control | Product build | Quality assurance | Package | Release |

Custom integration scripts

| IDEs | Languages | Source control | Change mgmt | Build tools | Test tools | Release tools |

| Platforms | UNIX, Windows, Macintosh, Linux, proprietary |

Build Forge provides a framework to automate the entire process, end-to-end. This means not just automating of individual tasks, but also the hand-offs between the various steps in the process. Build Forge was designed to integrate existing scripts and tools, so there is no need to replace your pre-existing assets. Build Forge offers a comprehensive Application Development Process Management solution that provides complete management and control of the application development life cycle. Build Forge automates, standardizes and optimizes complex processes, integrating diverse toolsets to deliver a repeatable and reliable application development life cycle process.
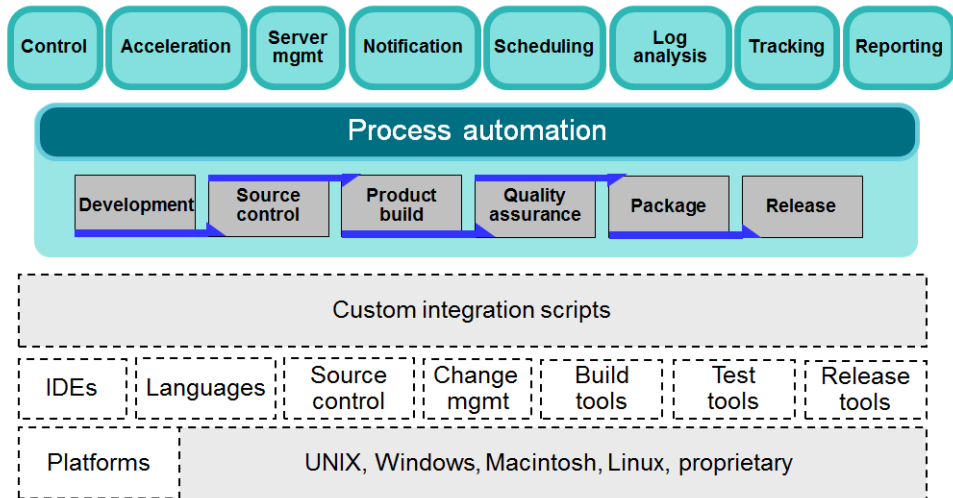
Automation is really just the first piece of what Build Forge does. Build Forge also provides value-added capabilities including build acceleration, server management, automatic notifications, scheduling, robust permissions and security models, and much more.
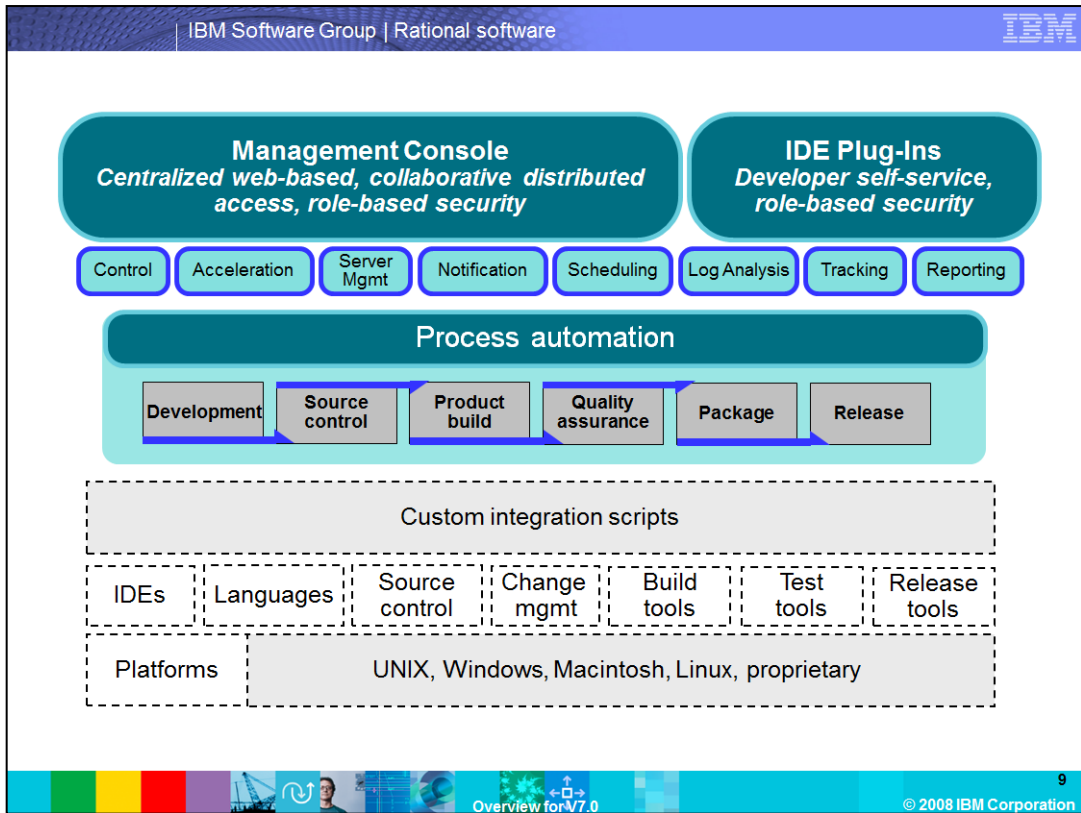
These capabilities are accessed through the management console: a secure, Web-based interface that allows managed delegation of tasks to other groups, even offshore teams.
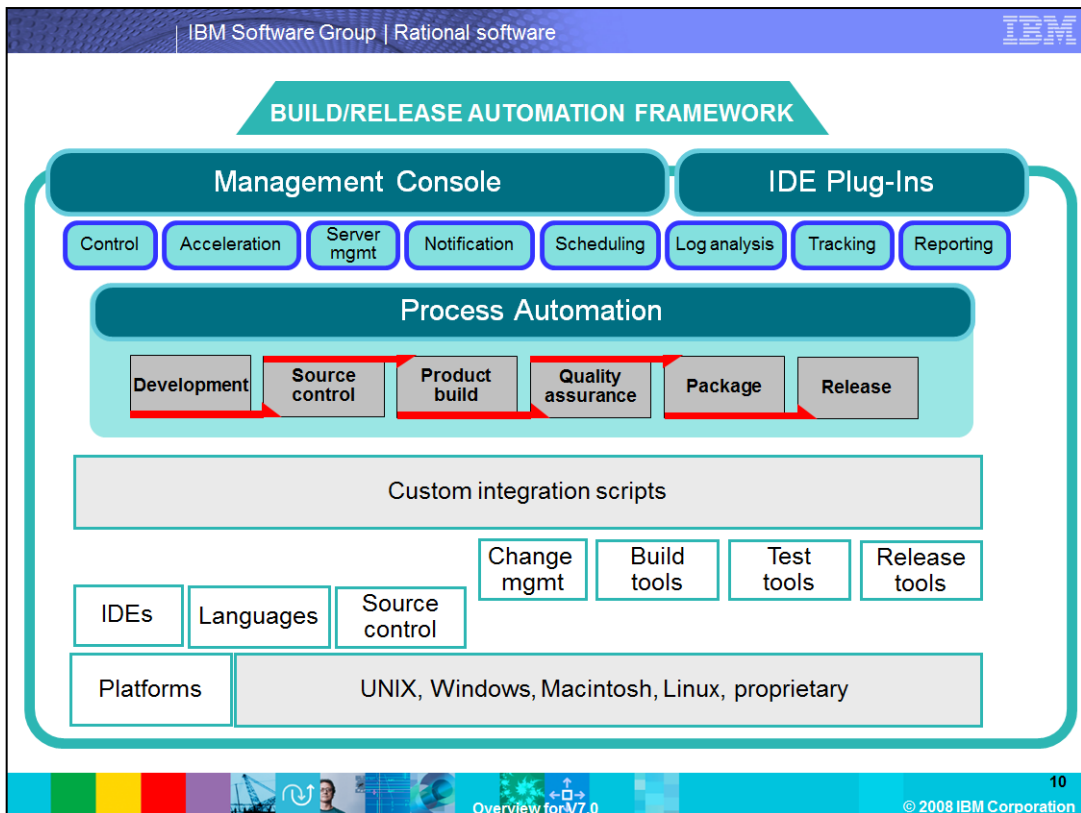
The results are faster development cycles, higher product quality, increased staff productivity, and a shorter time-to-market.

