



IBM Software Group | Rational software

# Installing IBM Rational Build Forge V7.02 on UNIX and Linux

## Overview and recommendations

Rational software



@business on demand.

© 2008 IBM Corporation  
Converted to video June 29, 2015

This module will cover an overview and recommendations for installing IBM Rational® Build Forge® Version 7.0.2 on UNIX® or Linux®.

This module assumes you are familiar with IBM Rational Build Forge basics. If you feel you need a primer on Build Forge, you can exit this module and first review the Introduction to Build Forge module, and then continue with this more advanced topic.

## Module objectives

- These topics are covered in this module:
  - ▶ Overview
  - ▶ Build Forge architecture
  - ▶ Installation process walkthrough
  - ▶ Advanced configuration
- Upon completion of this module, you will be able to:
  - ▶ Identify dependencies before installation
  - ▶ Understand technology requirements
  - ▶ Understand basic terminology
  - ▶ Understand fundamental architecture
  - ▶ Use some best practices



This module will cover an overview of the installation, and is intended to set your expectations on the complexity and difficulty you can anticipate when installing on UNIX or Linux. It also will cover briefly Build Forge Architecture, walk you through an installation process, and finally provide information about advanced configurations.

Upon completion of this module, you will be able to: Identify dependencies before installation, understand technology requirements, understand basic terminology, understand fundamental architecture, and use some best practices.

## Overview

- Installing on Build Forge UNIX and Linux is much different than a monolithic product installation
- Requires open-source technologies that you obtain, compile, and install
- Several required subcomponents must be present in the technologies
- Dependencies can run as deep as OS libraries, and the dependencies differ by OS
- An improperly installed technology or subcomponent may not show an effect until run time



This overview section is intended to set your expectations on the complexity and difficulty you can anticipate when installing Build Forge on UNIX and Linux.

Installing Build Forge on UNIX and Linux is much different from a monolithic product install. It is important to note that the installation requires open-source technologies that you obtain, compile, and install. Within these technologies, several subcomponents are required and will be discussed in this section. There are also operating system dependencies associated with these subcomponents; they can vary by operating system and operating system version.

Be mindful of the dependencies and have them taken care of as you install the technologies. An improperly installed technology, subcomponent, or a missing dependency may not show its effect until you run Build Forge.

## Installing on UNIX and Linux is not simple

- Open source technologies are required that are not included with the Build Forge installation
- Each technology must be located, downloaded, compiled, configured, and installed
- The Build Forge Installation Guide gives you instructions
- Attempting an installation without reading the documentation is not recommended

The most important message of this presentation is that installing Build Forge on UNIX and Linux is not simple. It requires steps that are not typical to installing other products, notably locating, downloading, compiling, and installing open-source components. Read the Build Forge Installation Guide before starting an installation. Attempting an installation without reading the documentation is not recommended.

## Technology requirements

- Apache – HTTP server used to deliver PHP pages for the Build Forge UI
- Perl – used for step interpretation and internals
- PHP – used for DB access, UI presentation, and internals. PHP DB drivers are required for your database
- Tomcat – used to produce reports. JDBC™ DB drivers are required for your database.
- Database – data store for internal data and your project data
- C compiler – used to compile the technologies
- JVM™ – required by Tomcat



Here are the technology requirements. There are several open-source or vendor-provided technologies that must be installed as part of installing Build Forge: Apache, Perl, PHP, Tomcat, a database, C compiler, and a Java Virtual machine (JVM).

Apache is the HTTP server used to deliver PHP pages for the Build Forge user interface. Perl is used for step interpretation and internals.

PHP is used for database access, UI presentation, and internals. PHP DB drivers are required for your database. Tomcat is used to produce reports. JDBC DB drivers are required for your database. The database is a data store for internal data and your project data.

The C compiler is used to compile the technologies, and finally, the JVM is the Java virtual machine required by Tomcat.

## Technology requirements

- Already have one or more of these components installed?
  - ▶ If you already have one or more technologies installed, you probably can use it
  - ▶ However, be careful during configuration, especially if the technology is not located in a Build Forge default location



You might already have one or more of these technologies installed, and if you do, you can probably use it. You must, however, be careful during configuration -- especially if the technology is not located in a Build Forge default location. This is where the Build Forge installation guide can help guide you through the configuration.

