

IBM Rational Build Forge Installation Guide Version 7.0.2.5

Table of Contents

Build Forge Installation Guide	1
Build Forge Terminology and Components	2
Terminology	2
Components	2
Build Forge Product Editions	3
Requirements	4
Management Console Requirements	4
Database Requirements	5
Agent Requirements	8
Web Client Requirements	8
Networking Requirements for IPv6 Support	9
Integration Requirements	11
National Language Support in Build Forge	11
Planning the Installation	15
Standalone Deployment	15
Distributed Deployment	15
Clustered Management Console Deployment	15
Integration with IDEs	15
Build Forge Installation Road Maps	16
Road Map: Management Console on Windows with DB2 Express	16
Road Map: Management Console on Windows with a Supported Database	17
Road Map: Management Console on UNIX/Linux	19
Road Map: Agent on Windows	24
Road Map: Agent on UNIX/Linux	25
Installing the License Server	26
Installing a License Server	26
Installing Licenses	26
Changing the License Server to Use	27
Installing a Standalone Deployment on Windows Using Launch Pad	28
Accessing Launch Pad	28
Running Launch Pad	29
Installing Management Console on Windows	30
Pre-Installation Setup	30
Build Forge Installation (Windows)	40
Post-Installation Setup	42
Installing Management Console on UNIX or Linux	45
Prerequisites	45
Database Installation and Configuration	45
Apache Installation and Configuration	51
PHP Installation and Configuration	54
Perl Installation and Configuration	58
Build Forge Installation and Configuration (UNIX/Linux)	59
Apache Tomcat Installation and Configuration	64
Installing Agents	68
Installing the Agent on Windows Platforms	68
Installing the Agent on UNIX and Linux Platforms	70

Installing the Agent on System i Platforms	72
Running an Agent	73
Configuring the Agent	75
Troubleshooting Agents	86
Alternate Configurations	92
Configuring Management Console to Use an Alternate Port	92
Configuring a Console Cluster	93
Installing the System on VMware	93
Configuring Support for SSL and HTTPS	94
Silent Install Option for the Management Console on Windows	103
Setting Up Users	106
Root User	106
Creating and Editing Users	106
Creating a Read-only User for Reporting	109
Verifying the Installation	110
Creating a Hello World Project	110
Configuring Servers	111
Running Projects	112
Integrating with Other Products	114
About Plug-ins for IDEs	114
Plug-ins for Eclipse and Rational Application Developer	114
Plug-in for Microsoft Visual Studio	116
Special Variables for Test Projects	118
Upgrading an Existing Version	120
Upgrading the Management Console on Windows	120
Upgrading the Management Console on UNIX or Linux	125
Upgrading Agents	130
Migration Script Reference	130
Upgrading APIs	132
Troubleshooting Common Problems	132
Administration	134
Managing the Engine	134
Managing the Database	135
Support	137
Executables List	137
Contacting IBM Customer Support for Rational Products	138
Notices for IBM Rational Build Forge documentation	139

List of Figures

1. User Details Form	108
2. Creating HelloWorld	110
3. Start Project Page	113

Build Forge Installation Guide

This guide provides installation instructions for the Management Console and the agent.

Note

Before using this information and the product it represents, read the information in [“Notices for IBM Rational Build Forge documentation ” on page 139](#).

First Edition June 2009.

This edition applies to version 7.0.2 of IBM[®] Rational[®] Build Forge[®] and all subsequent releases and modifications until otherwise indicated in new editions.

Document version and build: 7.0.2-005.0012.

© **Copyright International Business Machines Corporation 2003, 2009. All rights reserved.**

US Government Users Restricted Rights - Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Build Forge Terminology and Components

This section introduces you to Build Forge[®] components and terminology.

Terminology

The primary purpose of the system is to enable you to define and run *projects*. A project accomplishes a task, like building and packaging software. The power of the system comes from its ability to track, control, and report on the execution of complex tasks distributed over multiple hosts.

Project: a set of *steps* to be executed.

Step: a single executable operation that can be individually scrutinized for success or failure. When a project is run, each of its steps is run in the specified order on the specified *servers*. It can be run in an *environment* you specify. A step is made up of one or more commands. Each command may itself launch an executable file, a batch file, or script that launches many other commands. A command is anything that can be invoked from the command line of a given server.

Environment: defined in an Environment which consists of one or more environment variables. Environments can be assigned to *servers*, *projects*, or *steps*.

Server: is a host where an agent is installed. The agent enables the Management Console to communicate with the host and run a project or project step.

Components

The Build Forge[®] system is made up of the following components arranged in a three-tier architecture:

- **Web Client:** Web browsers are used by end users and administrators to access the Management Console.
- **Management Console:** the primary user and administrator interface. Users can define and run projects. Administrators can manage server resource, user authorizations, and project characteristics.
- **Database:** provides backing store for the Management Console. The database stores project definitions, system configurations, and user configurations.
- **Agent:** an installed interface on a host that lets the Management Console use it as a server resource. An agent must be installed on every host that you want to be used as a server resource by the Management Console.

The components can be deployed in a variety of ways, ranging from all components on a single host to a system that uses clustered Management Consoles and a large number of distributed server resources.

Build Forge Product Editions

The editions of the Build Forge product are

- IBM Rational Build Forge Express Edition
- IBM Rational Build Forge Standard Edition
- IBM Rational Build Forge Enterprise Edition

The table lists the distinct components or features for the editions.

Note

Editions do not support the same operating systems and databases for the Management Console.

Component or Function	Express Edition	Standard Edition	Enterprise Edition
Management Console	Windows only	Windows, UNIX/Linux	Windows, UNIX/Linux
Database	supported Windows databases (including DB2 Express)	supported Windows databases (including DB2 Express) supported UNIX/Linux databases	supported Windows databases (including DB2 Express) supported UNIX/Linux databases
Agents	all supported operating systems	all supported operating systems	all supported operating systems
License Server	not required (5 concurrent users)	required (25 concurrent users)	required (150 concurrent users)
Adaptor Toolkit	supported	supported	supported
Quick Report	not supported	supported	supported
APIs (Perl, Java)	not supported	not supported	supported
Dynamic server management	not supported	not supported	supported

Requirements

This section lists hardware and software requirements for installing the Build Forge[®] system.

It identifies requirements for system components:

- Management Console
- Database
- Agent
- Web client
- Integration

Management Console Requirements

This section identifies licensing requirements and operating system requirements. It also lists additional requirements that apply if you intend to install the Management Console on UNIX or Linux systems.

Licensing Requirements

The Build Forge system uses a FLEXlm license server to govern runtime usage. You can use floating licenses or node-locked licenses. The supported license servers are:

- Rational License Server - version 7.0 or above. This is a FLEXlm server that is distributed and supported by IBM Rational. Use this license server if you do not already have a FLEXlm license server.
- FLEXlm License Server - Build Forge should work with any FLEXlm server on any platform if it is still under active support by Macrovision.

Operating System Requirements

Resource Requirements: the Management Console typically requires 2GB of disk space for its files and 1GB RAM (minimum) to operate, 2G RAM if using Quick Report.

The Management Console runs on the following **operating systems** and *hardware platforms*.

Windows Systems

- **Windows Server 2003 SP1:** *32-bit Intel*
- **Windows XP Pro:** *32-bit Intel*

UNIX and Linux Systems

- **AIX 5.3+:** *System p, Power5*

- **Red Hat Enterprise Linux 4.x:** *32-bit Intel*
- **Solaris 8, 9, 10:** *64-bit SPARC, 32-bit SPARC*

Note

Build Forge Management Console and Agent are compiled as 32-bit applications. You can run them on the 32-bit Solaris OS and the 64-bit Solaris OS.

Additional Technology Requirements

The following technologies are required as part of a Management Console installation. What you need to do depends on the platform you use to run the Management Console.

- Windows platforms: the technologies are installed automatically when you install Management Console.
- UNIX or Linux platforms: the technologies need to be installed before installing Management Console. You are given instructions for downloading them and compiling them from source.

Specific PHP and Perl database driver requirements are shown in Database Requirements, below.

- Apache 2 - Get from <http://www.apache.org/>
- PHP 5.2.1 - Get from <http://php.net/> (PHP 5.1.2 is installed with the Windows installation of Build Forge). Database driver requirements are shown in Database Requirements, below.
- Perl 5.8 - Get from <http://www.cpan.org/>. Database driver requirements are shown in Database Requirements, below.
- Tomcat 5.5.9 - Get from <http://tomcat.apache.org/>
- JDBC drivers - Database driver requirements are described in Database Requirements, below.

Database Requirements

The system installs a DB2[®] Express database by default (on the machine where you install the Management Console). The following databases and versions are supported:

- DB2 Express 9.1.1 (this is the version packaged with the Build Forge system)
- DB2 8.x and 9.x on UNIX, Linux or Windows platforms only. DB2/390 on System z is not supported.
- Microsoft[®] SQL Server 2005 (**on Windows platforms only**). *Support for SQL Server is for English-only, because it does not support UTF-8.* The database must be created using a case-insensitive collation, such as the default: `SQL_Latin1_General_CP1_CI_AS`.
- MySQL 5.0 (**except:** for Management Console on AIX platforms, MySQL is not supported)
- Oracle 9.2+, 10g, 10.2

- Sybase Adaptive Server Enterprise (ASE) 12.5 (**on Windows platforms only**)

Database Client Drivers

You must have database client drivers already installed on the host where you install Management Console. The installation procedures include instructions on how to install and configure them. The following table shows the technology used as a database driver (on the Management Console host) to access the database (typically installed on a different host).

Database Client Drivers

OS / DB	DB2 Express	Microsoft SQL Server	MySQL	Oracle	Sybase
Windows	DB2 Connect Client	ODBC	Native	Oracle Client (can use Instant Client)	ODBC
UNIX or Linux	DB2 Connect Client	n/a	Native	Oracle Client (can use Instant Client)	n/a

Database Connection Port

Build Forge assumes that you are using the default connection port for your database. If you are not using the default port, add the `db_tcp_port` keyword to the `buildforge.conf` file and set its value to the database connection port you are using.

PHP and Perl Database Drivers

These drivers are installed automatically during a Management Console installation on Windows. You install them manually on UNIX or Linux. See the installation procedures for more information.

PHP and Perl Database Drivers

Database	UNIX/Linux		Windows	
	PHP Driver	Perl Driver	PHP Driver	Perl Driver
DB2 Express	ibm_db2	DBD::DB2 (note 1)	included	included (note 1)
Microsoft SQL Server	not supported	not supported	included	included
MySQL	mysqli (note 3)	DBD::mysql (note 3)	mysql and mysqli	included
Oracle	oci8 (note 2)	DBD::Oracle	oci8 (note 2)	included
Sybase	not supported	not supported	included	included

- *Note 1: **DBD::DB2** is installed automatically if you install the DB2 Express for Windows included with Build Forge. You may have to add it if you are setting up an existing DB2 Express installation or installing it on a UNIX/Linux platform.*
- *Note 2: **oci8** may already be present if an Oracle client has been installed. **Version 5.2.1 of the driver is installed as part of the PHP 5.1.2 installation on Windows.** To check for its presence,*

place a `test.php` web page containing `<? phpinfo(); ?>` on the intended Management Console host below `<bf-install>/webroot/public`. Check that Apache or Build Forge is running. View `test.php` using a browser. Look for the `oci8` module listing.

- *Note 3:* except on AIX. Management Console on AIX does not support MySQL.

JDBC Drivers

JDBC drivers are required for use with Tomcat, which drives the reporting system. JDBC drivers for DB2 Express on Windows are installed automatically. For all other supported databases on Windows, UNIX, or Linux, you must download and install the drivers. Instructions are provided for each database on each platform.

Perl Modules

These modules are installed automatically during a Management Console installation on Windows. You install them manually on UNIX or Linux. Version number requirements are highlighted where relevant. Use the Perl CPAN utility to check for modules and install modules. Use the CPAN web site to get more information, <http://www.cpan.org/>.

- To check for a module: `perl -M<Module>::<Name> -e 1`
- To open a CPAN shell: `perl -MCPAN -e shell`
- To install a module from a CPAN shell: `cpan> install <Module>::<Name>`

Perl Modules Required for Management Console

Perl Modules		
Config::IniFiles	IO::File	Socket6 (for IPV6)
Data::Dumper	IO::Select	Sys::Hostname
Data::UUID (UNIX only)	IO::Socket	Text::Balanced
DBI 1.49+	IO::Socket::INET6 (for IPV6)	Time::Piece
File::Basename	IPC::Open2	Time::Local
File::Path	LWP::UserAgent	URI
File::Temp	LWP::Simple	URI::Escape
Getopt::Std	MIME::Lite	XML::Writer
HTML::Template 2.8+	POSIX	XML::Writer::String
HTTP::Request	Set::Crontab	XML::XPath 1.13+

Agent Requirements

The agent runs on the following **operating systems** and *hardware platforms*, requiring the following (RESOURCES).

Windows Systems

- **Windows Server 2003:** 32-bit Intel (160KB DISK, 1MB RAM¹)
- **Windows XP Pro:** 32-bit Intel (160KB DISK, 1MB RAM¹)
- **Windows Vista Ultra, Windows Vista Enterprise:** 32-bit Intel (160KB DISK, 1MB RAM¹)

UNIX and Linux Systems

- **AIX 5.3+:** System p, Power5 (1MB DISK, 1MB RAM¹)
- **HP-UX 11i (v1, 2, 3):** HP 700, 800 (1.1MB DISK, 1MB RAM¹)
- **Macintosh OS 10.x, OS X Server 10.x:** 32-bit Intel, PowerPC (200KB DISK, 1MB RAM¹)
- **Red Hat Enterprise Linux 4.x:** 32-bit Intel (600KB DISK, 1MB RAM¹)
- **Solaris 8, 9, 10 :** 64-bit SPARC, 32-bit SPARC (850KB DISK, 1MB RAM¹)
- *IBM Systems*
- **IBM System i v5r4:** AS400
- **IBM System z 1.6, 1.7, 1.8:** Mainframe

Note

The agent is provided as a 32-bit application. It should not be compiled from source as a 64-bit application. The compiled 32-bit agent, for example, can run on a 64-bit Solaris OS.

Note

¹ Memory requirements are for a running agent only. Add to that requirement any requirements for applications that the agent will run during processing. For example, running a step to perform a code checkout is a common usage of the agent. Performing that step requires a significant amount of additional memory on the agent host. For best results, perform live tests and obtain memory consumption metrics before putting the system into production.

Web Client Requirements

Some minimum resolution and browser version levels are required when accessing the Management Console from a client system.

Use a web browser to access Management Console as a client.

- Browsers supported:
 - **Internet Explorer:** IE 6 and IE 7. The UI experience is better in IE 7. ActiveX controls must be enabled or the console server host must be added to the Trusted Sites list. (Tools > Internet Options > Security). Note that Windows 2003 disables ActiveX controls by default.
 - **Mozilla Firefox:** 1.5 or 2.0 and later.
 - **Mozilla Suite:** 1.7.13 and later.
- Minimum display resolution:
 - 1024 x 768 minimum in order to display the Management Console correctly. Recommended resolution: 1280 x 1024 or higher.

Note

Do not shrink the browser window smaller than the size that the Build Forge application uses (1024 x 768). In some cases shrinking the window hides necessary controls and causes unpredictable behavior.

- **Clear cache:** *all web clients* must clear their cache after the Management Console is updated (upgrade installation) or has an iFix applied.

Networking Requirements for IPv6 Support

The Build Forge system can run on IPv6 and mixed IPv6/IPv4 networks with some restrictions.

The Build Forge system can run on IPv6 networks and heterogeneous IPv4/IPv6 networks. Support is mostly transparent, but there are some requirements and restrictions:

Know the address configuration of machines where you install the Management Console and agents

The main requirement for IPv6 support is that your machines and network are configured correctly to support IPv6. If there are network configuration problems, then host names and addresses specified from within the Build Forge system will not resolve correctly.

Use the correct address format when entering IP addresses within Build Forge

There are only a few places within Build Forge where administrators or users enter a host name or IP address:

- During installation, administrators specify a host name or IP address that the Management Console uses to communicate with the license server (Standard and Enterprise Editions) and the database.
- In the agent configuration (bfagent.conf file), there is an optional setting to restrict connections to the agent to a particular address or range of addresses.
- Users enter a URL in a browser in order to view the Management Console user interface. The URL consists of

the host name or IP address of the machine where the Management Console is running. For example, to access Management Console installed on a machine named **BFmachine** that has *both an IPv4 and IPv6 address configured*, a user could enter any of the following in the web browser address bar:

- `http://BFmachine/`
- `http://localhost/`
- `http://127.0.0.1/` (IPv4 loopback address)
- `http://::127.0.0.1/`,
`http://0:0:0:0:0:0:127.0.0.1/` (IPv6 abbreviations of an IPv4 loopback address), or simply `http://::1/` (IPv6 compressed notation for the loopback address)

The IPv4 and IPv6 address formats differ in format as well as length:

- IPv4 format: 32 bits, specified as four decimal-separated decimal values, for example 255.255.255.255
- IPv6 format: 128 bits, specified as eight colon-separated hexadecimal values, for example FE80:0000:0000:0000:0202:B3FF:FE1E:8329. There are a number of conventions for using the higher order fields. There are also rules for abbreviation. Build Forge does not perform any interpretation of IP addresses, they are passed directly to the network. Therefore any legal and valid abbreviation should work. Please see other references for more information on IPv6 address conventions.

Put components that do not support IPv6 on a host machine with an IPv4 address

The following components do not support IPv6. They must be installed on a machine that has an IPv4 address.