IBM Rational Build Forge provides reliable, high-performance builds for agile development and streamlined software delivery. This framework increases team efficiency with centralized build and release management, using the tools you have in place today. It helps to accelerate build and release cycles through iterative development, parallel processes and efficient use of hardware. It improves product quality through consistent, repeatable processes and early detection and troubleshooting of errors, and much more.

Section

Build Forge architecture

Overview for V7.0

11

© 2008 IBM Corporation

This section will cover the Build Forge architecture.
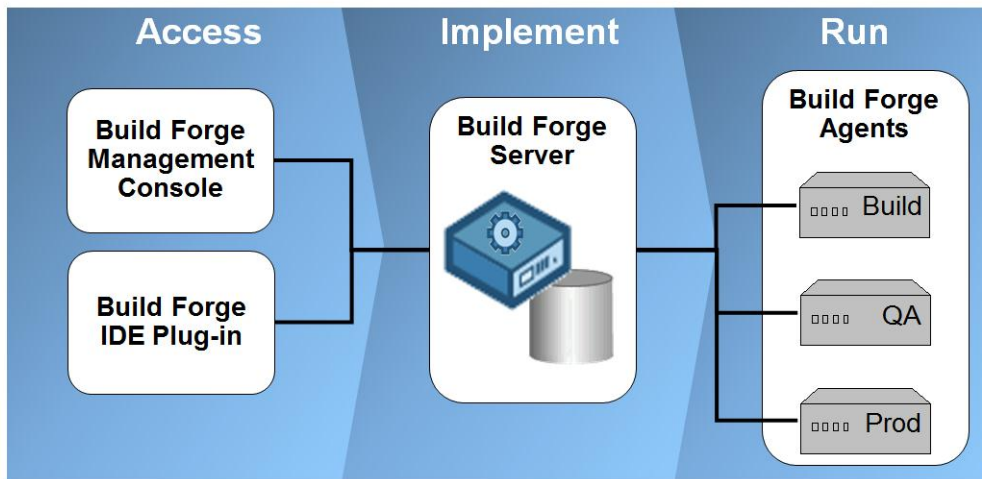
The Build Forge Architecture consists of one or more server machines, performing the various tasks of the build system. The Build Forge Server is at the heart of the system; managing the worker systems, organizing the projects, and managing user access. Users communicate with the Server using the Management Console or the IDE plug-ins. The Build Forge Agents, running on the worker systems, take orders from the Build Forge Server and run the specified commands on the worker system.
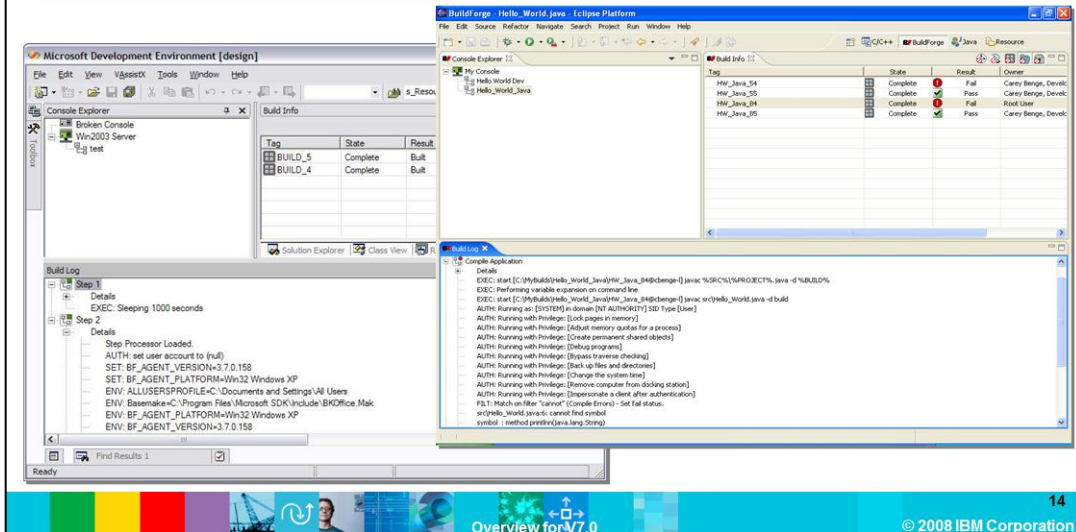
The Build Forge Management Console is the main control UI for the system. It is a PHP-based Web application running on an Apache HTTP server. Through it, you can manage users, configure resources, organize commands into projects, and manage the server resources and environments that those commands need.

In many organizations, only Administrators, Configuration Managers, and Build Engineers use the Management Console. For unit test and integration builds, users interact with Build Forge using the IDE, and are informed of build status by e-mail notification.