## Technology subcomponents and dependencies

- PHP and Perl must be configured to use several subcomponents (for example, XML parsing, LDAP capability)
- Subcomponents have OS dependencies, typically on particular libraries
- The most reliable source of dependencies is the technology documentation
- TIP: If you are installing on a machine that has never had to support these technologies, investigate the dependencies.



When you compile PHP and Perl, you specify several subcomponents that are required in order for those technologies to work with Build Forge.

Some subcomponents, in turn, have dependencies on the presence of certain operating system libraries, many of which are not present by default in a bare installation. For example, PHP and Perl must be configured to use a number of subcomponents for things like XML parsing or LDAP capability.

To learn more about these library dependencies, consult the documentation provided with the technology, since it will have the latest information on that technology.

As a tip, if you are installing on a machine that has never had to support these technologies, it is worth your while to investigate the dependencies first. Consult the documentation for the technologies. Web site addresses are listed at the end of this module.

## Keys to success

- Do your homework – investigate the dependencies for your setup before you start
- Go slowly
- Read the documentation and follow the directions carefully
- Be prepared to discover undocumented OS dependencies



The keys to success are common sense for any complicated procedure. Make sure you do your homework and investigate the dependencies for your setup before you start.

In some cases you may discover undocumented operating system dependencies. These dependencies are being researched and communicated as IBM learns about them. The dynamic nature of the open-source software community entails that you can expect changes over time as new issues are discovered and resolved and new features are added.



## Section

# ***Build Forge architecture***

Now that you have an understanding of the complexities of installing Build Forge on UNIX and Linux and the dependencies required before installation, this section will cover the fundamentals for Build Forge Architecture.

## Basic terminology

- Build Forge is a system that lets users define and run projects to accomplish work
  - ▶ *Projects* are made up of *steps*
  - ▶ *Step* is made up of an executable command
  - ▶ Running of steps is dispatched to *servers*
  - ▶ Servers are a resource for executing steps. An *agent* must be installed on all machines that are to be server resources



The following terminology should help acquaint you with what Build Forge does, and by implication, why certain technologies are part of the Build Forge system.

Build Forge is a system that lets users define and run projects to accomplish work. The work is represented as projects that are run as jobs in the system.

*Projects* are made up of *steps*. A *Step* is made up of an executable command. Step execution is then dispatched to *servers*. Servers are a resource for executing steps. An *agent* must be installed on all machines that are to be server resources.

## Basic terminology

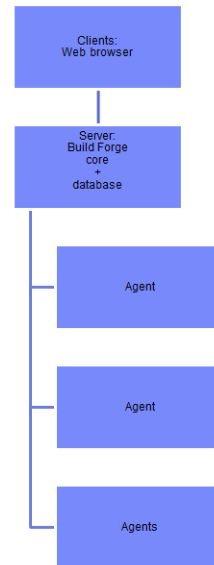
- The Build Forge core is the heart of the system.
  - ▶ Consists of two components – the Engine and the Console UI
  - ▶ Stores project definitions, users, server definitions, and other data.
  - ▶ Monitors and maintains status for running jobs
  - ▶ Dispatches steps to servers



The Build Forge core is the heart of the Build Forge system as shown in a diagram later in this module. It consists of two components – the Build Forge Engine and the Build Forge Console UI. It stores project definitions, users, server definitions, and other data. It monitors and maintains status for running jobs. And during job execution, it dispatches steps to servers to be executed.

## Fundamental architecture

- Clients: Web browsers pointed to a URL for the installed Build Forge core
- Server: Build Forge core components (plus database as backing store).
  - ▶ Stores project definitions, users, and servers (resources fulfilled by agents)
  - ▶ Runs jobs (a job is a running project).
  - ▶ Dispatches job steps to server resources
  - ▶ Maintains status of running jobs
- Agents: serve as resources for executing steps within projects



The fundamental architecture of Build Forge is relatively simple. Using a Web browser, users define projects and run them as jobs.

The Build Forge core components operate as a server which uses a database to store system information like project definitions, users, and servers. It handles requests to run jobs and maintains and communicates job status.

In the course of running jobs, it dispatches steps to agents on other machines, where the steps are performed and their completion status determined.

## Installation

- Client: No installation needed for clients. Users point their Web browsers at the Build Forge Console UI's URL
- Database: installation is a separate process. Follow vendor's instructions
- Agent: installation is relatively simple -- small executables, some configuration
- Build Forge core installation is the big project



There is no installation requirement for clients. After installation, users point their Web browsers to the Build Forge Console user interface's URL.