- Rational License Server, the license server for Build Forge Standard and Enterprise Editions
- DB2 database: the PHP database drivers do not yet support IPv6.
- MySQL database: the PHP database drivers do not yet support IPv6.
- Oracle database *other than 11g*: the PHP database drivers do not yet support IPv6.
- *Windows*: due to a limitation in the way Apache binaries were compiled and released by The Apache Foundation,

the standard installation of Management Console does not support IPv6 on Windows platforms.

To support IPv6, the Apache source must be compiled using the flag `-D APR_HAVE_IPV6`. At present the Windows binaries of Apache (2.2.8) have not been compiled with this flag.

If you need IPv6 support for the Management Console on Windows platforms, contact IBM Customer Support for Rational for assistance in replacing the Apache binaries.

Integration Requirements

This section identifies product and version requirements for SCM (source configuration management) adaptors and IDE (integrated development environment) plug-ins.

SCM Adaptor Requirements

Build Forge can be integrated via adaptors to source configuration management systems. Adaptors are provided for the following systems at the designated version(s):

- ClearQuest: version 7.0 or later
- ClearCase: version 6.0 or later

IDE Plug-in Requirements

Plug-ins are provided for IDEs that allow IDE users to operate the Build Forge console from the IDE user interface. Plug-ins are provided for the following systems at the designated version(s):

- Eclipse: version 3.02 or later
- Rational Application Developer: version 6.0 or later
- Microsoft Visual Studio: 2003 and 2005 (.NET IDE plug-in)

Tivoli IT Asset Manager Requirements

The product is shipped ready for use with Tivoli IT Asset Manager products. The following file is installed in the installation root directory and must not be deleted, renamed, or edited:

`IRBFSRV0700.SYS2`

National Language Support in Build Forge

The Build Forge product provides localized support for French, German, Italian, Brazilian Portuguese, Spanish, Japanese, Korean, Simplified Chinese, and Traditional Chinese.

This section provides information about language support in Build Forge.

Language Settings for the Management Console and Agent

This topic describes how Build Forge configures language settings for the Management Console, agent, and engine components.

Management Console Language Settings

The language setting for the current user determines the language used to display interface controls in the Management Console.

The Management Console language setting for a user account is set as follows:

Root user	The installation program creates a root user account that the administrator uses to log in to the console the first time. On initial log in, the language setting for the root user is based on the operating system language for the Build Forge engine host.
Language default for other users	For administrator-created user accounts, the default console language is initially set using the language preference of the Web browser that the user logs in on. (This is a configuration setting of the Web browser, not Build Forge.) If the administrator does not change the language preference for the user (Administration >Users >Language), it is inherited from the Web browser. LDAP-created user accounts always use the language preference configured for the Web browser.
Changing the default user language	After logging in to the console, the administrator may select a different language setting for individual Build Forge users (Administration > Users > Language). If language preferences for users are configured in this way, the Management Console displays interface controls in the language selected for the user regardless of the language configured for the Web browser.

Agent and Build Forge Engine Language Settings

The operating system language for the Build Forge engine host dictates:

- the language used by the Build Forge engine
- the default language for the Build Forge agent (The agent language setting controls the language used to produce system messages and job output.)

Regardless of how the language is set for the Management Console (either through the Web browser or as a Build Forge user preference), the agent logs data for system messages and job output in the operating system language of the Build Forge engine host.

To avoid mixed languages in the Management Console interface, the language selected for the Management Console must match the language used by the Build Forge engine host.

Recommended Language Setup

Build Forge recommends using the same language for the Web browser, the operating system on the Management Console host, and the Build Forge user to ensure that the language displayed in the Management Console matches the language that the agent uses to log data for system messages and job output.

International Data Support for the Database Host

To display and manipulate international data, the host machine for the Management Console database:

- must be configured to use the Unicode UTF-8 character set
- must have the necessary fonts installed

Changing User Language Preference in the Management Console

The default language for all Management Console users is initially set to the language of the Web browser.

To change the language setting for a Build Forge user, select Administration > Users > Language.

The Management Console displays user interface controls in the language selected for the user, but the agent continues to log data for system messages and job output in the operating system language of the Build Forge engine host.

Determining the Language/Charset for UNIX/Linux Hosts

If the Management Console or agent is installed on a UNIX/Linux host, use the `locale` command as follows.

- To determine the language/charset currently being used by the operating system:

```
$locale
```

- To determine the language/charset combinations available to the operating system:

```
$locale -a
```

- To set locale on login, a rc or profile script is used.

Determining the Language Code Page for Windows Hosts

If the Management Console or agent is installed on a Windows host, use the `chcp` command as follows:

- To determine the active code page number:

```
> chcp
```

- To set the code page, enter the number for the language:

```
$chcp code_page
```

The following table lists the Windows character encodings for the NLV1 languages that Build Forge supports:

Language	Code Page
English	1252
French	1252
Spanish	
Italian	
German	
Portuguese	
Japanese Shift-JIS	932
Korean	949
Simplified Chinese GBK	936
Traditional Chinese Big5	950

Planning the Installation

This section describes the planning needed for installing the Build Forge[®] system. It describes the following deployment configurations:

- Standalone
- Distributed
- Clustered Management Console
- Integration with IDEs

The system is highly configurable. Within a deployed system, project definitions can be configured to take advantage of resources to meet different needs.

Standalone Deployment

A standalone deployment has the database, Management Console, and one agent deployed on a single host. This deployment is typically used for evaluation or development purposes.

Distributed Deployment

A distributed deployment distributes the database, Management Console, and multiple agents across multiple hosts. This deployment is the most typically used production environment.

Clustered Management Console Deployment

If request and job traffic will exceed the capabilities of a single host, the Management Console can be clustered on multiple hosts to increase capacity. One installation of the Management Console is designated the master. Others are designated secondary.

Integration with IDEs

The system can be integrated with Integrated Development Environments. Typically this means that users can view projects and run jobs from inside the IDE.

Build Forge Installation Road Maps

Use the road maps in this section to survey the list of tasks required tasks to install and configure Build Forge components in your environment.

Road Map: Management Console on Windows with DB2 Express

Use this road map to install the Management Console and the provided DB2 Express database on a single Windows host. Choosing the provided DB2 Express database is the quickest setup option.

Road Map: Management Console on Windows with DB2 Express on a Single Host

	Task	Topic
	A license server must be running and accessible from the Management Console host.	Installing a License Server
	Install the license files in a temporary directory on the license server host	Installing Licenses
	Install licenses for separately licensed components, like the Adaptor Toolkit and Quick Report.	Installing Licenses
	Locate mc-<version>-Windows.exe on the download site or the product CD. Click the file and save it to a temporary directory.	Build Forge Installation
	Double-click mc-<version>-Windows.exe and start the installation program.	Build Forge Installation
	Follow the instructions in the installation program to install the Management Console.	Build Forge Installation
	Launch the Management Console: <ul style="list-style-type: none"> • http://<hostname>[:<portnumber>] (from any Web browser) • http://localhost (from the local Web browser) 	Build Forge Installation
	Wait for the database schema to load. It might take a minute or so for the Management Console to load the database schema the first time. Do not stop the Management Console engine before the load completes; doing so may require removing and reinstalling the database.	Build Forge Installation
	Log on using root/root . Change the root password immediately to restrict console access.	Build Forge Installation

Road Map: Management Console on Windows with a Supported Database

Use this road map to install the Management Console on a Windows host and install a supported database (other than DB2 Express) on a separate host.

Road Map: Management Console on Windows with a Supported Database on a Separate Host

	Task	Topic
	A license server must be running and accessible from the Management Console host.	Installing a License Server
	Install the license files in a temporary directory on the license server host	Installing Licenses
	Install licenses for separately licensed components, like the Adaptor Toolkit and Quick Report.	Installing Licenses
	Choose a supported database to use as the database for the Management Console	Database Requirements
	Verify that a network connection exists for database host and the Management Console host	Pre-Installation Setup
	<p>Set up the database for the Management Console.</p> <p>Create a database, database user, grant owner access to the database user.</p> <ul style="list-style-type: none"> • Configure the database client driver (DLL) on the Management Console. • If you intend to use international data, the database must be configured to use a character set that complies with UTF-8. 	Pre-Installation Setup DB2 Setup Microsoft SQL Server Setup MySQL Setup Oracle Setup Sybase Setup
	Locate mc-<version>-Windows.exe on the download site or the product CD. Click the file and save it to a temporary directory.	Build Forge Installation
	Double-click mc-<version>-Windows.exe and start the installation program.	Build Forge Installation
	<p>Follow the instructions in the installation program to install the Management Console.</p> <p>At Database Settings, select Connect to existing database.</p>	Build Forge Installation
	Download and unzip the JDBC driver for your database. Copy the jar file only to \$CATALINA_HOME/common/lib. Add \$CATALINA_HOME to your system PATH.	Post-Installation Setup
	Verify that the client driver (DLL) for your database is listed in your system PATH.	Post-Installation Setup

	Task	Topic
	(Proxy server only) To enable Management Console access, add the proxy server name and path to the <bf-install>/Apache/php.ini file.	Post-Installation Setup
	(Oracle only) Set ora_ph_type=96 so the driver does not strip trailing blanks.	Post-Installation Setup
	(DB2 only) Set DB2CODEPAGE=1208 to correctly display international characters.	Post-Installation Setup
	Launch the Management Console: <ul style="list-style-type: none"> • Enter http://<hostname>[:<port_number>] (from any Web browser) • Enter http://localhost (from the local Web browser) 	Build Forge Installation
	Wait for the database schema to load. It might take a minute or so for the Management Console to load the database schema the first time. Do not stop the Management Console engine before the load completes; doing so may require removing and reinstalling the database.	Build Forge Installation
	Log on using root/root . Change the root password immediately to restrict console access.	Build Forge Installation

Road Map: Management Console on UNIX/Linux

Use this road map to install the Management Console on a supported UNIX/Linux operating system with a supported database on a separate host.

Note

This road map works for all product editions except the Express Edition, which is Windows only.

The road map tasks are described in the following tables:

- Management Console Pre-Installation Tasks for UNIX/Linux
- Management Console Installation Tasks for UNIX/Linux
- Management Console Post-Installation Tasks for UNIX/Linux

- Management Console Start Up Tasks for UNIX/Linux

Install Prerequisite Products

Several prerequisite products must be installed on the Management Console host. For each task, see the related topic in the installation guide for detailed instructions.

Management Console Pre-Installation Tasks for UNIX/Linux

Task	Topic
Verify that the target UNIX/Linux operating system is supported for the Management Console.	Operating System Requirements
A license server must be running and accessible from the Management Console host.	Installing a License Server
Install the license files in a temporary directory on the license server host	Installing Licenses
Install licenses for separately licensed components, like the Adaptor Toolkit and Quick Report.	Installing Licenses
Choose a supported database to use as the database for the Management Console	Database Requirements
Verify that a network connection exists for the database host and the Management Console host	Database Installation and Configuration
<p>Set up the database for the Management Console:</p> <p>Create a database, database user, and grant owner access.</p> <ul style="list-style-type: none"> (DB2 only) Set up a database user name and password in your UNIX/Linux operating system. If you intend to use international data, configure the database to use a UTF-8 character set. 	Database Installation and Configuration International Data Support DB2 Setup MySQL Setup Oracle Setup
Download, install, and configure Apache web server	Apache Installation and Configuration
Download, install, and configure PHP. (Proxy server only) To enable Management Console access, add the proxy server name and path to the <php-install>/lib/php.ini file.	PHP Installation and Configuration
Download, install, and configure Perl.	Perl Installation and Configuration

Installation Tasks

Complete the following tasks as a non-root user. Instructions may vary by platform. For each task, see the related topic in the installation guide for detailed instructions.

Management Console Installation Tasks for UNIX/Linux

	Task	Topic
	Create an installation directory, for example, <code>usr/local/buildforge</code> .	Build Forge Installation and Configuration
	Locate the <code>mc-<version>-unix.tar.gz</code> file for your UNIX/Linux operating system on the download site or the product CD.	Build Forge Installation and Configuration
	Click the <code>mc-<version>-unix.tar.gz</code> file and save it to the installation directory.	Build Forge Installation and Configuration
	Unzip and extract the file contents to the installation directory: <pre>\$gunzip -c -mc-<version>-unix.tar.gz tar xvf -</pre>	Build Forge Installation and Configuration
	Change to the installation directory and run the <code>bfinstall.pl</code> script: <pre>\$/bfinstall.pl - - with-dbtype=<database_type></pre>	Install Build Forge

Post-Install Tasks

Post-install tasks are summarized here. Some optional tasks are not listed. For each task, see the related topic in the installation guide for detailed instruction.

Management Console Post-Installation Tasks for UNIX/Linux

	Task	Topic
	Configure the Management Console to use license server: In <bf-install>/Platform, create a bfkey.txt file. Add the license server host name to the bfkey.txt file.	Installing the License Server
	Configure the Management Console to start at system startup: <ul style="list-style-type: none"> • Copy <bf-install>/rc/buildforge script to init.d. • Create run-level links from the run-level directory to the buildforge script in init.d 	Include Build Forge in System Startup
	Set the owner of <bf-install>/ui/templates_c to the Apache user.	Set Permissions for Apache
	Download and install J2SE 5.	Apache Tomcat Installation and Configuration
	Download and install Apache Tomcat.	Apache Tomcat Installation and Configuration
	Download and unzip the JDBC driver for your database. Copy the jar file only to \$CATALINA_HOME/common/lib. Add \$CATALINA_HOME to your system PATH.	Apache Tomcat Installation and Configuration
	Configure and start Tomcat.	Configure Tomcat for Build Forge

Start Up Tasks

Start up tasks are summarized here.

Management Console Start Up Tasks

	Task	Topic
	Launch the Management Console: <ul style="list-style-type: none"> • Enter http://<hostname>[:<port_number>] (from any Web browser) • Enter http://localhost (from the local Web browser) 	Build Forge Installation
	Wait for the database schema to load. It might take a minute or so for the Management Console to load the database schema the first time. Do not stop the Management Console engine before the load completes; doing so may require removing and reinstalling the database.	Build Forge Installation
	Log on using root/root . Change the root password immediately to restrict console access.	Build Forge Installation

Road Map: Agent on Windows

Complete the tasks in the table to install the Build Forge Agent on a Windows host.

Agent Installation Tasks for Windows

Task	Topic
Locate the win-bfagent-<version>.exe on the CD or the download site. Click the file and save it to a temporary directory.	Installing the Agent on Windows Platforms
Double-click win-bfagent-<version>.exe to start the installation program.	Installing the Agent on Windows Platforms
Read the license agreement, if you agree, click I Agree .	Installing the Agent on Windows Platforms
Choose an installation directory. The default location, C:\Program Files\Build Forge\Agent, is recommended.	Installing the Agent on Windows Platforms
Select Install as a service option to automatically start the agent when Windows starts up and stop the agent when Windows shuts down.	Installing the Agent on Windows Platforms
Click Install .	Installing the Agent on Windows Platforms

Road Map: Agent on UNIX/Linux

Complete the tasks in the table to install the Build Forge Agent on a UNIX/Linux host. The table summarizes general tasks. Task details vary by platform. For detailed instructions, see the installation guide.

Agent Installation Tasks for UNIX/Linux

Task	Topic
Locate the <platform>-bfagent-<version>.tar.gz file on the CD or the download site. Click the file and save it to the agent installation directory. For example, /usr/local/bin.	Installing the Agent on UNIX and Linux Platforms
Unzip the file: \$gzip -d <platform>-bfagent-<version>.tar.gz	Installing the Agent on UNIX and Linux Platforms
Extract the file contents: \$tar xvf <platform>-bfagent-<version>.tar	Installing the Agent on UNIX and Linux Platforms
Change to the installation directory and start the agent installer: \$./install.sh	Installing the Agent on UNIX and Linux Platforms

Installing the License Server

You must have a license server running and licenses installed in order to run Build Forge[®] components. It is preferable to set up licensing before you install Build Forge[®] so you can immediately test the installation.

This section includes the following topics:

- Installing a License Server
- Installing Licenses

Installing a License Server

If you do not have an existing license server available, install and start a license server on a host that is accessible from all hosts where you intend to install a Management Console. You can download the Rational License Server from the IBM Rational support site,

<http://www-306.ibm.com/software/rational/support/>

Important

Configure the license server to start automatically on system boot. On Windows you must manually set the service to Auto. It is not done by default. For example, use **Control Panel** → **Administrative Programs** → **Services** to make that setting in Windows XP.

Installing Licenses

To install licenses, perform the following steps:

1. **Obtain the licenses.** Put the license files in a temporary directory on the host where the license server runs.

Important

The minimum you need is both a *server* license and *user* licenses. You also need license keys for optional licensed components, such as the Adaptor Toolkit and API. Your account representative can assist you with obtaining licenses. You may also get licenses from the IBM Rational support site. See **How to Get Your Rational License Keys** at <http://www-306.ibm.com/software/rational/support/licensing/>.

2. **Import the license keys to the server.** Use the IBM Rational tools:

- Windows: use the License Key Administrator (LKAD)
- UNIX or Linux: use the `license_update` script.

See *Rational License Management Guide* for instructions that apply to your host type and license type(s).

3. Configure the Management Console to use the license server.

- Windows: provide the license server name when prompted during installation.
- UNIX or Linux: put the license server name in `<bf-install>/Platform/bfkey.txt`. When you start the Management Console for the first time, The console reads `bfkey.txt`, connects to the license server, and then deletes `bfkey.txt`.

Changing the License Server to Use

If you need to change the license server after you have installed the Management Console, you need to make manual changes to Management Console and to license server settings.