The Build Forge IDE plug-in provides seamless integration with IDEs providing centralized access to build projects, status, and log file results directly within the IDE. The IDE plug-in also provides the ability to selectively choose local file changes from within the IDE and preview the build results in a remotely staged, pre-configured environment before committing the file changes to source code control.

In the middle is the Build Forge Server. The Build Forge server is a centrally managed, 3-tier architecture. The server oversees all of the activities on the managed worker systems, organizes the projects, manages user access and permissions, logging, and audit trails.

The system runs on various databases. Build Forge can use various databases, including MySQL, Sybase, Microsoft SQL, Oracle, and IBM DB2. The Build Forge Windows edition includes DB2 Express.
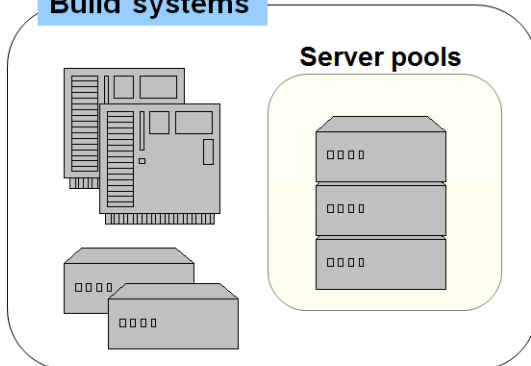
Build Forge also provides mechanisms to seamlessly integrate your existing IDE, SCM tools and repositories, test suites, and deployment tools. This is done by means of command-line integrations, API-level integrations, and continuous and agile development adaptors.

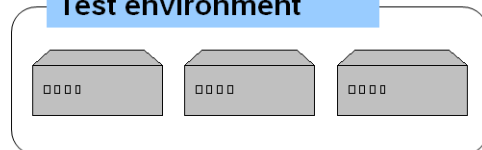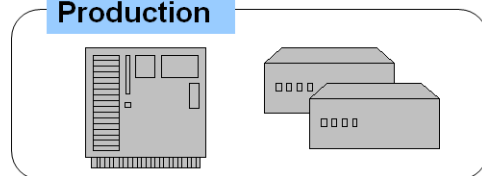Finally, Build Forge agents are deployed across worker machines. The agent is a small process that does three basic things. It listens for instructions from the Management Server. It executes those instructions, or project steps. It sends results from the instructions back to the Management Server in the form of output or logs.

Agents can run on a variety of server platforms in your environment. These are AIX, Linux, Windows, Solaris, HP-UX, Z-Series, System i, and Mac.

Section

Build Forge roles

Overview for V7.0

17

© 2008 IBM Corporation

This section covers the roles in Build Forge.

There are several roles that you can fulfill in a Build Forge environment: the tool administrator, configuration manager or build engineer, or the developer. Note that a person can assume more than one role.

The **Tool Administrator** installs and maintains Build Forge, sets up and maintains the Build Forge hardware, environment, and network, and manages Build Forge users and security.

The **Configuration Manager** or **Build Engineer** creates and executes Build Forge Projects, designs and implements the development environment, policies, and processes, and monitors and reports on project status.

The **Developer** performs multiple builds a day to validate changes, might perform multiple test deployments a day, and works with the Build Engineer to resolve build issues.
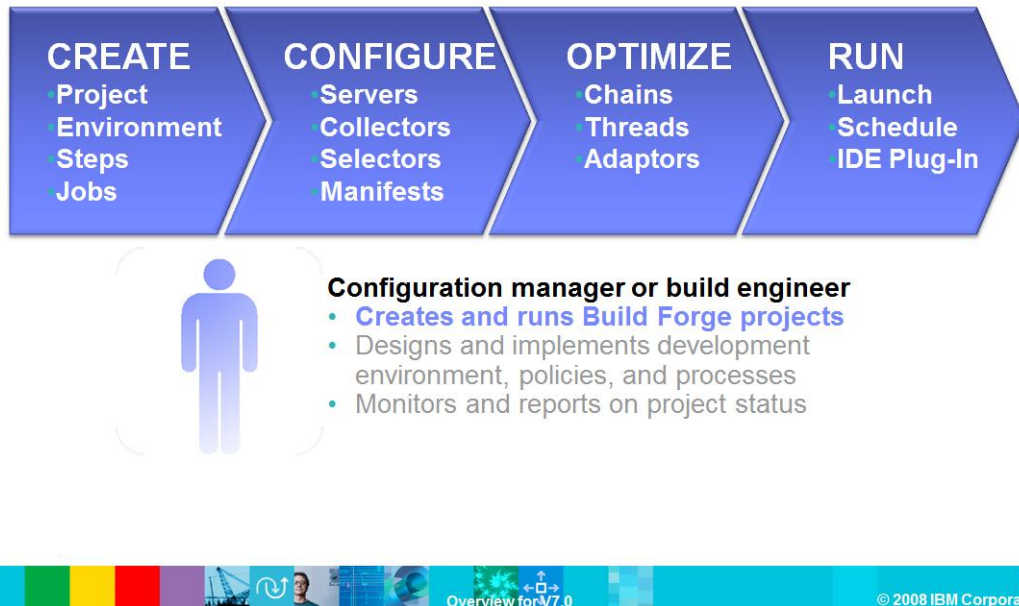
Although there are several different software development roles involved in the build and release process, the focus of Building a Project and the activities typically performed in setting up a Build Forge environment is the role of the Configuration Manager, the Build Engineer.