The database installation is independent of Build Forge installation. If you do not already have a database system installed, install it before installing Build Forge. Consult your database installation documentation for details. Subsequent steps in the Build Forge installation will require you to create database tables in that database.

Consult the Build Forge Installation Guide to install agents. You may do this after installing the Build Forge core. Note: It must be done before users can define Server resources in the Build Forge Console UI.

The agent installation is relatively simple. You extract archived files into the required location on each machine where you want work to be done. In most cases the machines are not the same machine where the Build Forge core runs. However, in test environments an agent can be installed on the same machine as the Build Forge core components.

This leaves the Build Forge core installation. It is the largest and most complex task in deploying Build Forge. The remainder of this module focuses on that task.

## Build Forge core technologies and roles

- **Apache:** provides Web services between client and Build Forge Console UI
- **Database client:** provides access to database
- **PHP:** user interface
  - ▶ PHP database drivers access stored definitions of projects, users, other system data
- **Perl:** processing engine
  - ▶ Perl database drivers
- **Tomcat:** reporting engine
  - ▶ Requires J2SE (before installing Tomcat)
  - ▶ Requires JDBC database drivers (after installing Tomcat)



The following list shows you very broadly what role each of the required technologies plays in a running Build Forge system. It should also be a hint as to where to look when you encounter errors in the system.

The Apache Web server provides Web services between client and Build Forge Console UI.

The database client software for your database is used by various components to access the database.

PHP, an open-source technology, is used to present the user interface. Within PHP, the PHP database drivers access stored definitions of projects, users, and other system data so they can be dynamically presented through the user interface.

Perl, another open-source technology, is used to write the Build Forge processing engine, which is responsible for executing projects as jobs and maintaining job status. Within Perl, the Perl database drivers provide database access. For example: when you enter and save a project, it is stored in the database. When you use the Build Forge Console UI to start a job, the processing engine reads the project from the database and then dispatches its steps to server resources.

Apache Tomcat provides the basis for the reporting engine in Build Forge. It has other requirements as well. First, it requires the Java runtime environment, J2SE. You must install it before installing Tomcat. Second, it requires its own JDBC database drivers. You install them after installing Tomcat. The JDBC drivers are distributed by the database vendor, not the Tomcat open-source project.

## Section

# *Installation process walkthrough*

Now that you have the installation basics, and architecture fundamentals, this section will cover a walkthrough of the installation process.

## Walkthrough

- Prerequisites
- Location
- Technologies
  - ▶ Apache
  - ▶ Perl
  - ▶ PHP
  - ▶ Tomcat and dependencies
- Build Forge



This walkthrough is a short tour to help you understand what is required to install each technology. It first will cover prerequisites, then where to install Build Forge, followed by technologies and Build Forge installations.



## Prerequisites

- Tools: C compiler and make
- Depending on your system's version and configuration, additional libraries may need to be added:
  - ▶ LDAP – can use Open LDAP. Requires Apache configuration for LDAP.
  - ▶ HTTPS/Secure SSL – requires additional modules in Apache setup
  - ▶ LIBXML – generally required to enable XML processing
- OS-Specific:
  - ▶ Varies, but PHP sub-modules often drive additional library updates

Make sure you have these prerequisites covered:

A C compiler and make facility are required. Depending on your system's version and configuration, additional libraries may need to be added.

If you intend to use LDAP for user management, an implementation of LDAP is required. Open LDAP subcomponent for Apache is required. You can use the open source Open LDAP. The Apache Web server you use for Build Forge must be configured to use LDAP.

If you intend to use Secure Socket Layer and HTTPS, the Apache Web server you use for Build Forge must be configured with the modules that support them.

Build Forge uses XML extensively, so XML support is required. LIBXML is an open source XML parser library.

Finally, there are OS-specific prerequisites. While these may vary, PHP sub-modules often drive additional library updates.

## Installation location

- Conventions for application locations vary across UNIX and Linux operating systems
  - ▶ /usr/bin on Linux, /opt on Solaris, and so on
  - ▶ Local IT conventions may trump these general conventions
- Build Forge default is to install to /usr/bin
  - ▶ Key issue: root-level permissions needed during installation and configuration
- Best to accept defaults for installing technologies if possible
- Pay close attention when using alternate locations. Configuration files and PATH variables are both common issues



One factor to consider before starting installation is the location of Build Forge and the required technologies.