1. In each running Management Console, go to **Administration** → **System** and change the License Server setting. Set it to the name of the host to use as the license server host.
2. Stop the Management Console engine
3. Change the value of the `RATIONAL_LICENSE_FILE` variable where it is set in your operating system. Set it to the name of the host to use as the license server host, using the format `port@hostname`.
 - **Windows:** the variable is stored in the registry at `HKEY_LOCAL_MACHINE\SOFTWARE\Flexlm License Manager`
 - **UNIX or Linux:** the variable is set in the `.flexlmrc` file.
4. Start the Management Console engine.
5. Open the Management Console to test that the engine was able to obtain a license from the new license server.

Installing a Standalone Deployment on Windows Using Launch Pad

Launch Pad allows you to install a standalone Build Forge deployment on a single Windows host. It automatically installs the included DB2 Express database when it installs the Management Console.

Launch Pad installs the following components:

- Management Console for your edition (either Standard, Enterprise, or Express) and the included DB2 Express database
- Agent for Windows

Launch Pad also gives you quick access to the installation guide for your edition.

Important

There are prerequisite steps and post-installation steps that need to be performed. Note them carefully in the sections below.

Launch Pad cannot be used to install any of the following. See other parts of this installation guide for the appropriate instructions.

- Management Console on a UNIX or Linux system
- Management Console on a Windows system where you want to use an *already-installed* database of any type, including DB2 Express.
- Agents on a non-Windows operating system

Accessing Launch Pad

Access Launch Pad from one of these places:

- download site
- Build Forge for Windows product CD

Download Site

1. Run a web browser on the Windows machine you intend to install the Build Forge system.
2. Locate the **BuildForge-CLPv2.1.0.zip** file on the download site.
3. Double-click the zip file to save and extract the files to a local directory.
4. Open the directory and double-click **launchpad.exe** to run the Launch Pad utility.

Use the Launch Pad to select and install Build Forge components.

Build Forge for Windows Product CD

Insert the Build Forge for Windows product CD in the CD drive. The Launch Pad utility runs automatically if auto-run is enabled.

Note

If auto-run is disabled, use Windows Explorer to view the contents of the CD. Double-click **launchpad.exe** to start Launch Pad.

Running Launch Pad

When you run Launch Pad you are given the following options. Follow the instructions below.

- **Management Console Installation**

1. Prerequisite step: before running the installation, create a user in Windows that the console uses to access DB2 Express. See [DB2 Express Setup](#) for details.
2. Use this option to Install the Management Console for Windows on the local host
3. Post-installation steps:
 - install JDBC drivers
 - check the PATH variable to be sure it includes the directory containing `db2cli.dll` and `sqlar.dll`.
 - See [Post-Installation Setup](#) for details.

- **Agent Installation**

- Use this option to Install the agent for Windows on the local host

- **Installation Documentation**

- Use this option to open a PDF of the installation guide for your edition.

Note

Your download file or CD contains only the edition you are licensed for. The Launch Pad for each edition can install only that edition. The Express Edition includes five user licenses. The Standard Edition and Express Edition require that a license server be installed first and loaded with your licenses.

Installing Management Console on Windows

This section describes how to install the Management Console on Windows platforms.

You proceed in the following steps:

1. Preinstallation setup: setup for your database. There is setup needed for the provided DB2 Express database as well as any existing database.
2. Build Forge Installation: run an installer to install Build Forge, including all required technologies (PHP, Perl, Apache, Tomcat, and database drivers). If you specify it, the provided DB2 Express database is also installed.
3. Post-installation setup: install JDBC drivers, check the PATH environment variable and (optional) configure PHP to use a proxy server.

Pre-Installation Setup

This section describes what you need to do before installing Build Forge on Windows. Perform the steps in this section only if you are setting up an existing database to be used by a Windows-based Management Console. The database can reside on the same host as the Management Console or on a different host.

Note

If you are using the DB2 Express provided by the Build Forge installer these steps do not apply to your setup.

In general you must complete the following tasks:

- Identify the database system you intend to use. Verify that it is one that Build Forge supports and that the necessary network and application connectivity (client drivers) exist between the Build Forge host and the database host. If a proxy server is required to access the database, get the proxy server name and path.
- If you intend to use international data, verify that the database is configured to use a character set that is compliant with UTF-8.
- Create database objects and permissions, in general as follows:
 - Database: It is named **build** in examples, but you can use a different name.
 - Database User: The Management Console uses this user name to access the database. The user is named **build** in examples, but you can use a different user name.

Note

For DB2 Express and DB2, the database user and password are not database objects. The database user and password must be set up as a user in your operating system.

- Database User Password
- Permissions for the **build** user to create tablespaces in the **build** database. Owner permissions are required in order to create, modify, and delete data.

Specific instructions are provided for each database type.

International Data Support (Windows)

Build Forge must be set up to support international data if you intend to display international data in the Management Console.

- **Web browsers:**

- must have the language set
- must have the fonts you use to display data installed

- **Agents:**

Build Forge recommends using the UTF-8 character set on agent servers.

On Windows, use the `chcp` command to check the code page:

```
> chcp
```

- **All databases:**

Typically support for international data is specified when you create the database.

The fonts you intend to use to display data must be installed. Support for the UTF-8 character set and an appropriate collation must be configured.

Build Forge requires the use of international data (UTF-8 character sets) with the exception of Microsoft SQL Server which does not support UTF-8.

- **DB2:**

1. Set the codeset and territory. Example: `CREATE DATABASE USING CODESET UTF-8 TERRITORY US` (or select the appropriate codeset and territory in Control Center).
2. Set `DB2CODEPAGE=1208`.

If an existing database has data in it that you need to migrate to UTF-8, the following document can help:

<http://publib.boulder.ibm.com/infocenter/db2luw/v9/index.jsp?topic=/com.ibm.db2.ucb.admin.doc/doc/t0024033.htm>

- **Microsoft SQL Server** (for use with Management Console on Windows only): *No support for international data.* Microsoft SQL Server uses UCS-2 for storing Unicode data and does not support UTF-8.
- **MySQL**: Set the server character set and collation. If your installation of MySQL does not currently support international data, you can recompile it from source and use `./configure --with-charset=utf8 --with-collation=utf8_bin`. The Build Forge engine will not start if this support is not configured.
- **Oracle**: Set the character set to **UTF8 - Unicode 3.0** on the instance when you install it. In the Database Configuration Assistant, the setting is made on the Initialization Parameters step on the Character Sets tab.
- **Sybase** (for use with Management Console on Windows only): Before creating any tables, configure the default character set to be `utf8` and the default sort order to be `utf8bin`. Consult the Sybase documentation for your version for more details. If the database server has data on it, follow directions for data migration to `utf8`.

DB2 Express Setup

Note

The Build Forge installer gives you the option of installing and setting up the provided DB2 Express database.

Choosing the provided DB2 Express database is the quickest database setup option for Windows.

During installation, make the following selections to setup the provided DB2 Express database:

- Use IBM Rational supplied database.
- The default database name is **build**. For DB2 Express, do not change the database name.
- The default user name is **build**. You can change it.
- A default password is not provided. Supply a unique password.

Note

Build Forge accesses the database using the schema for the user.

DB2 Setup

Use this procedure to set up support for DB2.

Database Objects for DB2

1. *In your operating system*, create a user. The Management Console will use this name to log in to the database. Example: user name **build**, password **build**.

Note

DB2 does not support creating a user from a SQL script.

Perform the remaining steps in DB2:

2. Create a user tablespace with 8K page sizes.
3. Create an empty database named **build** with the necessary tablespaces
4. Grant user access to BFUSE_TEMP tablespace.

Note

Build Forge accesses the database using the schema for the user.

Example 1. Sample DB2 SQL Command Script

Use the following commands at a DB2 Command Line Processor to create the tablespace and database.

```
// Create database
db2 CREATE DATABASE BFT ALIAS BUILD USING CODESET UTF-8 TERRITORY US AUTOCONFIGURE USING
MEM_PERCENT 40 APPLY DB ONLY
db2 CONNECT TO BUILD

db2 CREATE BUFFERPOOL "BUFFP1" IMMEDIATE SIZE 1000 PAGESIZE 8192 NOT EXTENDED STORAGE;
db2 CONNECT RESET;
db2 CONNECT TO BUILD;

// Create tablespaces
db2 CREATE SYSTEM TEMPORARY TABLESPACE TEMPSPACE2 PAGESIZE 8192 MANAGED BY SYSTEM
  USING ('<path to database>/SQL003.0')
  EXTENTSIZE 64
  PREFETCHSIZE 64
  BUFFERPOOL BUFFP1;

db2 CREATE USER TEMPORARY TABLESPACE BFUSE_TEMP PAGESIZE 8192 MANAGED BY SYSTEM
  USING ('<path to database>/SQL004.0')
  EXTENTSIZE 64
  PREFETCHSIZE 64
  BUFFERPOOL BUFFP1;

db2 CREATE REGULAR TABLESPACE USERSPACE2 PAGESIZE 8192 MANAGED BY SYSTEM
  USING ('<path to database>/SQL005.0')
  EXTENTSIZE 64
  PREFETCHSIZE 64
  BUFFERPOOL BUFFP1;

// User must be granted use of BFUSE_TEMP tablespace
db2 GRANT USE OF TABLESPACE BFUSE_TEMP TO USER BUILD WITH GRANT OPTION

db2 commit work;
db2 CONNECT RESET;
db2 terminate;
```

Microsoft SQL Server Setup

Use these procedures to install and configure access to a Microsoft SQL Server database from a Windows[®]-based Management Console.

Install/configure the following items using the instructions in the sections below.

Database Objects for Microsoft SQL Server

In these steps you create a user to be the database owner and the database for Build Forge to use.

1. Create a user to serve as the database owner.

Build Forge uses this username to log on to the database. Use **build** unless you must use a different name.

- a. Open SQL Server Management Studio.
- b. Open the database server in the Object Explorer (left pane).
- c. Right-click the **Security** folder and choose **New** → **Login** .
- d. In the Login - New dialog, specify the login name and choose options as follows. Important: uncheck User must change password at next login.
 - Choose **SQL Server authentication** and provide a password.
 - Uncheck **Enforce password expiration**
 - Uncheck **User must change password at next login**

2. Create the database.

- a. Open SQL Server Management Studio.
- b. Open the database server in the Object Explorer (left pane).
- c. Right-click the **Databases** folder and chose **New Database** .
- d. In the New Database dialog, specify parameters for the database:
 - Specify a Database name. Use **build** unless you must use another name. The name of the data and log files are updated automatically in the Database files box.
 - Specify the Database owner.
 - Click the **[..]** control to the right of the field.
 - In the Select Database Owner dialog, click **Browse** .
 - Check the name of the user you created, then click **OK** .

- Click **OK** in the Select Database Owner dialog.
 - Specify the Database files parameters. In the Database files table, do the following:
 - For both files: set the Initial Size to 500 (in MB)
 - For both files: set Autogrowth. In the Autogrowth column, click the [...] control to open the dialog. Check the **Enable Autogrowth** box, set growth to 500 MB, and select **Unrestricted Growth** , then click **OK** .
3. **Set the default database for the user.**
- a. Open SQL Server Management Studio.
 - b. Open the database server in the Object Explorer (left pane).
 - c. In Object Explorer, open **Security** → **Logins** .
 - d. Right-click the user you created and choose **Properties**
 - e. On the General page, select a Default database. Select the database you created.
 - f. Click **OK**

TCP/IP Setup on Microsoft SQL Server

TCP/IP must be enabled on Microsoft SQL Server in order to use it with Management Console. It is disabled by default on MS SQL Server 2005.

To enable TCP/IP on MS SQL Server 2005, do the following:

1. Open MSSQLServer Configuration Manager.
2. Under **SQL Server 2005 Network Configuration** , click **Protocols for MSSQLSERVER** .
3. Right-click **TCP/IP** , then choose **Enable** from the menu.

ODBC Data Source Setup for Microsoft SQL Server

In these steps you create an ODBC data source that is used by a Windows-based Management Console to access the database you created in Microsoft SQL Server.

1. From the Windows start menu, select **Settings** → **Control Panel** → **Administrative Tools** → **Data Sources** .

The **ODBC Data Source Administrator** dialog appears.

2. On the **System DSN** tab, click **Add** .

A list of drivers appears.

3. Select **MS SQL Server** from the list of drivers, then click **Finish** .

In the **ODBC Setup** dialog box that appears, enter the following information:

- **Data Source Name:** the name for this data source, *must be the same as Database Name..*
- **Description:** description for this data source.
- **Server Name:** host name of the host where the MS SQL Server database is installed.
- **Database Name:** database name you created above.

Click **OK** to close **ODBC Setup** , then **OK** to close **ODBC Data Source Administrator** . The data source is created.

Make a note of the following information. It is requested by the installation program when you install the Management Console.

- **Data source name:** as you assigned when creating the ODBC data source
- **Data source type:** SQL Server
- **User name:** user name you created for the database (for example, **build**)
- **Password:** password for the user name (for example, **build**)

Important

Use the same value for the Database Name and Data Source Name. A limitation in JDBC drivers requires this constraint. If they are not the same, the Quick Report reporting feature and the services-layer APIs for Java and Perl do not work.

MySQL Setup

Use this procedure to install and configure support for MySQL.

Install/configure the following items. Use the instructions in the sections below.

Database Objects for MySQL

1. Create an empty database named **build**.
2. Create a user associated with it (user name **build**, password **build**).

You could use the following commands to create the database **build** and create a user **build@localhost** with the password ("identified by") **build**:

```
mysql -u root
mysql> create database build;
mysql> grant all on build.* to build@localhost
-> identified by "build";
```


MySQL Configuration

- Increase maximum database connections to 200.

Edit `my.ini` as follows:

```
max_connections=200
```

The value should at least equal the total of your **Max Console Procs** and **Run Queue Size** system settings (in the Management Console's **Administration** → **System** page).

Oracle Setup

Use this procedure to setup support for an Oracle database.

Install or configure the following items. Use instructions in the sections below.

Database Objects for Oracle

Create a local user on the Oracle host: user name **build** and password **build**.

- Add appropriate grants, including CREATE SESSION and CREATE TABLE.
- Add an appropriate QUOTA size in the DEFAULT TABLESPACE, to provide enough space for the system to store data.

Important

During installation, the same value is used for the database name and the Oracle SID. A limitation in JDBC drivers requires this constraint. If they are not the same, the Quick Report reporting feature and the services-layer APIs for Java and Perl do not work.

Oracle Client Configuration

To install and configure the client for Oracle:

1. Install an Oracle client driver on the Management Console host.

If you need to use Instant Client, use the latest version of **Instant Client Package - Basic** for your operating system. Download it from Oracle at <http://www.oracle.com/technology/software/tech/oci/instantclient/index.html>.

2. Set up the environment on the Management Console host.

Several environment variables must be set.

- `LD_LIBRARY_PATH`: set to include the client installation directory.
- `NLS_LANG`: set to an appropriate value for international language support. It must include AL32UTF8. Example: `AMERICAN_AMERICA.AL32UTF8`.

- ORACLE_HOME: set to the path of your Oracle client installation directory.
- ORA_NLS: set to the path where character-set data is located *on the server*.
- ORA_NLS32: set to the same path as ORA_NLS.
- PATH: set to include the client installation directly.
- TNS_ADMIN: set to the path where the tnsnames.ora file is located *on the server*.

To check the current language setting on the Oracle server, log in to Oracle and execute the following command:

```
SQL> host echo $NLS_LANG
```

Important

During installation, tnsnames.ora is set up to use the same value for the database name and the Oracle SID. A limitation in JDBC drivers requires this constraint. If they are not the same, the Quick Report reporting feature and the services-layer APIs for Java and Perl do not work.

Oracle Client Example Configuration on Windows

Example environment:

- Instant Client - Basic at C:\instantclient_10_2, to use American English.
- Oracle 10.2 on a Windows system at C:\oracle\product\10.2.0\db_1, installed to support international data

Variable settings on the system where the client and Build Forge are installed:

- LD_LIBRARY_PATH includes C:\instantclient_10_2\
- NLS_LANG=AMERICAN_AMERICA.AL32UTF8
- ORACLE_HOME=C:\instantclient_10_2\
- ORA_NLS=C:\oracle\ocommon\nls\admin\data
- ORA_NLS32=C:\oracle\ocommon\nls\admin\data
- PATH includes C:\instantclient_10_2\
- TNS_ADMIN=C:\oracle\product\10.2\db_1\network\admin

Sybase Setup

Use this procedure to install and configure access to a Sybase database from a Windows-based Management Console.

Install or configure the following items using the instructions in the sections below.

Sybase Setup

In these steps you create a database, a database owner, and owner privileges.

1. Create the database.
Use **build** as the database name.
2. Create the database owner.
Use **build** as the owner's name and password.
3. Assign privileges to the database owner.

ODBC Data Source Setup for Sybase

In these steps you create an ODBC data source that is used by a Windows-based Management Console to access the database you created in Sybase.

1. From the Windows start menu, select **Settings** → **Control Panel** → **Administrative Tools** → **Data Sources** .

The **ODBC Data Source Administrator** dialog appears.

2. On the **System DSN** tab, click **Add** .

A list of drivers appears.

3. Select **Sybase** from the list of drivers, then click **Finish** .

In the **ODBC Setup** dialog box that appears, enter the following information:

- **Data Source Name:** the name for this data source, **Build Forge Sybase** suggested.
- **Description:** description for this data source.
- **Server Name:** host name of the host where the Sybase database is installed.
- **Database Name:** database name you created above.
- Connection Tab: set **Charset** to `utf8`.

Click **OK** to close **ODBC Setup** , then **OK** to close **ODBC Data Source Administrator** . The data source is created.

Make a note of the following information. It is requested by the installation program when you install the Management Console.

- **Data source name:** as you assigned when creating the ODBC data source
- **Data source type:** Sybase

- **User name:** user name you created for the database (for example, **build**)
- **Password:** password for the user name (for example, **build**)

Build Forge Installation (Windows)

Use this procedure to install the Build Forge Management Console on Windows platforms.