The four steps shown here: Create, Configure, Optimize, and Execute describe the steps involved in building a Build Forge project.

In the Create step, the build engineer creates a Project, Steps, Environment, and runs the project. In the Configure step, the engineer configures Servers, Collectors, Selectors, and Manifests. In the Optimize step, the engineer implements Chains, Threads, and Adaptors. Finally, in the Execute step, the engineer schedules and launches project runs, and launches project runs in the IDE.

Summary

- Build Forge editions – Express, Standard, Enterprise

- Key capabilities of the Build Forge system

- Key roles in Build Forge

- Key steps in creating a Build Forge project

References

▸ Rational Build Forge
  http://www.ibm.com/software/awdtools/buildforge/
▸ Information on attending a Build Forge course
  http://www.ibm.com/learning

Overview for V7.0

20

© 2008 IBM Corporation

In summary, you should now have a good overview of IBM Rational Build Forge, including: the various editions available to you, key capabilities, Build Forge architecture, and the key roles and steps in creating a project in Build Forge.

Note that this module was adapted heavily from IBM Rational University course **RS543 – The Essentials of IBM Rational Build Forge, V7.0.1.**

Trademarks, copyrights, and disclaimers

The following terms are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both:

Build Forge   IBM   Rational   System i

Rational is a trademark of International Business Machines Corporation and Rational Software Corporation in the United States, Other Countries, or both.

Windows, and the Windows logo are registered 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.

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

Product data has been reviewed for accuracy as of the date of initial publication. Product data is subject to change without notice. This document could include technical inaccuracies or typographical errors. IBM may make improvements or changes in the products or programs described herein at any time without notice. Any statements regarding IBM's future direction and intent are subject to change or withdrawal without notice, and represent goals and objectives only. References in this document to IBM products, programs, or services does not imply that IBM intends to make such products, programs or services available in all countries in which IBM operates or does business. Any reference to an IBM Program Product in this document is not intended to state or imply that only that program product may be used. Any functionally equivalent program, that does not infringe IBM's intellectual property rights, may be used instead.

Information is provided "AS IS" without warranty of any kind. THE INFORMATION PROVIDED IN THIS DOCUMENT IS DISTRIBUTED "AS IS" WITHOUT ANY WARRANTY, EITHER EXPRESS OR IMPLIED. IBM EXPRESSLY DISCLAIMS ANY WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NONINFRINGEMENT. IBM shall have no responsibility to update this information. IBM products are warranted, if at all, according to the terms and conditions of the agreements (for example, IBM Customer Agreement, Statement of Limited Warranty, International Program License Agreement, etc.) under which they are provided. Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products in connection with this publication and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products.

IBM makes no representations or warranties, express or implied, regarding non-IBM products and services.

The provision of the information contained herein is not intended to, and does not, grant any right or license under any IBM patents or copyrights. Inquiries regarding patent or copyright licenses should be made, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

Performance is based on measurements and projections using standard IBM benchmarks in a controlled environment. All customer examples described are presented as illustrations of how those customers have used IBM products and the results they may have achieved. The actual throughput or performance that any user will experience will vary depending upon considerations such as the amount of multiprogramming in the user's job stream, the I/O configuration, the storage configuration, and the workload processed. Therefore, no assurance can be given that an individual user will achieve throughput or performance improvements equivalent to the ratios stated here.

© Copyright International Business Machines Corporation 2008. All rights reserved.

Note to U.S. Government Users - Documentation related to restricted rights-Use, duplication or disclosure is subject to restrictions set forth in GSA ADP Schedule Contract and IBM Corp.

Overview for V7.0

21

© 2008 IBM Corporation