Conventions for storing applications vary across specific UNIX and Linux platforms. For example, it is common to use /opt on Solaris operating systems. It is common to use /usr/bin on Linux operating systems. In addition, conventions in your IT environment may use a non-default location.

The Build Forge installation defaults to /usr/bin. Be sure that you have root permissions wherever you intend to install. It is easiest to accept the default installation location for the open source technologies.

If you use an alternate location, pay close attention when installing the open source technologies and Build Forge itself. You are prompted for installation locations during installs, and specifying the wrong directory in later steps will result in Build Forge not being able to run.

A common troubleshooting tool should be to inspect configuration files for the correct installation locations and PATH variables for the inclusion of all the right installation locations.

## Tools

- These tools are required to compile and install the required technologies:
  - ▶ C compiler
  - ▶ make
- Use the tools recommended for your operating system.
  - ▶ AIX®
  - ▶ Solaris
  - ▶ Linux

Because the required technologies must be downloaded as source and then built and installed, you need a C compiler and a make facility. The ones to use vary by operating system. Use the one recommended for your operating system.

## Apache

Configure	<code>./configure --prefix=/usr/local/apache-2.2.4 \ --enable-so</code>
Make (run the compile)	<code>make</code>
Install	<code>make install</code>

- Get from <http://www.apache.org>
- Note: Build Forge documentation does not tell you to do the `--enable-so` on the configure command

Apache installation typically is straightforward. Download it from the Web site, then follow the three-step process to configure, compile, and install the Web server.

If you already have Apache installed and it meets Build Forge version requirements, write down its installation location for use later. Visit [apache.org](http://apache.org) for more information.

## Perl

- Note the Perl version required for your Build Forge release. If Perl is already installed, check its version against the version required
- Get from <http://www.cpan.org>
  - ▶ Automatic install process
    - `perl -MCPAN -e shell`
    - `cpan> install [module]::[name]`
  - ▶ Manual install process

There are several add-on modules for Perl that are used by Build Forge. These are libraries that Perl uses to perform different tasks. These can be downloaded using CPAN's automatic Perl module installer or manually.

The list of required Perl modules appears in the Requirements section of the Build Forge Installation Guide.

The automatic installer is invoked with the command line listed – **perl -MCPAN -e shell**. This command invokes the CPAN shell. The CPAN shell will automatically download and install modules one at a time by running the **install [module]::[name]** command for each of the modules listed in the table.

The automatic installation process assumes network connectivity from the computer on which it is run. If that computer does not have Internet connectivity, there is a manual install process whereby the modules can be separately downloaded from [www.cpan.com](http://www.cpan.com) and then transferred to the machine on which you are setting up Perl.

The process for both the automated tool and manually adding modules is fully documented at the [www.cpan.org](http://www.cpan.org) Web site.

## PHP

Configure	<code>./configure [options]</code>
Make (run the compile)	<code>make</code>
Install	<code>make install</code>

- Get from <http://php.net/>. Commercial versions like Zend Core <http://zend.com> may include built-in OS dependency checking and support.
- Be sure prerequisites are installed
  - ▶ LibXML2
  - ▶ OpenLDAP

There are several add-on modules for PHP that are used by Build Forge.

The database driver requirements are given in the Requirements section of the Build Forge Installation Guide. Other modules are specified during a configuration step before compiling. They are documented in the “Install PHP” section in “Installing Management Console on UNIX or Linux” in the same guide.

One option that is not in the Install Guide is `--with-libxml=<dir>`. Visit [php.net](http://php.net) for more information.



## Tomcat

- Three parts:
  - ▶ J2SE 5 (prerequisite). Get from IBM or Sun and install first
  - ▶ Tomcat: Get from <http://tomcat.apache.org>
  - ▶ JDBC drivers: Get from your database vendor, sites listed in Install Guide
- Tomcat provides the Build Forge reporting mechanism. It is a Java-based Web server that operates as a JSP application container
  - ▶ Be **sure** that you have your JRE environment set up correctly
- Caveats:
  - ▶ Make sure you have set up the Java environment properly
    - PATH
    - JAVA\_HOME
  - ▶ Make sure you set up the \$CATALINA\_HOME environment variable – omitted from the documentation
  - ▶ Watch your permissions as you set up the Tomcat directories. The user that Tomcat runs as must have write permission on the reports directory you create.

Tomcat is required to run the Quick Report reports. Note that it is a separately purchased option. If you have not purchased it, none of these steps applies.

There are three parts to Tomcat installation: the Java runtime environment, the Tomcat installation itself, and JDBC drivers.