Prerequisite: Locate `mc-<version>-Windows.exe` on the download site or the product CD. Click the file to save it to a temporary directory. Double-click `mc-<version>-Windows.exe` to start the installation program.

Procedure: respond to the dialogs as follows:

1. *Welcome*

Click **Next** .

2. *License*

Click **I Agree** .

3. *License host and type of console* – fill in the requested information, then click **Next** .

- **License Host** : enter the host name where the Rational License Server runs.

You must enter a valid host name for the license server. If you plan to provide this information later, do not leave this field blank. Enter a character or value in this field. Leaving this field blank may result in an incomplete and unusable product.

- The default HTTP port for the Management Console is port 80. If port 80 is not available or you require a different port, enter the port number in **Http server port** .

Important

Do not use port 8080; it is the default port for Apache Tomcat.

- **Primary/Secondary Settings** : choose **Normal** .

4. *Database settings* – fill in the requested information, then click **Install** .

- **Location Name:** This name is displayed in the user interface when you log in. It is optional for Normal or Primary consoles, but *required* for Secondary consoles.

- **Use IBM Rational supplied database** : choose this option to install a new copy of the DB2 Express database.

The **DB Name** and **DB Type** fields are filled in. The default database name **build** is the required database name; do not change it.

- Fill in the following:

- **DB Host** - Use `localhost` or the host name for the current host.
- **Username** - the default is `build`; you may change the user name
- **Password** - supply a password for the user name

Note

The installer does not check to see if DB2 Express is already installed. The new database server is installed on the same host as the Management Console.

Note

If you are upgrading, the installer does not display your previous database settings, it displays the installer defaults. If you changed the defaults, you must re-specify your settings.

- **Connect to existing database** : choose this option if you have an existing database that you want the Build Forge system to use as its database. Fill in the following:
 - **DB Name**
 - **DB Type**
 - **DB Host**
 - **Username**
 - **Password**

5. If you elected to install DB2 Express above:

a. *Install DB2 Express.*

The installer starts automatically and displays a progress bar.

b. *Reboot Now*

When you are asked when to reboot, reboot now. DB2 Express requires a reboot before you can access it through Build Forge.

6. *Access the Management Console.*

Start a browser. Go to the URL for the Management Console:

- **General form:** `http://<hostname>[:<portnumber>]`. The port number is optional if you used the HTTP default, port 80.
- **Local:** if you are running the browser on the same host where the Management Console is running, you can use `http://localhost`.

Note

If you have installed a database other than DB2 Express and you cannot login, wait a minute or so and try again. On first startup the engine (bfengine) has to load the database schema.

Important

Do not stop bfengine immediately after an install. Doing so can corrupt the database schema. In that case you would need to drop all Build Forge tables and reinstall.

7. Log in.

Use user name **root**, password **root**. Change the root password immediately.

Post-Installation Setup

This section describes what you need to do after installing Build Forge on Windows. You need to perform the steps in this section only if you are setting up an existing database to be used by a Windows-based Management Console.

- Download and install JDBC drivers
- Check the PATH variable
- Identify the proxy server for PHP to use. This step is optional. Perform it only if the Management Console needs to go through a proxy server to access the database.
- Perl driver setting for Oracle 9
- DB2CODEPAGE setting for DB2 to enable international characters on non-English systems

Install JDBC Drivers

Download and unzip JDBC drivers for your database. Copy the jar file only to \$CATALINA_HOME/common/lib. (\$CATALINA_HOME is the Tomcat installation root and must be set as an environment variable). See the installation documentation for your driver for instructions.

- DB2 - download not needed. During installation, driver jar files are placed in `<bfinstall>/tomcat/common/lib`:
 - db2jcc.jar
 - db2jcc_license_cu.jar
 - icu4j-3_6.jar
- Microsoft SQL Server -
<http://www.microsoft.com/downloads/results.aspx?pocId=3&=JDBC&DisplayLang=en>.

Locate the link for the SQL Server 2005 JDBC driver. If you installed an earlier version of this driver, replace it with the 2005 driver. Build Forge no longer supports earlier SQL Server JDBC drivers.

- **MySQL** - <http://www.mysql.com/products/connector/j/> Click the link for **MySQL Connector/J 5.0**.
- **Oracle** - http://www.oracle.com/technology/software/tech/java/sqlj_jdbc/index.html Click the download link next to your version of Oracle - requires Oracle account registration
- **Sybase** - <http://www.sybase.com/products/allproductsa-z/softwaredeveloperkit/jconnect> Click **Download jConnect 5.5 / 6.0.5**. Download the version 6.05 driver. See the Installation Guide for Adaptive Server Enterprise to install the driver.

Check PATH Variable

The PATH environment variable needs to include the path to database client or driver DLLs. Check PATH manually for these databases:

- **DB2** - directory containing `db2cli.dll` and `sqlar.dll`
- **MySQL** - directory containing `libmysql.dll`
- **Oracle** - directory containing `oci.dll`

For other databases (Microsoft SQL Server, Oracle, Sybase), installing the client or setting up the ODBC connection takes care of this requirement.

Identify Proxy Server

Optional: this step is needed only if the Management Console needs to use a proxy server to access its database. PHP needs to be configured to use the proxy server.

- Edit the `php.ini` file. It is located in `<bf-install>/Apache/php`, for example `C:\Program Files\IBM\Build Forge\Apache\php`.

Add the following entries:

```
bf_proxyHost=<your_proxy_server_hostname>
bf_proxyPath=<your_proxy_path>
bf_symlinkPath=<symlink_to_proxy_path>
```

Perl Driver Setting for Oracle 9

This step is needed only for a Management Console using an Oracle 9.x database.

The `ora_ph_type` parameter in `<bf-install>/db.ini` must be set to a value that does not cause the driver to strip trailing blanks. The setting `ora_ph_type=96` is recommended. If set to a value that strips trailing blanks (for example, `SQL_VARCHAR`), you see an error when running Build Forge:

```
DBD::Oracle::st execute failed: ORA-01400: cannot insert NULL into  
("BLDFORGE"."BF_RESULT"."BF_ENV") The problem does not occur with Oracle 10.
```

DB2CODEPAGE Setting for DB2

This step is needed only for a Management Console using a DB2 database. It is not needed if you are using the DB2 Express database provided.

You must set DB2CODEPAGE=1208 in the OS environment in order to display international characters correctly on non-English systems.

Installing Management Console on UNIX or Linux

Installations on UNIX or Linux systems requires a separate database and require you to install a number of prerequisites to prepare the machine for installation of the system.

Perform the following tasks to install the system:

- Install a database server, create Build Forge database objects, and perform other database setup
- Compile and install Apache
- Compile and install PHP, including database drivers for your database
- Compile and install Perl
- Compile and install Apache Tomcat
- Install and configure the Build Forge system

Prerequisites

You need the following in order to perform an installation on UNIX or Linux:

- Internet access. If you do not have Internet access from the machine where you are installing Build Forge, you need to download files from a machine that does have access and transfer them to the Build Forge machine to complete the steps.
- C compiler that is valid and working on your platform (for example, the gcc compiler on Linux).
- make facility that is suggested by your compiler's manufacturer (for example, gnu-make for use with gcc)
- Privileges as `root`

Database Installation and Configuration

Use this section to install the database you intend to use with a Build Forge Management Console installation on UNIX or Linux.

A database must be installed and configured with database objects before you install and configure other technologies and Build Forge. You need to do the following:

In general you must complete the following tasks:

- Identify the database system you intend to use. Verify that it is one that Build Forge supports and that the necessary network connectivity exists between the database host and the Build

Forge host. If a proxy server is required to access the database, get the proxy server name and path.

- If you intend to use international data, verify that the database is configured to use a UTF-8 character set.
- Create database objects and permissions, in general as follows:
 - Database: It is named **build** in examples, but you can use a different name.
 - Database User: The Management Console uses this user name to access the database. The user is named **build** in examples, but you can use a different user name.
 - Database User Password
 - Permissions for the **build** user to create tablespaces in the **build** database. Owner permissions are required in order to create, modify, and delete data.

Specific instructions are provided for each database type.

International Data Support (UNIX/Linux)

Build Forge must be set up to support international data if you intend to display international data in the Management Console.

- **Web browsers:**
 - must have the language set
 - must have the fonts you use to display data installed
- **Agents**

Build Forge recommends using the UTF-8 character set on agent servers.

On UNIX/Linux, use the following command to check the locale and character set:

```
locale
```

You should see values that designate your language and character set. The following example is from a Solaris system where US English is the language and UTF-8 is the character set:

```
LANG=en_US.UTF-8  
LC_CTYPE="en_US.UTF-8"
```

- **All databases:**

Typically support for international data is specified when you create the database; international data support cannot be configured after database creation.

The fonts you intend to use to display data must be installed on the database host machine.

Build Forge requires the use of international data (UTF-8 character sets).

- **DB2:**

1. Set the codeset and territory. Example: CREATE DATABASE USING CODESET UTF-8 TERRITORY US (or select the appropriate codeset and territory in Control Center).
2. Set DB2CODEPAGE=1208.

If an existing database has data in it that you need to migrate to UTF-8, the following document can help:

<http://publib.boulder.ibm.com/infocenter/db2luw/v9/index.jsp?topic=/com.ibm.db2.uclb.admin.doc/doc10024033.htm>

- **MySQL:** Set the server character set and collation. If your installation of MySQL does not currently support international data, you can recompile it from source and use `./configure --with-charset=utf8 --with-collation=utf8_bin`. The Build Forge engine will not start if this support is not configured.
- **Oracle:** Set the character set to **UTF8 - Unicode 3.0** on the instance when you install it. In the Database Configuration Assistant, the setting is made on the Initialization Parameters step on the Character Sets tab.

DB2 Setup

Use this procedure to set up support for DB2.

Database Objects for DB2

1. *In your operating system*, create a user. The Management Console will use this name to log in to the database. Example: user name **build**, password **build**.

Note

DB2 does not support creating a user from a SQL script.

Perform the remaining steps in DB2:

2. Create a user tablespace with 8K page sizes.
3. Create an empty database named **build** with the necessary tablespaces
4. Grant user access to BFUSE_TEMP tablespace.

Note

Build Forge accesses the database using the schema for the user.

Example 2. Sample DB2 SQL Command Script

Use the following commands at a DB2 Command Line Processor to create the tablespace and database.

```
// Create database
db2 CREATE DATABASE BFT ALIAS BUILD USING CODESET UTF-8 TERRITORY US AUTOCONFIGURE USING
MEM_PERCENT 40 APPLY DB ONLY
db2 CONNECT TO BUILD

db2 CREATE BUFFERPOOL "BUFFP1" IMMEDIATE SIZE 1000 PAGESIZE 8192 NOT EXTENDED STORAGE;
db2 CONNECT RESET;
db2 CONNECT TO BUILD;

// Create tablespaces
db2 CREATE SYSTEM TEMPORARY TABLESPACE TEMPSPACE2 PAGESIZE 8192 MANAGED BY SYSTEM
USING ('<path to database>/SQL003.0')
EXTENTSIZE 64
PREFETCHSIZE 64
BUFFERPOOL BUFFP1;

db2 CREATE USER TEMPORARY TABLESPACE BFUSE_TEMP PAGESIZE 8192 MANAGED BY SYSTEM
USING ('<path to database>/SQL004.0')
EXTENTSIZE 64
PREFETCHSIZE 64
BUFFERPOOL BUFFP1;

db2 CREATE REGULAR TABLESPACE USERSPACE2 PAGESIZE 8192 MANAGED BY SYSTEM
USING ('<path to database>/SQL005.0')
EXTENTSIZE 64
PREFETCHSIZE 64
BUFFERPOOL BUFFP1;

// User must be granted use of BFUSE_TEMP tablespace
db2 GRANT USE OF TABLESPACE BFUSE_TEMP TO USER BUILD WITH GRANT OPTION

db2 commit work;
db2 CONNECT RESET;
db2 terminate;
```

Enabling International Characters with DB2CODEPAGE

You must set DB2CODEPAGE=1208 in the OS environment in order to display international characters correctly on non-English systems.

MySQL Setup

Use this procedure to install and configure support for MySQL.

Install/configure the following items. Use the instructions in the sections below.

Red Hat Linux 4 Requirements for MySQL

MySQL on Red Hat Linux 4 requires additional setup before configuration for Build Forge.

- Install compatibility shared libraries.

You need to install the `MySQL-shared-compat` package or RPM (but not both).

- Rename startup script.

You need to rename the startup script to cause it to cause MySQL to be loaded earlier in the startup process than it does by default. The following example should move it up enough.

```
mv /etc/rc3.d/S99mysql /etc/rc3.d/S50mysql
```

Database Objects for MySQL

1. Create an empty database named **build**.
2. Create a user associated with it (user name **build**, password **build**).

You could use the following commands to create the database **build** and create a user **build@localhost** with the password ("identified by") **build**:

```
mysql -u root
mysql> create database build;
mysql> grant all on build.* to build@localhost
-> identified by "build";
```

MySQL Configuration

- Increase maximum database connections to 200.

Edit `my.ini` as follows:

```
max_connections=200
```

The value should at least equal the total of your **Max Console Procs** and **Run Queue Size** system settings (in the Management Console's **Administration** → **System** page).

Oracle Setup

Use this procedure to setup support for an Oracle database.

Install or configure the following items. Use instructions in the sections below.

Database Objects for Oracle

Create a local user on the Oracle host: user name **build** and password **build**.

- Add appropriate grants, including CREATE SESSION and CREATE TABLE.
- Add an appropriate QUOTA size in the DEFAULT TABLESPACE, to provide enough space for the system to store data.

Important

During installation, the same value is used for the database name and the Oracle SID. A limitation in JDBC drivers requires this constraint. If they are not the same, the Quick Report reporting feature and the services-layer APIs for Java and Perl do not work.

Oracle Client Configuration

To install and configure the client for Oracle:

1. Install an Oracle client driver on the Management Console host.

If you need to use Instant Client, use the latest version of **Instant Client Package - Basic** for your operating system. Download it from Oracle at <http://www.oracle.com/technology/software/tech/oci/instantclient/index.html>.

2. Set up the environment on the Management Console host.

Several environment variables must be set.

- LD_LIBRARY_PATH: set to include the client installation directory.
- NLS_LANG: set to an appropriate value for international language support. It must include AL32UTF8. Example: AMERICAN_AMERICA.AL32UTF8.
- ORACLE_HOME: set to the path of your Oracle client installation directory.
- ORA_NLS: set to the path where character-set data is located *on the server*.
- ORA_NLS32: set to the same path as ORA_NLS.
- PATH: set to include the client installation directly.
- TNS_ADMIN: set to the path where the tnsnames.ora file is located *on the server*.

To check the current language setting on the Oracle server, log in to Oracle and execute the following command:

```
SQL> host echo $NLS_LANG
```

Important

During installation, tnsnames.ora is set up to use the same value for the database name and the Oracle SID. A limitation in JDBC drivers requires this constraint. If they are not the same, the Quick Report reporting feature and the services-layer APIs for Java and Perl do not work.

Oracle Client Example Configuration on UNIX or Linux

Example environment:

- Instant Client - Basic at `/usr/local/instantclient_10_2`, to use American English.

- Oracle 10.2 on a UNIX system at `/usr/local/oracle/product/10.2.0/db_1`, installed to support international data

Variable settings on the system where the client and Build Forge are installed:

- `LD_LIBRARY_PATH` includes `/usr/local/instantclient_10_2`
- `NLS_LANG=AMERICAN_AMERICA.AL32UTF8`
- `ORACLE_HOME=/usr/local/instantclient_10_2`
- `ORA_NLS=/usr/local/oracle/ocommon/nls/admin/data`
- `ORA_NLS32=/usr/local/oracle/ocommon/nls/admin/data`
- `PATH` includes `/usr/local/instantclient_10_2`
- `TNS_ADMIN=/usr/local/oracle/product/10.2/db_1/network/admin`

Apache Installation and Configuration

Use this procedure to download, install, and configure Apache web server for use with the Management Console on UNIX and Linux.

This section includes instructions for you to perform the following steps:

- Download Apache
- Install Apache
- Edit Apache configuration files
- Configure Apache for your database

Prerequisites

You need the following in order to perform the tasks in this section:

- Internet access. If you do not have Internet access from the machine where you are installing Build Forge, you need to download files from a machine that does have access and transfer them to the Build Forge machine to complete the steps.
- C compiler that is valid and working on your platform (for example, the gcc compiler on Linux).
- make facility that is suggested by your compiler's manufacturer (for example, gnu-make for use with gcc)
- Privileges as `root` (or the user that Apache runs as, if it does not run as root).

Download Apache

To obtain the Apache web server source code:

1. Download the Apache source code to the intended Build Forge host.

Do the following as a non-root user.

- Make a temporary working directory.
- Go to <http://www.apache.org/>
- Download the version marked Best Available Version
For example, at this writing the file name is as follows:

```
httpd-2.2.4.tar.gz
```

2. Extract the image.

```
$ gunzip -c httpd-2.2.4.tar.gz | tar xvf -
```

Note that tar is set to use *stdin* for the pipe (using the `-` parameter)

3. Go to the newly created working directory

```
$ cd httpd-2.2.4
```

Install Apache

1. Configure Apache for installation in the working directory you just created.

```
$ ./configure --prefix=/usr/local/apache-2.2.4
```

This step specifies where Apache will be installed. It is installed in `/usr/local` by default. The example shows how to put it in `/usr/local/apache-2.2.4`.

2. Compile Apache.

```
$ make
```

This step compiles executables in your local directory.

3. Install Apache (do as `root`).

```
# make install
```

This step must be performed as a user who has write privileges for the directory where Apache is installed (`/usr/local/apache-2.2.4` in this example). It is normally done as `root`. Your local administrative setup may vary.

Edit Apache Configuration Files

1. Edit Apache vhosts file.

Add information about Build Forge in `httpd-vhosts.conf`.

```
cd <apache-dir>/conf/extras/  
vi httpd-vhosts.conf
```

Add the following lines:

```
<VirtualHost *:80>  
    ServerAdmin build@yourdomain.com  
    DocumentRoot /usr/local/buildforge/ui/public  
    ServerName ausbuild01.yourdomain.com  
    ServerAlias build.yourdomain.com mc.yourdomain.com #optional server aliases  
    ErrorLog logs/ausbuild.error_log  
    CustomLog logs/ausbuild.access_log common  
</VirtualHost>
```

- `<VirtualHost *:80>`: leave as port 80 (or the port you run HTTP on locally)

Important

Do not use port 8080; it is the default port for Apache Tomcat.

- `ServerAdmin`: email address of the Build Forge administrator
- `DocumentRoot`: location of the entry page for the Build Forge application
- `ServerName`: server where the Build Forge application is installed
- `ServerAlias`: optional aliases for the Build Forge `ServerName` URL
- `ErrorLog`: Apache error log for the Build Forge application
- `CustomLog`: Apache error log for logging access to the Build Forge application

To make the Build Forge console pages your document root, update the `httpd.conf` file in Apache, as follows:

- Set `DocumentRoot` in Apache `httpd.conf` to `/usr/local/buildforge/ui`
- Modify the Apache `httpd.conf` and update the directive to point to `/usr/local/buildforge/u`

Configure Apache for Your Database

You need to add information to `httpd.conf`, depending on your database.

- DB2:
- MySQL:
- Oracle:

Apache Configuration for DB2

1. Add the following line to the beginning of the Apache startup script (normally `/etc/init.d/httpd` or `/etc/init.d/apache2`, depending on your distribution).

```
source /home/db2bf/sqlllib/db2profile
```

2. Add the following lines to `httpd.conf`:

```
PassEnv LD_LIBRARY_PATH
PassEnv CLASSPATH
PassEnv LIBPATH
PassEnv VWSPATH
```

Apache Configuration for MySQL

No additional configuration is required.

Apache Configuration for Oracle

Add the following lines to `httpd.conf`:

```
PassEnv LD_LIBRARY_PATH
PassEnv NLS_LANG
PassEnv ORACLE_HOME
PassEnv ORA_NLS
PassEnv ORA_NLS32
PassEnv TNS_ADMIN
```

Add the following lines to the script that starts Apache at boot time (commonly `/etc/init.d/httpd` or `/etc/init.d/apache2`), filling in for `<value>` :

```
export LD_LIBRARY_PATH=<value>
export NLS_LANG=<value>
export ORACLE_HOME=<value>
export ORA_NLS=<value>
export ORA_NLS32=<value>
export TNS_ADMIN=<value>
```

PHP Installation and Configuration

Use this procedure to download, install, and configure PHP for use with the Management Console on UNIX and Linux.

This section includes instructions for you to perform the following steps:

- Download PHP
- Install PHP
- Configure PHP

- Edit the Apache configuration file
- Identify proxy server to use to access the database (*optional*, needed only if the Management Console host needs to access the database through a proxy server).

Download PHP

To obtain PHP source code:

1. Download the PHP source code to the intended Build Forge host. Do the following as a non-root user.
 - a. Make a temporary working directory.
 - b. Go to <http://php.net/downloads.php/>
 - c. Download the gzipped version under Complete Source Code (file extension *.gz*)

For example, at this writing the file name is as follows:

```
php-5.2.1.tar.gz
```

2. Extract the image.

```
$ gunzip -c php-5.2.1.tar.gz | tar xvf -
```

Note that `tar` is set to use *stdin* for the pipe (using the `-` parameter)

3. Go to the newly created working directory.

```
$ cd php-5.2.1
```

Install PHP

This section describes how to compile and install PHP from source. If you have an existing installation of PHP and do not wish to recompile, you need only determine if the appropriate database drivers are installed. If you need to install a database driver, consult PHP documentation for the installation method to use. Install the database driver for the database to use with Build Forge, as follows:

- DB2: `ibm_db2` driver
- MySQL: `mysqli` driver
- Oracle: `oci8` driver

1. Configure PHP for installation in the working directory you just created.

```
$ ./configure --prefix=/usr/local/php-5.2.1 --with-<database>=shared \  
--with-apxs2 --with-ldap=shared --enable-mbstring --enable-shmop \  
--with-xml --with-zlib=shared
```

Replace `--with-<database>` as follows:

- **DB2:** `--with-ibm_db2[=dir]`. If `=dir` is not specified, the default value is used: `/home/db2inst1/sqllib`
- **MySQL:** `--with-mysqli[=file]`. The optional file parameter is a pathname to `mysql_config`.
- **Oracle:** `--with-oci8[=dir]` If `=dir` is not specified, the default value is used: `ORACLE_HOME`. If you are using Oracle Instant Client to connect to the database, use `--with-oci8=instantclient,lib` where `lib` is the path to the Instant Client lib directory.

Note the line-continuation character `\` in the code block. This step specifies where PHP will be installed and what options it will be installed with. It is installed in `/usr/local` by default. The example shows how to put it in `/usr/local/php-5.2.1`. This location is used in later examples.

2. Compile PHP.

```
$ make
```

This step compiles executables in your local directory.

3. Install PHP (do as `root`).

```
# make install
```

This step must be performed as a user who has write privileges for the directory where Apache is installed (`/usr/local/apache-2.2.4` in this example). It is normally done as `root`. Your local administrative setup may vary.

Configure PHP

1. Copy extension files to the extension directory.

The extension files for the database need to be copied from the repository up to the active extensions directory. The following example assumes that PHP is installed in `/usr/local/php-5.2.1`. Note that `<datestamp>` is a string of numbers.

```
$ cd /usr/local/php-5.2.1/lib/php/extensions/no-debug-non-zts-<datestamp>/
$ cp <db-extensions> ..
```

The `<db-extensions>` files correspond to your database for Build Forge, as follows:

- **DB2:** `ibm_db2.so`
- **MySQL:** `mysql.so` and `mysqli.so`
- **Oracle:** `oci8.so`

2. Edit the PHP configuration file `php.ini`.

The following example assumes that PHP is installed in `/usr/local/php-5.2.1`:

```
$ cd /usr/local/php-5.2.1/lib/  
$ vi php.ini
```

Add the following entries:

```
extension_dir=/usr/local/php-5.2.1/lib/php/extensions  
upload_tmp_dir=  
                <directory>  
  
extension=<db-extension-so>
```

Use the `<db-extensions-so>` file name (or file names) for your database, as follows:

- DB2: `extension=ibm_db2.so`
- MySQL: two entries -

```
extension=mysql.so  
extension=mysqli.so
```
- Oracle: `extension=oci8.so`

Note

the directory used for `upload_tmp_dir` must be writable by the user that the Apache web server runs as. Commonly this user is `nobody`, but your local administrative practice may vary.

Edit Apache Configuration File

1. Edit Apache configuration file. Add information about PHP in `httpd.conf`.

```
cd <apache-dir>  
vi httpd.conf
```

Add the following lines:

```
LoadModule php5_module modules/libphp5.so  
AddHandler php5-script .php  
AddType text/html .php  
DirectoryIndex index.php
```

Identify Proxy Server

Optional: this step is needed only if the Management Console needs to use a proxy server to access its database.

- Edit the PHP configuration file `php.ini`.

It is located in `<php-install>/lib`, for example `/usr/local/php-5.2.1`.

Add the following entries:

```
bf_proxyHost=<your_proxy_server_hostname>
bf_proxyPath=<your_proxy_path>
bf_symlinkPath=<symlink_to_proxy_path>
```

Perl Installation and Configuration

Use this procedure to download, install, and configure Perl for use with the Management Console on UNIX and Linux.

This section includes instructions for you to perform the following steps:

- Download Perl
- Install Perl

Download Perl

To obtain the Perl source code:

1. Download the Perl source code to the intended Build Forge host.

Do the following as a non-root user.

- Make a temporary working directory.
- Go to <http://www.perl.com/>
- Download the version marked Stable Production Release
For example, at this writing the file name is as follows:

```
stable.tar.gz
```

2. Extract the image.

```
$ gunzip -c stable.tar.gz | tar xvf -
```

Note that tar is set to use *stdin* for the pipe (using the - parameter)

3. Go to the newly created working directory. At this writing the stable production release was 5.8.8.

```
$ cd perl-5.8.8
```

Install Perl

1. Configure Perl for installation in the working directory you just created. Note the capital C in Configure.

```
$ ./Configure -Dprefix=/usr/local/perl-5.8.8 -Dusetthreads -Duseithreads \
             -ds -e
```

This step specifies the following:

- `-Dprefix=perl-5.8.8`: directory where Perl will be installed. It is installed in `/usr/local` by default. The example shows how to put it in `/usr/local/perl-5.8.8`.
- `-Dusetthreads`: required. Build Forge will fail to start with an `undefined symbol` error if this is not set.
- `-Duseithreads`: required. Build Forge will fail to start with an `undefined symbol` error if this is not set.
- `-ds`: required to run silently (without prompting—there are hundreds of prompts)
- `-e`: required to run silently

2. Compile Perl.

```
$ make
```

This step compiles executables in your local directory.

3. Install Perl. Do the installation as the `root` user.

```
# make install
```

This step must be performed as a user who has write privileges for the directory where Perl is installed. The directory is `/usr/local/perl-5.8.8` in this example. It is normally done as `root`. Your local administrative setup may vary.

Perl Setting for Oracle 9

This step is needed only for a Management Console using an Oracle 9.x database and the `DBD::Oracle` driver.

- Set `ora_ph_type`.

The `ora_ph_type` parameter in `<bf-install>/db.ini` must be set to a value that does not cause the driver to strip trailing blanks. The setting `ora_ph_type=96` is recommended. If set to a value that strips trailing blanks (for example, `SQL_VARCHAR`), you see an error when running Build Forge: `DBD::Oracle::st execute failed: ORA-01400: cannot insert NULL into ("BLDFORGE"."BF_RESULT"."BF_ENV")` The problem does not occur with Oracle 10.

Build Forge Installation and Configuration (UNIX/Linux)

Use this procedure to install the Build Forge Management Console on UNIX and Linux.

This section includes instructions for you to perform the following steps:

- Prepare Build Forge files
- Install Build Forge files
- Include Build Forge in system startup
- Set permissions for Apache (you need to know the user name that Apache runs as)
- Understand `bfinstall.pl` (`bfinstall` reference)

A reference section for the `bfinstall.pl` script is included.

Prepare Build Forge Files

To prepare the Build Forge files for installation:

1. Download the Build Forge tarball from IBM (or copy it from the installation media if it has been provided).

Do the following as a non-root user.

- Make a installation directory. Recommended: `/usr/local/buildforge`
- Copy the tarball from the installation media to the installation directory.

```
mkdir /usr/local/buildforge
cd /usr/local/buildforge
<download tarball>
```

2. Extract the image.

```
$ gunzip -c mc-<version>-<build>.tar.gz | tar xvf -
```

Example: `mc-7.0.1-2039.tar.gz`. Note that tar is set to use *stdin* for the pipe (using the `-` parameter)

Install Build Forge

1. Go to the newly created installation directory.

This example uses `/usr/local/buildforge`.

```
$ cd /usr/local/buildforge
```

2. Run the `bfinstall.pl` Perl script with the `--with-dbtype` parameter for your database type.

Values are:

- DB2: `---with-dbtype=db2`
- MySQL: `--with-dbtype=mysql`

- Oracle: `--with-dbtype=oracle`

The following example installs for DB2:

```
$ ./bfinstall.pl --with-dbtype=db2
```

The script asks you to confirm the installation of PHP and Perl you are using, then validates the versions and installed components of PHP and Perl. You may see messages passed through from your compiler as well as output from the script itself.

To make the Build Forge console pages your document root, update the `httpd.conf` file in Apache, as follows:

- Set `DocumentRoot` in Apache `httpd.conf` to `/usr/local/buildforge/ui`
- Modify the Apache `httpd.conf` and update the directive to point to `/usr/local/buildforge/u`

Include Build Forge in System Startup

In general you need to do the following to configure your system to start the Management Console at startup:

- Copy the `<bf-install>/rc/buildforge` script to `init.d`.
- Create run-level links from the run-level directory to the `buildforge` script (where it resides in `init.d`). Links are needed to start Management Console at one or more run levels (S links) and to terminate it on shutdown (K links)

The directory locations, directory names, run-level usage, and linking conventions vary slightly from platform to platform. Specific instructions for each platform follow.

Important

If you are running Management Console on the same host as the database it uses, be sure the database is started first.

- AIX
 - Init directory: `/etc/rc.d/init.d`
 - Run-level directory: `/etc/rc.d/rc2.d`
 - Default run level: usually 2. To verify: `grep initdefault /etc/inittab`
 - Linking: hard link, starts with K or S, followed by the script name.

Example:

```
cp buildforge /etc/rc.d/init.d
ln /etc/rc.d/init.d/buildforge /etc/rc.d/rc2.d/Kbuildforge
ln /etc/rc.d/init.d/buildforge /etc/rc.d/rc2.d/Sbuildforge
```

- Solaris

- **Init directory:** `/etc/init.d`
- **Run-level directory:** `/etc/rc3.d`
- **Default run level:** 3. Solaris has its own Service Management Framework, see its documentation for more information.
- **Linking:** Hard link, starts with K or S followed by a two-digit number and the script name. The system initializes services in numerical order, so you can control the start order. If you are running Management Console and its database on the same host, the database must be started first.

Example (note the K entries for each run level and the S entry for level 3):

```
cp buildforge /etc/rc.d
ln /etc/init.d/buildforge /etc/rc0.d/K02buildforge
ln /etc/init.d/buildforge /etc/rc1.d/K02buildforge
ln /etc/init.d/buildforge /etc/rc2.d/K02buildforge
ln /etc/init.d/buildforge /etc/rc3.d/S98buildforge
ln /etc/init.d/buildforge /etc/rc6.d/K02buildforge
```

- **Linux**
 - **Init directory:** `/etc/rc.d/init.d`
 - **Run-level directory:** `/etc/rc.d/rc3.d`
 - **Default run level:** usually 3. To verify: `grep initdefault /etc/inittab`
 - **Linking:** symbolic relative link, starts with K or S followed by a two-digit number and the script name. The system initializes services in numerical order, so you can control the start order.

Example:

```
cp buildforge /etc/rc.d/init.d
ln -s ../init.d/buildforge /etc/rc.d/rc0.d/K02buildforge
ln -s ../init.d/buildforge /etc/rc.d/rc1.d/K02buildforge
ln -s ../init.d/buildforge /etc/rc.d/rc2.d/K02buildforge
ln -s ../init.d/buildforge /etc/rc.d/rc3.d/S98buildforge
ln -s ../init.d/buildforge /etc/rc.d/rc5.d/S98buildforge
ln -s ../init.d/buildforge /etc/rc.d/rc6.d/K02buildforge
```

Set Permissions for Apache

Set the owner of `<bf-install>/ui/templates_c` to the Apache user

1. Go to the Build Forge installation directory.

```
$ cd <bf-install>/ui/
```

2. Set the owner as the user Apache runs as.

```
$ chown <apache-user> templates_c
```

Depending on your configuration, the Apache user may be one of the following: `apache`, `apache2`, `httpd`, `nobody`, `www`, `www-data`, `www-user`. If Apache is running, execute the following command to find out the user name: `ps -ef | grep httpd`.

3. Use `chmod` to set permissions for the user as follows.

- `ui: set +rx`
- all directories below `ui: set +r`
- `templates_c: set +rwx`

bfinstall.pl Reference

The `bfinstall.pl` script can take the following parameters:

- `--with-dbtype=dbtype[,DBlibdir]` Required, specifies the type of database being used as the Build Forge database. The `dbtype` must be one of the following values: `db2`, `mysql`, `oracle`. The `DBlibdir` directory
- `--with-perl=path` specifies the installation directory of the Perl installation you want to use with Build Forge.
- `--with-php=path` specifies the installation directory of the PHP installation you want to use with Build Forge.
- `--with-bfdir=path` specifies the installation directory of the Build Forge installation.
- `--use-CPAN=[y | n]` Access CPAN to install any necessary Perl additions or updates. Requires that you be connected to the Internet. If not set, the script prompts you.