Install the Java runtime environment first. You can obtain it from IBM or Sun.

Install Tomcat second. Download it from the apache.org Web site and follow the directions to install it.

Install the JDBC drivers last. Get them from your database vendor. Web sites for downloads are listed in the Build Forge Installation Guide in the section on installing Tomcat.

Common issues in this step involve environment variables:

The PATH and JAVA\_HOME environment variables must be set correctly.

In addition, you must set up the CATALINA\_HOME variable as described in the Tomcat installation documentation. The Build Forge documentation does not currently list directions for this.

Permissions on the Tomcat directories are another possible problem. During installation you create a reports directory. That directory must be writeable by the user that the Tomcat server runs as.



## Build Forge

- **buildforge.pl script**
  - ▶ Asks several questions about the location of required technologies.
  - ▶ Performs internal checking on module and version requirements (Perl, PHP)
  - ▶ Installs Build Forge files
- Apache user must own the `templates_c` directory
- You should include Build Forge startup in the OS system initialization



You use the `buildforge.pl` script to install Build Forge files. You respond to a series of prompts for locations of required technologies. During its run, the script checks for Perl and PHP module compliance and reports errors.

A directory, `templates_c`, must be set to be owned by the username that Apache runs as. The Installation Guide has a full description.

Build Forge is typically a continuously running application, so you need to include the appropriate startup instructions in the OS system initialization, as documented in the Installation Guide.

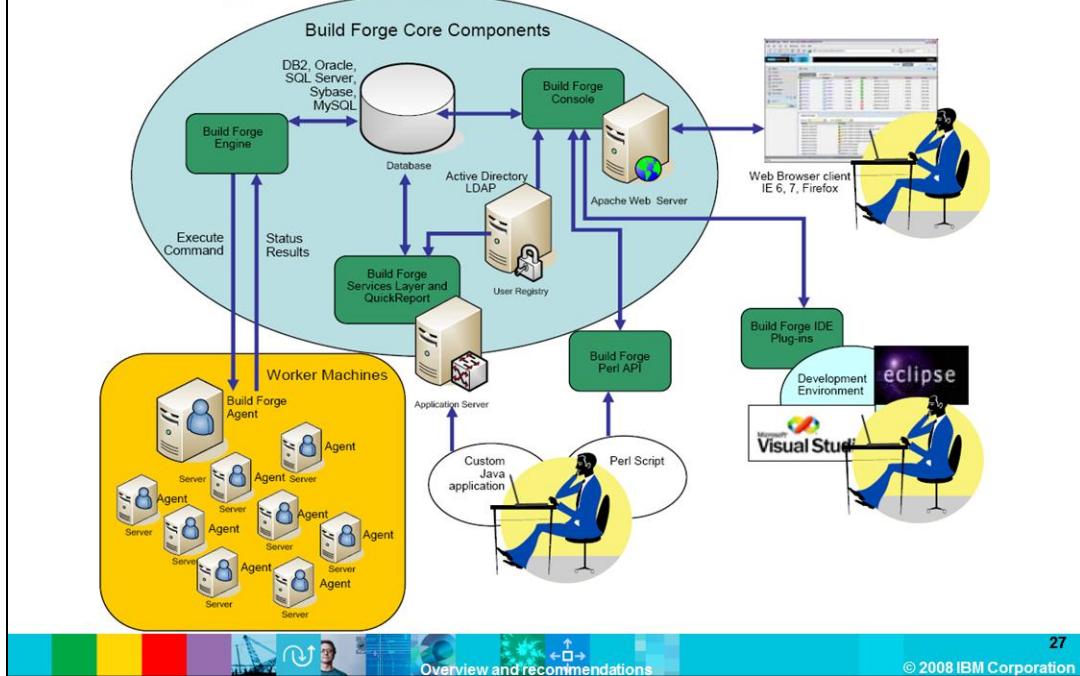
## Section

# *Advanced configuration*



This section covers advanced configuration

## Build Forge extended architecture overview



Build Forge is architected as a typical 3-tiered system.

In the middle is the Build Forge server, which contains the Build Forge core Components. It is the heart of the system. It oversees all of the activities on the managed worker systems, organizes the projects, manages user access and permissions, logging, audit trails, and so on. Note: Express edition supports the Build Forge core running on Windows server platforms only.

The system runs on various databases. It comes with DB2® Express, but also supports Oracle, MySQL, Microsoft SQL Server, and Sybase.