The following example shows an entire session, with user input in bold. It was set up to show how to supply the path to an alternate Perl installation and to show how the script detects any missing PHP database drivers.

```
The current perl binary is /usr/bin/perl. Is this the same perl that
Build Forge will be using? n
```

```
Which perl will Build Forge be using? /usr/local/installtest/bin/perl
I'm re-executing myself with that perl now. Wait one...
```

```
Where did you unpack the buildforge tar file? /usr/local/buildforge
```

```
Checking perl modules (ALL are required):
```

```
Checking Perl version ... 5.8.8
Checking Perl module Config::IniFiles ... 2.38
Checking Perl module Data::Dumper ... 2.121_08
Checking Perl module DBI ... 1.52
Checking Perl module File::Basename ... 2.74
```

```
Checking Perl module File::Path ... 1.08
Checking Perl module File::Temp ... 0.16
Checking Perl module Getopt::Std ... 1.05
Checking Perl module HTML::Template ... 2.8
Checking Perl module IO::Select ... 1.17
Checking Perl module IO::Socket ... 1.29
Checking Perl module IO::File ... 1.13
Checking Perl module IPC::Open2 ... 1.02
Checking Perl module MIME::Lite ... 3.01
Checking Perl module POSIX ... 1.09
Checking Perl module Set::Crontab ... 1.00
Checking Perl module Sys::Hostname ... 1.11
Checking Perl module Text::Balanced ... 1.95
Checking Perl module Time::Local ... 1.11
Checking Perl module XML::Writer ... 0.602
Checking Perl module XML::Writer::String ... 0.1
Checking Perl module XML::XPath ... 1.13
MySQL: Checking Perl module DBD::mysql ... 3.0007
buildforge.conf exists and looks like this:
db_hostname localhost
db_database bf_71
db_username build
db_password build
db_type mysql
db_provider mysql

Checking PHP module and database compliance...

Is the php being used for the web server /usr/local/bin/php? y

Checking PHP version ... 5.2.1
Launching PHP checks ... launched

PHP mysql driver is INSTALLED
PHP mysqli driver is MISSING

Missing PHP drivers need to be added to the php.ini as
extensions and the extensions need to be copied into the
php extension_dir specified in the php.ini file.
These extensions can be build from the php source by adding
a configure parameter to add shared DB drivers
(look at the output of ./configure --help in the php sourcedir
for a list of parameters)

The php source can be downloaded from http://php.net/releases
If you are running a unix that has a package manager, check for
packages containing the php module for your database

The supported and tested version of php is 5.2.1.
If you need to run a later version of php, please contact
support
```

Apache Tomcat Installation and Configuration

Use this procedure to download, install, and configure Apache Tomcat for use with the Management Console on UNIX and Linux.

The Tomcat servlet/JSP container is required in order to run reports and to support the service layer Java and Perl APIs. This section includes instructions for you to perform the following steps:

- Install J2SE 5
- Download Tomcat
- Install JDBC drivers for your database
- Configure Tomcat for Build Forge

Install J2SE 5

J2SE 5 is a prerequisite for running Apache Tomcat.

To install J2SE 5:

1. Download J2SE 5.

You can get J2SE 5 from IBM or from Sun.

2. Install J2SE 5.

Follow the provider's instructions to install on your platform.

Note

`$JAVA_HOME` must point to this installation and be available in the environment used to start Apache Tomcat.

Download Tomcat

To download the Tomcat binaries:

1. Download version 5.5.9 of the Tomcat binaries to the intended Build Forge host.

Do the following as a non-root user.

- a. Make a temporary working directory where you want Tomcat to be installed, such as `/usr/local`.
- b. In a browser, go to <http://archive.apache.org/dist/tomcat/tomcat-5/archive/v5.5.9/bin/>
- c. Download and verify the `jakarta-tomcat-5.5.9.tar.gz` and `jakarta-tomcat-5.5.9.tar.gz.md5`
- d. Verify the download using the `md5` file.

2. Extract the image.

```
$ gunzip -c jakarta-tomcat-5.5.9.tar.gz | tar xvf -
```

Note that tar is set to use *stdin* for the pipe (using the `-` parameter)

3. Go to the newly created working directory

```
$ cd /usr/local/jakarta-tomcat-5.5.9
```

Install JDBC Drivers

Download and unzip the JDBC driver for your database. The download may contain many files and subdirectories. Locate the jar file for the JDBC driver and copy the jar file only to `$CATALINA_HOME/common/lib`.

`$CATALINA_HOME` is the Tomcat installation root and must be set as an environment variable. See the installation documentation for your JDBC driver for instructions.

- **DB2** - <http://www-306.ibm.com/software/data/db2/express/download.html>

Click the download link next to **DB2 Driver for JDBC and SQLJ** - requires IBM account registration. You also need to locate and install a licensing `.jar`, `db2jcc_license_cu.jar`.

- **MySQL** - <http://www.mysql.com/products/connector/j/>

Click the link for **MySQL Connector/J 5.0**.

- **Oracle** - http://www.oracle.com/technology/software/tech/java/sqlj_jdbc/index.html

Click the download link next to your version of oracle - requires Oracle account registration

Configure Tomcat for Build Forge

1. Create a `reports` directory in the Tomcat root directory.

It must be writable by the user that Tomcat runs as.

```
$ cd $CATALINA_HOME
$ mkdir reports
```

2. Copy `buildforge.conf` from the Build Forge directory to the Tomcat root directory.

This example uses `/usr/local/buildforge/Platform`.

```
$ cd /usr/local/buildforge/Platform
$ cp buildforge.conf $CATALINA_HOME
```

3. Copy the `.war` files for the reporting application and the services-layer APIs for Java and Perl.

```
$ cd /usr/local/buildforge/tomcat/webapps
$ cp quickReport.war $CATALINA_HOME/webapps
$ cp rbf-services.war $CATALINA_HOME/webapps
```

4. Copy `icu4j-3_6.jar` from the Build Forge lib directory to the Tomcat lib directory.

```
$ cd /usr/local/build/tomcat/common/lib
$ cp icu4j-3_6.jar $CATALINA_HOME/common/lib
```

5. Set maximum memory to 1GB or more.

Running reports requires 1GB (1024MB) heap size for the JVM. Set `JAVA_OPTS` to include the `-Xmx` (maximum heap size) setting. If you get Out of Memory errors while running reports (possible on large reports), increase this setting. That may in turn require you to add memory to the host.

```
JAVA_OPTS -Xmx1024M
```

6. Start Tomcat

```
$ $CATALINA_HOME/bin/catalina.sh start
```

Installing Agents

This section describes how to install, run, configure and troubleshoot agents.

You need to install an agent on *each host* that you want to use as a Server resource from the Management Console. An agent is a service that receives requests from the Management Console to run projects and steps.

Installing the Agent on Windows Platforms

To install the agent on Windows platforms:

1. Locate and start the agent installation program in the installation media, `win-bfagent-version.exe`.

Note

The Launch Pad starts this installation process when you choose **Rational Build Forge Agent Installation**.

2. If the installer detects an existing version of the agent, it prompts you to confirm to install over it. Click **OK**. Default: OK.
3. At the Welcome panel, click **Next**.
4. At the License Agreement panel, click **I Agree**.
5. At the Choose Install Location panel, set the Destination Folder you want, then click **Next**. The default location is recommended: `C:\Program Files\IBM\Build Forge\Agent`.
6. At the Configuration panel, choose the Agent Options you want, then click **Install**:
 - Choose one of: **Install as a service** (the default) or **Install User Mode Agent**. A user mode agent is necessary only if the agent must be able to run a GUI application.
 - **Enable Cygwin Support**

Default: No. If you use the Cygwin Linux[®] emulation environment, you can choose to install Cygwin support when you install the agent. Doing so causes the agent to use the Cygwin bash shell as the default console. If you install Cygwin support, be sure to do the following:

- During Cygwin installation, choose **DOS/text** line endings.
- In projects, use UNIX[®]-style syntax for commands.

Important

Cygwin works only with US ASCII. It does not support UTF-8, so it cannot be used with any other systems.

- **Port** : The port the agent uses to communicate with the Management Console. 5555 is the default.

7. At the Completing Setup panel, click **Finish** .

User Mode Agent

An agent installed as User Mode allows a user to interact with applications launched by a project.

User mode is an option offered for Windows agents only. The option is set during installation. It cannot be configured after installation. User mode has the following applications:

- *Collect input manually through a GUI application as the job runs.* This makes the job dependent on human input.
- *Troubleshoot projects and steps.* Output that is hidden during service mode operation is visible in user mode. Each step produces a console window while the step is running.

Keep in mind the following differences when setting up projects to use a machine in user mode:

- The user mode agent operates as the currently-logged-in user on the system. The agent is only active during the time that this user is logged into the machine. A server running a user-mode agent can't be used if the user is logged out.
- Steps run on a machine in user mode are visible to any users of the machine.
 - Each step opens a console window on the Windows machine where the agent is running. It displays command activity from the Management Console.
 - If you launch a GUI application from a step, the application window appears on the Windows machine where the agent is running. The Management Console waits until the application exits before continuing the step.

Alternative: Use the start command if you want the job to continue without waiting.

- Do not use the `_USE_BFCREDS` variable. Any steps that use this variable will fail.
- The user must have the following privileges. They are typically not available by default. They must be added explicitly.

```
SeInteractiveLogonRight
SeAssignPrimaryTokenPrivilege
SeImpersonatePrivilege
SeIncreaseQuotaPrivilege
SeTcbPrivilege
```

Silent Agent Installation on Windows

To perform an automatic and silent installation of an agent on Windows, use the `/s` option. Example:

```
win-bfagent-7.0.1.2305.exe /s
```

The silent installation uses the following settings. They cannot be modified.

- Overwrite existing install: yes
- Install location: `C:\Program Files\IBM\Build Forge\Agent`
- Install as a Service: yes
- Cygwin support: no
- Port: 5555

Note

The `/S` option is case sensitive.

Installing the Agent on UNIX and Linux Platforms

Follow the installation instructions for your platform:

- **AIX**

1. Use the `aix5-bfagent-version.tar.gz` file or `aix5np-bfagent-version.tar.gz` file.

The `aix5np` file does not contain support for PAM authentication. Use it for agents on System i.

Caution

If you install the `aixnp` agent to run as root, at runtime the agent will use an AIX authentication call to authenticate, using the Server Authentication credentials you specify. If install the agent to run as a different, then you must also use the `magic_login` setting in the `bfagent.conf` file to restrict access to it.

2. Unzip the file.

```
gzip -d gzipfilename.gz
```

3. Extract from the unzipped tarball.

```
tar xvf tarballname.tar
```

4. Install the agent.

```
cd extracted-agent-directory  
./install.sh
```

- **HP-UX**

1. Set a `PATH` value if one is not already set. Although HP-UX supports operation without a `PATH` value, problems occur when the agent tries to add to the path. You can set the

value by editing the system's `/etc/PATH` file, or by using one of the shell standard `PATH` set or append operations.

2. Get the `hpux11-bfagent-version.tar.gz` file from the installation media. Place it where you want the agent installed.

3. Unzip the file.

```
gzip -d gzipfilename.gz
```

4. Extract from the unzipped tarball.

```
tar xvf tarballname.tar
```

5. Install the agent.

```
cd extracted-agent-directory  
./install.sh
```

- **IBM System z**

1. Follow the instructions below to get the source pack and extract it.
2. Follow the instructions in the file `README.ZOS`, which is in the root directory of the extracted source pack.

- **Mac OS**

1. Get the `mac-bfagent-version.dmg` file from the installation media and place it where you want it.
2. Double-click the file to extract its components.

- **Red Hat Linux**

1. Get the `rh9-bfagent-version.rpm` file from the installation media.
2. Run the following:

```
rpm -iUvh rh9-bfagent-version.rpm
```

- **Solaris**

1. Use the `sol9-bfagent-version-sparc-opt.gz` file for Solaris 9 or Solaris 10 on Sparc.
2. Unzip the package:

```
gzip -d solN-bfagent-version-platform-opt.gz
```

3. At a command prompt, run:

```
pkgadd -d ./unzipped-package
```

- **Other Platforms - Compiling from Source**

If you need an agent for another platform, use the `src-bfagent-version.tar.gz` :

1. Extract the downloaded tarball
2. Run the configure script located in the src directory
3. Run make in the src directory

The source kit requires the GNU C compiler or the C compiler for your system. The source kit (and prebuilt agents that do not include an installer for the local system) includes an `install.sh` script to install the agent in the system's `inetd/xinetd` configuration files.

Caution

If an agent is compiled for AIX with the configure option `--without-pam`, authentication for the agent is turned off. If it is then installed with root privilege, it allows people to connect as any valid user regardless of the password that they specify. If you must compile an agent to run on an AIX system that does not use PAM, be sure to use a dedicated account for the agent, install it to run as that user, and use the `magic_login` setting in the `bfagent.conf` file to restrict access to it.

Installing the Agent on System i Platforms

To install the agent on System i platforms:

1. Use the `aix5np-bfagent-version.tar.gz` file.
2. Unzip the file.

```
gzip -d gzipfilename.gz
```

3. Extract from the unzipped tarball.

```
tar xvf tarballname.tar
```

4. Place the `bfagent` executable in the agent installation directory. Example: `/usr/local/bin`.
5. Edit the shell configuration in the `bfagent.conf` file. Set the `shell` value to `/bin/sh`.
6. Place the `bfagent.conf` file in `/etc`.
7. Configure the system to run `bfagent` at startup. The following set of commands is an example, based on the executable being in `/bin`.

```
CRTLIB BFAGENT
```

```

CRTSBSD SBSD(BFAGENT/BFAGENT) POOLS((1 *BASE)) TEXT('Build Forge Agent subsystem')

CRTJOBQ JOBQ(BFAGENT/BFAJOBQ) TEXT('Build Forge Agent job queue')

CRTUSRPRF USRPRF(BFAGENT) PASSWORD(*NONE) INLMNU(*SIGNOFF) LMTCPB(*YES) SPCAUT(*ALLOBJ)
TEXT('Build Forge Agent user profile')

CRTJOBDB JOBDB(BFAGENT/BFAJOBDB) JOBQ(BFAGENT/BFAJOBQ) TEXT('Build Forge Agent autostart')

USER(BFAGENT) RQSDTA('CALL PGM(QP2SHELL) PARM('/bin/bfagent' ' ' -s'))

CRTCLS CLS(BFAGENT/BFACLS) TEXT('Build Forge Agent job class')

ADDRTGE SBSD(BFAGENT/BFAGENT) SEQNBR(1) CMPVAL(*ANY) PGM(QCMD) CLS(BFAGENT/BFACLS)

ADDJOBQE SBSD(BFAGENT/BFAGENT) JOBQ(BFAGENT/BFAJOBQ) MAXACT(*NOMAX) SEQNBR(10)

ADDAJE SBSD(BFAGENT/BFAGENT) JOB(BFAGENT) JOBDB(BFAGENT/BFAJOBDB)

```

Running an Agent

This section describes how to set up an agent to run. It is normally run as an auto-starting service or daemon.

Running an Agent on Windows

The agent is normally installed as a service and is set to Auto so that it starts up at system start time. You must be logged on to the system where the agent is installed to start and stop it.

You can use the **Start** menu:

- **Start** → **Programs** → **IBM Rational Build Forge** → **Start Agent Service**
- **Start** → **Programs** → **IBM Rational Build Forge** → **Stop Agent Service**

You can also use the following commands at a command prompt:

- `net start bfagent`
- `net stop bfagent`

Running an Agent on UNIX, Linux, and MacOS

The agent is intended to be run as a service and needs to be restarted automatically at system restart.

Add a `bfagent` entry to your configuration of `inetd` or `xinetd` as appropriate. The following example is the `bfagent` entry in `xinetd.d` on a Linux system, where the agent is installed in `/usr/local/bin`:

```

# description: The IBM Rational Build Forge Agent serves build requests
#   from the IBM Rational Build Forge Management Console.
service bfagent
{
    disable          = no

```

```
    flags          = REUSE
    socket_type    = stream
    wait          = no
    user          = root
    server        = /usr/local/bin/bfagent
    log_on_failure += USERID
}
```

The agent can be run outside of the inetd/xinetd environment if necessary. To run it as a standalone daemon, use the `-s` option.

```
bfagent -s
```

When you use this option, the agent moves into the background and begins listening for connections. This line should be placed in a startup script so that the agent starts automatically on system start.

Running the Agent on System i

Note

The standard Build Forge agent port (5555) may be preassigned on a System i server. In this case, change the Build Forge agent port to an unassigned port before starting the agent. You can edit the BFAgent.conf file directly or use another method. For details, see [Changing the Agent Port](#).

You need to run the agent as a specified user. From the desired login context, run the agent standalone:

```
bfagent -s
```

To allow for authentication, configure the `magic_login` setting in the BFAgent.conf file. (Do not run the agent as the QSECOFR user.)

The agent runs as an i5/OS PASE program. PASE (Portable Application Solution Environment) is an environment included in i5/OS that allows the execution of AIX binaries and commands. It usually is installed by default.

- Execute `DSPSFWRSC` at a command line to see if it is installed.
- Load it from the installation CD if it is not installed, using option 33 of 5722SS1.

Commands in a step are interpreted by the PASE shell. You can also execute native commands using the following syntax:

```
system -biOE "<native commands>"
```

Important

Each `system` command in a step executes in its own process. This has implications for commands that work only within their own process.

For example, if you want to set library lists for a set of steps:

- You cannot simply use CHGSYSLIBL or ADDLIBL as step commands because they are native commands (not recognized by PASE).
- You cannot use the supported native command syntax (example: `system -biOE "ADDLIBL FLGHT400"`) in a step, because it changes the library list only for the command's own process. Subsequent commands and steps are not affected by the change.

Although you cannot set library lists for just a step, set of steps, or project, you can set them in the agent configuration. Doing so sets the library lists for all projects and steps. To set library lists, use a job description for the agent that has the needed library lists defined. The following example shows a job description that specifies `FLGHT400` and `FLGHT400M` in the library list:

```
10    UTLIB
20    QGPL
30    QTEMP
40    FLGHT400
50    FLGHT400M
```

The agent specifies this job description in its startup routine. For example, if it were named `BFAJOB`, the line in the system startup routine would be as follows:

```
ADDAJE SBSDB(BFAGENT/BFAGENT) JOB(BFAGENT) JOBD(BFAGENT/BFAJOB)
```

Note

This solution affects *all* commands (from any step and project) that are run on the Server resource associated with this agent.

bfagent Reference

The `bfagent` executable starts the Build Forge agent. It reads its configuration from a `BFAgent.conf` file in the same directory.

The syntax for the command is:

```
bfagent [-f configfile | -s]
```

Options

- | | |
|----------------------------|--|
| <code>-f configfile</code> | Run using the configuration file in <i>configfile</i> , rather than <code>BFAgent.conf</code> . This is a runtime option on UNIX or Linux. It is a debugging option when running the agent manually on Windows. It cannot be used for starting the service on Windows. |
| <code>-s</code> | Start as a standalone service. Can be used on UNIX or Linux only. Running this way is an alternative to starting <code>bfagent</code> through <code>inetd</code> or <code>xinetd</code> . |

Configuring the Agent

This section describes how configure the agent after installation.

Agent Configuration File

The agent configuration file, `BFAgent.conf`, provides runtime configuration of the Agent's operation. It contains comments that explain all of the possible options. The file is located in the agent installation directory.

- Windows default: `C:\Program Files\IBM\Build Forge\Agent\BFAgent.conf`
- UNIX and Linux default: `/etc/bfagent.conf`

Important

If you make any changes to the `BFAgent.conf` file in the installation directory, you must repeat them after any subsequent reinstallation or upgrade of the agent. The existing configuration file is overwritten during every installation.

You can specify an alternate configuration file:

- On UNIX or Linux systems, you can preserve agent configurations by using a configuration file outside of the agent installation directory. When you do this, use the `-f` command line option on the command that starts the agent. Example:

```
bfagent -f /opt/bfagent.conf
```

- On Windows systems, the service cannot be started with this option. It can be used only when running the agent manually. It is a tool for debugging.

Changing the Agent Port

If the agent is being installed on a server where port 5555 is already occupied, the agent port can be changed after it is installed.

On Windows systems:

1. Enter the registry editor. Choose **Start** → **Run**, then enter `regedit`.
2. Go to `HKEY_LOCAL_MACHINE\SOFTWARE\BuildForge Agent`.
3. Change the value of `AgentPort` to the port number you want.

On UNIX, Linux, and Macintosh systems:

1. Open the `/etc/services` file.
2. Change the value of `BuildForge Agent Port` to the port number you want.

Configuring a Different Shell

You can configure an agent to use a shell other than the default by editing parameters in the `BFAgent.conf` file.

For example, to change a Windows system to use the Korn shell provided by MKSTools, change the shell parameter as follows:

```
shell C:\MKSTools\mksnt\ksh.exe -L -c \"%s\"
```

The %s in this command line is replaced by the step command when the system sends a command to the server. In this case, the quotes are escaped by backslashes.

Running Agent Commands on a Network Share File System (Windows)

The Build Forge agent initially starts up with Windows system account credentials. To run commands, the agent later authenticates with Windows using Build Forge server auth credentials.

The server auth credentials are accepted for local commands but may fail for some commands that the agent must run on external, networked shared drives. For example, to modify files in a ClearCase dynamic view, the agent must access ClearCase files on a network shared drive.

The commands fail because the external file system ignores the agent server auth credentials; it only recognizes the agent's initial system account credentials.

If you experience problems running commands on a network shared drive, you have the following options:

Run commands using a server auth

To run commands using a Build Forge server auth with access to network shares, add the win_reexec_after_auth setting to the BFagent.conf file.

This is your only option if you want to use Build Forge server auth credentials to establish access to a network share.

The win_reexec_after_auth setting causes the agent to start a new process after authenticating with Windows. This forces the shared file system to recognize that the user credentials have changed by the agent.

When win_reexec_after_auth is set, the agent runs as a service and does not distinguish between commands that access network shares and those that do not, so you might notice a performance impact.

Run the agent in single user mode

During agent installation, set up the agent to run commands in single user mode (without Build Forge server auth credentials). Select the **Install User Mode Agent** option.

If the specified user is a member of the Administrator group, then the user's credentials must be specified using a server auth.

If the user is not an administrator, then use the magic_login setting in BFagent.conf to prevent unauthorized access to the agent.

When you log on to the Management Console, the agent starts up and runs as the user name you provide, which immediately

	authorizes access to the network shares using that user's credentials.
Run the agent as a service with a dedicated user account	<p>Set up the agent to run as a Windows service with a dedicated user account. This option restricts you to running as a single user account but does not require the agent to start a new process to re-authenticate, so there is no performance impact.</p> <ol style="list-style-type: none"> 1. On the Build Forge server, open the Windows Control panel to access the list of services (Administration Tools > Services). 2. Open the service for the IBM Rational Build Forge Agent. 3. Provide the user account information for the user you want to use to run agent commands. For example, the ClearCase admin user or other user with access to ClearCase dynamic views and VOBs).

BFAgent.conf Reference

The `BFAgent.conf` file stores settings for how the Build Forge agent runs. It is in the same directory as the `bfagent` executable.

Settings and internal defaults are all listed in the file. Inactive settings are commented out.

Settings

`allow IP-address-or-range [...]`

Used only with:

- Agents running on Windows
- Agents running in standalone mode on UNIX or Linux (-s option on startup).

Limits connections to the agent. Connections are allowed only from the IP addresses that match *IP-address-or-range*. By default connections are allowed from all addresses.

Specify one or more of the following:

- **IP Address:** fully-qualified IPv4 or IPv6 address. For example, for IPv4, 255.192.192.003. The specific IP address is allowed.
- **IP Address range:** partially-qualified IPv4 or IPv6 address. For example, for IPv4, 192.168 or 192.168.63. All IP addresses that match this qualification are allowed.

Note

If you are running the agent on a superserver like `inetd` or `xinetd`, use another method to control access. You may consider a firewall, TCP wrappers (`hosts.allow` and `hosts.deny`), or `xinetd`'s built-in filtering capability.

<code>ccviewroot</code> <i>root-path</i>	Sets the default view root for this host. See ClearCase documentation on <code>init</code> for more information. Internal defaults are as follows: <ul style="list-style-type: none">• Windows: <code>ccviewroot M:</code>• UNIX or Linux: <code>ccviewroot /view</code>
<code>command_output_cache</code> <i>size</i>	Causes the agent to cache output until it reaches the specified size (in bytes). The internal default is not to cache. Using a cache can significantly improve agent performance and reduce network overhead. The best size to use depends on how much output commands produce. Minimum value: 2048. A value of 2048 is used internally if the setting is less than that.
<code>cygwin</code>	<i>Used only with:</i> <ul style="list-style-type: none">• Agents on Windows Enables agent to work on a Windows host using Cygwin, a Linux-like environment. When using Cygwin, a number of Linux tools are available to the agent. When this setting is use, the <code>cygwin_script_magic</code> and <code>shell</code> settings may also need to be set. Example:: <pre>cygwin shell C:\cygwin\bin\bash.exe --login -c "%s" cygwin_script_magic #!/bin/bash</pre> The shell setting must match your installation of Cygwin.
<code>cygwin_script_magic</code>	<i>Used only with:</i> <ul style="list-style-type: none">• Agents on Windows when cygwin is set Specifies the <code>#!</code> line to use when executing steps. The default is <code>#!/bin/bash</code> .
<code>default_logon_domain</code>	Specifies the domain to use when an authentication request does not include a domain. If not specified, the agent machine's domain is used.
<code>disable_telnet</code>	Using telnet is recommended to test the agent connection.

	<p>For the agent, there is some built-in processing overhead associated with processing and correctly handling telnet control sequences.</p> <p>This setting allows you to disable the agent's normal internal handling of special telnet character codes which can slightly improve performance. (This setting is recommend for production environments.)</p>
<code>disable_transcode</code>	<p>Turns off processing the agent performs to convert international data if the operating system is not using UTF-8 encoding. (UTF-8 is recommended for the agent operating system to avoid mixed encodings and data corruption)</p> <p>If the operating sytem does not use UTF-8 encoding, to correctly communicate with other applications, the agent must convert data to the correct encoding for the operating system's locale settings.</p> <p>For operating systems that do not use UTF-8 this setting is recommended and can significantly improve the performance of the agent.</p>
<code>enable_agent_dll</code>	<p>Enables DLL process tracing, a debugging tool.</p>
<code>env_recursion_limit</code> <i>number-of-recursions</i>	<p>Sets the variable-replacement recursion limit for pre-parsing. If not set, the limit is 32.</p>
<code>extensions</code>	<p>Specifies path to external library of functions. The functions can be used as dot commands in a step. If not set, external libraries are not loaded.</p> <p>During parsing, the first token in the step command is taken as the function name. The second token is a string, and the third is an integer timeout value (in seconds).</p> <p>Requirement: dynamic loader support in the operating system. For example, <code>/usr/include/dlfcn.h</code> in UNIX or Linux. Default values used internally:</p> <ul style="list-style-type: none">• UNIX or Linux: <code>/usr/local/bin/bfextensions.so</code>• Windows: <code>c:\program files\ibm\build forge\agent\bfextensions.dll</code>
<code>getaddrinfo_using_addrconfig</code>	<p>Used only for running the agent as a standalone service on UNIX or Linux systems (<code>bfa_{gent} -s</code>). This setting makes the agent use <code>AI_ADDRCONFIG</code> when calling <code>getaddrinfo()</code> to select a listening interface. By default <code>AI_ADDRCONFIG</code> is not used.</p> <p>If set, the effect is for the agent to ignore interfaces that do not have a properly configured address. It listens only for interfaces that have a properly configured address.</p>

If not set (the default), the agent attempts to listen on all available interfaces, regardless of whether they are properly configured.

`lang lang-code`

Used only when:

- The Management Console does not provide a valid language

Sets the language used by the agent to write messages and command output. Normally it is not set explicitly because the agent uses the language specified by the Management Console. However, it can be useful if the desired locale is not available on the system. It is also useful as a backup, in case the Management Console fails to communicate a lang or communicates an invalid lang.

The internal default is en, as if it were explicitly set as follows:

```
lang en
```

`leave_tmp_file`

Used only with:

- Troubleshooting

This setting causes the temporary file used to hold step commands to be retained, rather than deleted after command execution. In troubleshooting, the file can be compared to the steps as they appear in the Management Console.

Note

Do not use this setting for normal operations.

`locale locale-code.charset-code`

Used only with:

- UNIX and Linux systems. Windows handles locales differently.

Sets the language and multibyte character set to be used by localized applications. This setting works by setting the LANG environment variable for the agent context.

To set up the agent to treat command output as US English UTF-8, use the UTF-8 locale for your operating system. For example, on Linux use the following representation.

```
locale en_US.UTF-8
```

To determine the correct representation of the UTF-8 locale for your operating system, run the locale -a command.

If not set, the agent uses the locale of the system. This setting is a convenience. It is especially useful if the default locale of

magic_login
user.encoded-password

the operating system is not the locale that you want the agent to use, or if changing the system locale to meet agent requirements is not practical.

The agent normally uses administrative privileges (such as root or admin) to log on to the operating system. The magic_login setting is an alternative to standard system authentication. It allows the system to authenticate your login with a single user name and password.

If the agent is run as the root or admin user, this setting is ignored and normal authentication is attempted.

Note

The agent runs all commands using the permissions of the user who started the agent, not the user name used to log in.

Used only with:

- Situations where running the agent with administrative privileges is not possible (for example, System i with i5/OS, or old UNIX systems that do not work with PAM).
- Situations where running the agent with administrative privileges is not permissible (due to security policies).

Note

Do not use special characters (e.g. %, \$, [,], {, }, ", or ') when creating passwords.

To configure a login for the agent:

1. Create a server authentication to use a user name and password. In the Management Console, go to **Servers** → **Server Auth** .

For this example the user name is **build** and the password is **MySecretPassword**.

2. Create a server that uses the agent. Associate the server authentication with this server (**Authentication** field).
3. Generate an encoded password for the agent. In the installation directory for the agent, execute `bfagent -P` with the desired password.

An SMD5 hash-encoded password is returned, as follows:

```
bfagent -P "MySecretPassword"
eca0b7f2f4fbf110f7df570c70df844e1658744a4871934a
```

4. In `BFAgent.conf`, set `magic_login` to use the desired user name and encoded password.

```
magic_login
build:eca0b7f2f4fbf110f7df570c70df844e1658744a4871934a
```

5. Start the agent.
6. Test the server connection. In **Servers** , select the server, then click **Test Server** .

`map drive-and-user-spec`; ...]

Specifies a mapped drive. Some systems may require drive mappings, for example because a shell is run from a shared drive. Mappings specified on the agent are performed before mappings specified by `_MAP` environment variables in the Management Console. Example of two drive mappings:

```
map X:=//host1/share;Z:=//host2/share(username,password)
```

`map_drive_is_failure`

If set, causes a step to fail upon encountering an unmapped drive specification, before step execution. If not set, steps ignore drive failures and attempt to execute the step. In that case you may need to take care that the failure generates a meaningful error message.

`monitor_pipe path`

Reserved. Do not uncomment this setting.

`no_preparse_command`

Disables the variable-expansion parsing that the agent normally performs on a command before passing it to the shell. See also the `_NO_PREPARSE_COMMAND` environment variable, which can be used for an single project or step.

`no_pty`

Used only with:

- Agents running on UNIX or Linux systems.

Can be used as an alternate method for fixing the tendency of the system shell to deadlock when interacting with the agent's pseudo-terminal. Commonly used with HP/UX and z/OS. Other methods are:

- Using an alternate shell
- Using the `nologonshell` setting

The `no_pty` setting disables pseudo-terminal allocation.

Note

Using `no_pty` has side effects on some commands. For example, the `ls` command returns output in a single column rather than three. If you use this setting, test thoroughly before deploying the job to a production environment.

`nologonshell`

Used only with:

- Agents running on UNIX or Linux.

Uses standard methods of requesting that the shell be a normal shell rather than a logon shell. This may not work on all platforms and in such cases, the `shellflag` setting may be used to pass flags to the shell in order to modify its behavior. Often used when logon shells provide verbose output, change environment settings in undesired ways, or attempt to communicate interactively with the user. Those behaviors are not desirable for the agent, because it runs as a user without being an interactive user.

Note

The Mac OS X 10.5 system uses `/bin/bash`, which does not respond to the presence or absence of `nologonshell`. Use `shellflag -l` to force it to use a login shell.

Note

The z/OS system always sources the `/etc/profile` script for both logon shells and non-logon shells. You may need to alter its contents or use another shell if its behavior does not work well with the agent.

See also the `shellflag` setting. Flags can be used to alter logon script behavior.

`port port-number-or-range [...]`

Used only with:

- Agents running in standalone mode on UNIX or Linux (`-s` option on startup).

Specifies the port that the agent uses to listen for connections with the Management Console.

Note

The port is set to 5555 by default. It is set for UNIX or Linux in `/etc/services`.

<code>shell <i>shell_name</i> [<i>options</i>]</code>	<p>Specifies the default shell. Internal defaults are as follows:</p> <ul style="list-style-type: none"> • <i>Windows</i>: <code>shell cmd.exe /q /c "%s"</code> unless the following settings are used: <ul style="list-style-type: none"> • If the <code>cygwin</code> setting is used, the default is <code>shell C:\cygwin\bin\bash.exe --login -c "%s"</code> • If the <code>cygwin</code> setting is not used, the default is <code>shell cmd.exe /u /q /c "%s"</code> • <i>UNIX or Linux</i>: the shell set for the user's account, or <code>/bin/sh</code> if the user's shell can not be determined. Note that you cannot specify parameters in this setting, but you can use the <code>shellflag</code> setting to pass them. The agent automatically forces the default to be a logon shell by inserting a hyphen. For example, <code>/bin/ksh</code> is sent as <code>-ksh</code>. If <code>shell</code> is set explicitly, then <code>nologonshell</code> is set implicitly. See <code>nologonshell</code>. • <i>System i</i>: Set the shell value to <code>/bin/sh</code> <p>This setting can be overridden from within a step. A step that starts with a line containing <code>#!</code> overrides the shell setting and the <code>nologonshell</code> setting is used to execute the step commands.</p>
<code>shell_compatible_undef_vars</code>	<p>Forces representation of undefined variables to be an empty string. If not set, the representation is the variable name for variables of format <code>\$VAR</code>, <code>\${VAR}</code>, or <code>%VAR%</code> and the empty string for <code>[\$VAR]</code>.</p>
<code>shellarg</code>	<p><i>Used only with:</i></p> <ul style="list-style-type: none"> • Agents running on UNIX or Linux. <p>Use this setting if it appears that commands are being scrambled. Some shells on Red Hat Linux Enterprise require this setting.</p> <p>The setting changes the way a command script is passed to the shell. Normally the script is passed through standard input:</p> <pre style="margin-left: 40px;">/bin/sh < /tmp/bfshellscript.sh</pre> <p>This setting causes scripts to be executed by passing them as parameters:</p> <pre style="margin-left: 40px;">/bin/sh /tmp/bfshellscript.sh</pre>
<code>shellflag <i>flag</i></code>	<p><i>Used only with:</i></p> <ul style="list-style-type: none"> • Agents running on UNIX or Linux.

Adds a flag when running a shell. Only one flag can be specified. It is commonly used to disable rc script processing in order to reduce output or undesired processing. Examples:

- csh and derivatives: use `shellflag -f` to disable rc script processing.
- bash: use `shellflag --noprofile` to disable profile script processing.

`update_path path`

Identifies the full path to the Build Forge agent executable. It is set automatically during installation. The directory is a default for the specific operating system or the installation directory you specify.

Note

Ignored on Windows agents. The update path is taken from registry keys. The keys are set during agent installation.

`win_reexec_after_auth`

Add this setting if you need to run agent commands on a network share file system using Build Forge server auth credentials. For example, to modify files in a ClearCase dynamic view, the agent must access ClearCase files on a networked shared file system. The Build Forge agent initially starts up with Windows system account credentials. To run commands, the agent later authenticates with Windows using Build Forge server auth credentials.

Without this setting, the network share only recognizes the initial Windows system account credentials and ignores the subsequent server auth credentials needed to access and write to files on the network share file system.