The primary user interface to the Build Forge core is a Web browser-based Console UI. It supports Internet Explorer (IE) and Mozilla-based browsers such as Firefox.

Build Forge IDE plug-ins are supported for Eclipse-based and Visual Studio .NET environments. They provide developers seamless integration and centralized access to build projects, status, and log file results directly within the IDE. The IDE plug-in also provides pre-flight builds. This is the ability to selectively choose local file changes from within the local workspace and preview the build results in a remotely staged, pre-configured environment, before committing the file changes to source code control.

Agents are deployed across worker machines. The agent is a very small process that does three basic things:

- 1 - Listens for instructions from the Management Server
- 2 - Executes those instructions
- 3 - Sends results from the instructions back to the server database

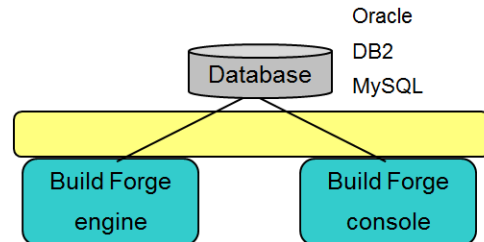
There are a wide number of agent platforms that are supported. UNIX flavors include AIX, Solaris, and HP-UX. Windows flavors include Windows 2003 Server, Windows XP Professional, and Windows Vista. Linux flavors include RedHat. Mac OS-X is included and so are IBM proprietary platforms for System z and System i (mainframe and AS/400).

The optional adaptor toolkit allows customers to extend Build Forge for more robust integrations with other ALM tools. Sample templates are provided for Rational ClearCase and Rational ClearQuest out of the box. Other source code control adaptor templates provided out of the box include CVS, Perforce, StarTeam, Subversion, and Visual SourceSafe. New templates can be created with the adaptor toolkit to support

integrations with test management tools as well.

## Build Forge core components revisited

- Components
  - ▶ Database
  - ▶ Build Forge core
    - Build Forge engine
    - Build Forge console
- Components can be in one machine or distributed. Production environments typically separate the database and Build Forge core components
- Watch for firewall issues and network permissions



Let's revisit the Build Forge architecture. Earlier, the key parts were identified as the database and the Build Forge core. The Build Forge core is further made up of subcomponents: the Build Forge Engine and the Build Forge Console UI.

Production environments typically place the database on one machine and the Build Forge core on another. The Engine may also be separated from the Console UI. This is an advanced configuration that requires help from your services representative.

## Clustering

- The Build Forge core can be configured in a redundant fashion
- Multiple installations, special configuration
- All Build Forge cores share the same database
- Scales request-handling and simultaneous-job capacity
- Described in the installation guide



Under some load conditions it can be advantageous to create a redundant configuration of the Build Forge core. A “Core Redundant” configuration increases request throughput and simultaneous job handling capacity.

See the Installation Guide for additional discussion. Depending on your needs, you may benefit from consulting from IBM Services to address loading and capacity.

## Summary

- Overview
  - ▶ Dependencies
  - ▶ Installing on Build Forge UNIX and Linux is much different than a monolithic product installation
- Technology requirements
  - ▶ Apache, Perl, PHP, Tomcat, a database, C compiler, JVM
- Fundamental architecture
- Clustering and redundant cores



In summary, this module covered an overview to Installing Build Forge on Linux and UNIX platforms. As you have learned, installing on these operating systems is much different from your regular install. You have many things to consider and plan before the install, including, but not limited to: the technology requirements, prerequisites, databases, and locations. You now also should have a good understanding of the fundamental architecture, what core components do in Build Forge, and a familiarity of clustering and redundant cores.

## Reference: Web site list

- IBM Rational Build Forge
  - ▶ <http://www.ibm.com/software/rational>
- Perl
  - ▶ <http://www.cpan.org>
- Java 2 Standard Edition (J2SE)
  - ▶ <http://java.sun.com>
- Apache HTTP Server
  - ▶ <http://httpd.apache.org>
- PHP
  - ▶ <http://www.php.net>
- Tomcat
  - ▶ <http://tomcat.apache.org>

For more information on IBM Rational Build Forge, or any of the topics discussed in this module, visit the links shown here.



## 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:

AIX          Build Forge          DB2          IBM          Rational

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

J2SE, Java, JDBC, JRE, JSP, JVM, and all Java-based trademarks are trademarks of Sun Microsystems, Inc. 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.