The `win_reexec_after_auth` starts a new process after authenticating with Windows again using the server auth credentials and forces the shared file system to recognize the changed credentials. When `win_reexec_after_auth` is set, the agent runs as a service and does not distinguish between commands that access network shares and those that do not, so you might notice a performance impact.

Troubleshooting Agents

This section describes procedures you can use to troubleshoot inoperative agents. Go through the procedures in order. If you are unable to get an agent working through these procedures, then please contact support.

Testing Host Name Resolution

Verify that the agent host can be reached from the Management Console host. Use the ping utility from the Management Console host to ping the agent host:

```
ping hostname
```

The following session was run on Windows where both the Management Console and the agent were installed. You should see a response like this:

```
C:\> ping localhost

Pinging jhgough.austin.ibm.com [127.0.0.1] with 32 bytes of data:

Reply from 127.0.0.1: bytes=32 time<1ms TTL=128
Reply from 127.0.0.1: bytes=32 time<1ms TTL=128
Reply from 127.0.0.1: bytes=32 time<1ms TTL=128
Reply from 127.0.0.1: bytes=32 time<1ms TTL=128

Ping statistics for 127.0.0.1:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 0ms, Average = 0ms
```

A message similar to the following one indicates a problem:

```
Unknown host
```

The problem lies in the network configuration of the Management Console host. Contact your network administrator.

Testing the Connection

You can test the connection to the agent by using telnet or using a test from the Management Console.

Testing from the command line:

1. Connect to the agent via telnet. If you are logged on to the host where the agent is running, you can use localhost as the host name.

```
telnet hostname 5555
```

You should see the following response:

```
200 HELLO - BuildForge Agent v7.0.1.buildnumber
```

2. Check authentication. Execute the following commands, using your login credentials:

```
telnet localhost 5555
username <user name>
password <password>
cmd ping
go
```

You should see a message like the following:

```
AUTH: set user account to <user name>
```

If the above tests work but builds are failing, and a test of your server shows a user authentication error, check the pluggable authentication modules (PAM) configuration. If you see a message like the following, proceed to the next step.

```
AUTH: unable to set user account to <user name>: unknown account (1)
```

3. Type the following commands:

```
cmd ping  
go
```

The output of the telnet session should look similar to this. In particular it should end with **RESULT 0 (success)**. This test was on an agent running on Windows.

```
300 DATA s 67  
AUTH: Running as: [SYSTEM] in domain [NT AUTHORITY] SID Type [User]  
300 DATA s 52  
AUTH: Running with Privilege: [Lock pages in memory]  
300 DATA s 66  
AUTH: Running with Privilege: [Adjust memory quotas for a process]  
300 DATA s 63  
AUTH: Running with Privilege: [Create permanent shared objects]  
300 DATA s 46  
AUTH: Running with Privilege: [Debug programs]  
300 DATA s 56  
AUTH: Running with Privilege: [Bypass traverse checking]  
300 DATA s 61  
AUTH: Running with Privilege: [Back up files and directories]  
300 DATA s 54  
AUTH: Running with Privilege: [Change the system time]  
300 DATA s 68  
AUTH: Running with Privilege: [Remove computer from docking station]  
300 DATA s 73  
AUTH: Running with Privilege: [Impersonate a client after authentication]  
300 HEARTBEAT 1  
300 DATA s 16  
PLAT: Windows XP  
250 RESULT 0  
PING: internal loopback test complete  
260 EOR
```

Testing from the Management Console:

Use this method only if a Server has been set up that uses the agent.

1. Go to **Servers** .
2. In the list of servers, click on the one to test.
3. Click **Test Connection**

After the test, the results are available in the **Test Results** tab.

A message similar to the following indicates a problem.

```
Could not open a connection to host on port 5555
```

Either the Management Console is unable to connect to the host or the agent is not functioning.

Troubleshooting an Agent on Windows

To troubleshoot on Windows, do the following:

1. Check for installed executable files. These files should be present in the agent installation directory:
 - `bfagent.exe`
 - `bfdispatch.exe`
2. Reinstall the agent. Most Windows agent problems are solved by reinstallation. It updates the executable files and restores registry keys.

Note

If you have a custom configuration, save `BFAgent.conf` outside of the installation directory, then restore it after installation.

Troubleshooting an Agent on UNIX, Linux, or MacOS

To troubleshoot an agent on UNIX, Linux, or MacOS, try the following steps.

1. Run `bfagent` from a shell. You should get the following response:

```
200 HELLO - Build Forge Agent v7.0.1.122
```

You should get messages if there are shared library problems. They can usually be resolved by setting the path correctly.

2. Check that the agent is listening. Use the following command (assuming the port is the default, 5555):

```
telnet localhost 5555
```

You should again get a 200 HELLO response. If you do not, check your systems network configuration. Verify that the `inetd` configuration is correct, or check with your Linux or UNIX system administrator.

3. Check authentication. Execute the following commands, using your login credentials:

```
telnet localhost 5555
username <user name>
password <password>
cmd ping
go
```

You should see a message like the following:

```
AUTH: set user account to <user name>
```

If the above tests work but builds are failing, and a test of your server shows a user authentication error, check the pluggable authentication modules (PAM) configuration. If you see a message like the following, proceed to the next step.

```
AUTH: unable to set user account to <user name>: unknown account (1)
```

4. Check PAM (Pluggable Authentication Module) configuration. This is a common issue on AIX platforms. Depending on your operating system, PAM is configured in one of two ways: with a line in the `pam.conf` file or with a file in the `pam.d` directory.

- **pam.conf** : see if `/etc/pam.conf` exists. If it does not, go to the instructions for `pam.d` below. If it does, create an entry for `bfagent`. Copy the lines for another application (`sshd` or `login` are good choices) and substitute `bfagent` for the `[application]` field.

The format for each line of the file:

```
[application] [when] [mode] [module]
```

The fields are as follows:

- `[application]` - name of the application that needs to authenticate users
- `[when]` - type of authentication request
- `[mode]` - actions to take (based on authentication success or failure)
- `[module]` - authentication module to invoke

The following is an example of entries copied from `login` to `bfagent`. Note that the module names may be different from system to system.

```
bfagent auth requisite          pam_authtok_get.so.1
bfagent auth required          pam_dhkeys.so.1
bfagent auth required          pam_unix_cred.so.1
bfagent auth required          pam_unix_auth.so.1
```

Important

Solaris 10 exception: delete any lines that specify a module of `pam_dial_auth`, for example `pam_dial_auth.so.1`. Agent authentication does not work if that module is included.

- **pam.d** : The `/etc/pam.d` directory contains several files, each named for an application. Within each file, each line is of the format

```
[when] [mode] [module]
```

Create a file named `bfagent` by copying it from another application (suggested: `sshd` or `login`).

Testing: after you set up the PAM entries, try to log in again as described in step 3.

For more information, see PAM documentation at <http://www.sun.com/software/solaris/pam/>.

Alternate Configurations

This section introduced you to Build Forge[®] components and terminology.

Configuring Management Console to Use an Alternate Port

You can run Management Console on a port other than the default (80).

To configure Management Console to run on an alternate port:

1. During installation, set the port to the value you want. If the system is already installed, you can change two settings in `<bf-install>/Apache/conf/httpd.conf`. For example, if myHost is your local machine and you want to run on port 81:

- `Listen 81`
- `ServerName myHost:81`

See also the comments in the httpd.conf file.

Important

Do not use port 8080; it is the default port for Apache Tomcat.

2. Start the console and log in as root (or a user name with administrative privileges)
3. Go to **Administration** → **System**. Change the following settings. The example is continued from step 1.
 - **Console URL**: set to the URL and port that Management Console runs on. In this example it is `http://myHost:81`. Do not use a trailing slash.
 - **QuickReport URL**: set to `http://myHost:8080/quickReport`. Do not use a trailing slash. The Management Console uses this URL to navigate to the Quick Report pages.
4. Stop and then start the engine.
 - Windows: **Start** → **Programs** → **IBM Rational Build Forge Management Console** → **Stop Engine Service** (and then **Start Engine Service**)
 - UNIX or Linux: Use the RC path/buildforge to stop and start the bfengine process. See [“Starting and Stopping the Engine on Linux or Solaris Systems” on page 135](#)

To make the Build Forge console pages your document root, update the httpd.conf file in Apache, as follows:

- Set DocumentRoot in Apache httpd.conf to `/usr/local/buildforge/ui`
- Modify the Apache httpd.conf and update the directive to point to `/usr/local/buildforge/ui`

Configuring a Console Cluster

You can set up multiple machines to run the Management Console, all communicating with the same database.

Clustering management consoles scales request handling.

Important

Clustering does not provide failover capability or other high-availability capability. It is intended only to scale request handling.

When you set up a cluster of Management Consoles, you need to create a *master* console and any number of additional *secondary consoles*.

- The installation process for a master console is the same as for an ordinary single Management Console installation.
- To set up a secondary console, use the procedure in this topic.

Note

If your master console uses the default DB2 database (or any database local to the master console machine), make sure the database server is configured to accept external connections via TCP.

Complete the following steps for each secondary console:

1. Install the Management Console on a different machine from the master console. During the installation, on the **Database Settings** page, select a non-default database and configure it with settings that match the database used by the master console.

The master console may use the default database (a DB2 Express installation local to the master console machine) or an external database, but all secondary consoles must be configured as using an external database (the master's database).

If the master console does not use the default DB2 database, then the database configuration for master and secondary machines is the same.

2. Configure database client software on the secondary console machine.

For example, if you are using DB2 as your database server, install the DB2 runtime client and catalog the system database to allow the client to find it.

Installing the System on VMware

If you want to test or demonstrate the system on a VMware instance, review the following guidelines before installing the system, to address system resource limitations that derive from running under VMware.

- Use a lightweight database, such as MySQL, rather than DB2. If you use the default DB2 database (or even your own installation of DB2), the combination of DB2 and the system consumes too many system resources to work under VMware.
- Set the memory used by VMware Workstation to run your virtual machine to at least 1GB. In VMWare Workstation, select **Edit** → **Preferences** → **Memory** to adjust this value.

Configuring Support for SSL and HTTPS

Configuring the Build Forge system to use SSL and HTTPS increases the security of the system. SSL includes endpoint authentication and data encryption.

By default the Build Forge system does not use SSL. You must configure the Management Console. If you have the Quick Report option, you also configure Apache Tomcat and Quick Report. The following list shows a summary of the tasks you perform:

- Management Console
 - Replace the Apache web server with a version that includes OpenSSL
 - Configure the Apache web server to use OpenSSL. This step requires that you recompile the binaries.
 - Create self-signed certificates.
- Tomcat Server (if Quick Report is installed)
 - Verify that `$JAVA_HOME` points to the JDK on the Management Console host.
 - Create a keystore and self-signed certificate.
 - Enable and configure the SSL connector element in `server.xml`.
- Quick Report (if Quick Report is installed)
 - On a running Build Forge system, use Management Console to change Quick Report settings. Use **Administration** → **Settings** .

This section includes the following topics:

- Configure SSL Support for the Management Console on Windows
- Configure SSL Support for the Management Console on UNIX/Linux
- Configure SSL Support for Tomcat (Quick Report)
- Edit System Settings in Management Console for SSL

Configure SSL Support for the Management Console on Windows

The Web browser client and the Apache Web server subcomponent of the Management Console communicate over an HTTP connection.

This section describes how to configure Apache for SSL to secure the HTTP connection. To do this, you must download some additional software, configure Apache for SSL, and create self-signed certificates.

These instructions assume that the Apache Web server subcomponent is installed on the Management Console host.

Stop the Build Forge Engine

Select Start > Programs > IBM Rational Build Forge Management Console > Stop Engine Service.

Download and Install Apache with OpenSSL

Apache-OpenSSL Obtain the SSL compiled version of Apache (use version 2.0.59) for Win32 platforms from a download site. For example: <http://hunter.campus.com>. Download the zip file and extract its contents to the Management Console host machine, then install Apache-OpenSSL. (Do not extract or install the SSL compiled version of Apache to the Build Forge directory.)

Java JDK The installer for the Management Console on Windows installs a version of the Java JDK that contains the keytool utility. Verify that \$JAVA_HOME points to the Java JDK installation so that the keytool is available to the Apache-OpenSSL application.

Configure Apache for SSL

1. Backup the Build Forge Apache binaries in `<bf-install>\Apache\bin`. Rename the bin directory to `bin_old`.
2. Backup the Build Forge Apache configuration directory: `<bf-install>\Apache\conf`. Rename the conf directory to `conf_old`.
3. Copy the downloaded Apache-OpenSSL bin directory (and its contents) to the Build Forge `<bf-install>\Apache` directory.
4. Copy the downloaded Apache-OpenSSL conf directory (and its contents) to the Build Forge `<bf-install>\Apache` directory.
5. Verify that the `openssl.cnf` file is in the new conf directory. If necessary, copy `openssl.cnf` to the new conf directory from `conf_old`.
6. Copy the following files from the Build Forge `<bf-install>\Apache\bin` directory to the `WINDOWS\system32` directory:

- ssleay32.dll
 - libeay32.dll
7. In the Apache-OpenSSL modules directory, locate the mod_ssl.so module included in the download.
 8. Copy the mod_ssl.so module to the Build Forge *<bf-install>*\Apache\modules directory.
 9. In the Build Forge Apache configuration directory (*<bf-install>*\Apache\conf), locate the Apache configuration file (httpd.conf).
 10. In the httpd.conf file, uncomment the following line by deleting the comment symbol (#):

```
#LoadModule ssl_module modules/mod_ssl.so
```
 11. In the Build Forge *<bf-install>*\Apache\conf, create a directory called ssl for storing the keys and self-signed certificates.
 12. In the Build Forge *<bf-install>*\Apache\conf, rename ssl.conf.example to ssl.conf. (The ssl.conf file is the Apache SSL configuration file.)

Create Self-Signed Certificates

1. At a command prompt, change to the *<Apache-OpenSSL_installdir>*\bin directory. (The openssl.exe file is the bin directory.)
2. Move the openssl.cnf file from the *<bf-install>*\Apache\conf directory to the *<bf-install>*\Apache\bin directory.

Note

The openssl.cnf file must be available to the openssl program to create keys.

3. Execute the following commands to create the buildforge.csr, buildforge.cert, and buildforge.key files.

```
#openssl req -config openssl.cnf -new -out buildforge.csr
#openssl rsa -in privkey.pem -out buildforge.key
#openssl x509 -in buildforge.csr -out buildforge.cert -req -signkey buildforge.key
-days 365
#openssl x509 -in buildforge.cert -out buildforge.der.crt -outform DER
```

4. Copy the key files from the *<Apache-OpenSSL_installdir>*\bin directory to the Build Forge *<bf-install>*\Apache\conf\ssl directory.
5. Open the ssl.conf file and verify that the path and file names for the buildforge.cert and buildforge.key files are correct.

Test the Apache SSL Configuration

Start the Build Forge engine in the foreground and look for any Apache startup errors:

Select Start > Programs > IBM Rational Build Forge Management Console > Start Engine (Foreground).

Test the Apache HTTPS connection to the Web Browser

Launch the Management Console and change the URL:

Select Start > Programs > IBM Rational Build Forge Management Console > Management Console.

In the address bar of the Web browser, change the URL http to https:

```
https://<hostname>
```

Configure SSL Support for the Management Console on UNIX/Linux

The Web browser client and the Apache Web server subcomponent of the Management Console communicate over an HTTP connection.

This section describes how to configure Apache for SSL to create a secure HTTPS connection. To do this, you must download some additional software, configure Apache for SSL, and create self-signed certificates.

These instructions assume that the Apache Web server subcomponent is installed on the Management Console host.

Stop the Build Forge Engine

If the Build Forge engine is running on the Management Console host, stop it by using one of the following options:

- If a Build Forge rc script exists (typically in /etc/rc.d/init.d):

```
$ /<rc_path>/buildforge stop
```

- If a Build Forge rc script does not exist, determine the process ID and use kill command:

```
$ ps aux | grep buildforge  
$ kill ${<PID>}
```

Download and Install OpenSSL

OpenSSL Locate the tar file that contains the latest OpenSSL source code on a download site. For example: <http://www.openssl.org>. Download the tar file to the Management Console host machine and unpack the file. (Do not extract file contents to the Build Forge directory.) To compile and install OpenSSL, follow the instructions on the download site.

Java JDK If you installed and configured Apache Tomcat for reporting, you already installed a version of the IBM or Sun Java JDK that contains the keytool utility. Verify that \$JAVA_HOME points to the Java JDK installation on the Management Console host so that the keytool is available to the OpenSSL application.

Configure Apache for SSL

1. Backup the Build Forge Apache binaries in `<bf-install>/Apache/bin`. Rename the bin directory to `bin_old`.
2. Backup the Build Forge Apache configuration directory: `<bf-install>/Apache/conf`. Rename the conf directory to `conf_old`.
3. Recompile the Build Forge Apache binaries with the OpenSSL compile option:

```
-- with_ssl=<path_to_openssl_binaries>
```

- The OpenSSL binaries are in the openssl directory.
 - For instructions on compiling Apache binaries, see the *Install Apache* topic in the *Build Forge Installation Guide*.
 - For documentation on OpenSSL compile options, see the Apache Web site (<http://apache.org>).
4. Verify that the `openssl.cnf` file is in the new conf directory. If necessary, copy `openssl.cnf` to the new conf directory from `conf_old`.
 5. In the openssl directory, locate the `mod_ssl.so` module included in the download.
 6. Copy the OpenSSL `mod_ssl.so` module to the Build Forge `<bf-install>/Apache/modules` directory.
 7. In the Build Forge Apache configuration directory (`<bf-install>/Apache/conf`), locate the Apache configuration file (`httpd.conf`).
 8. In the `httpd.conf` file, uncomment the following line by deleting the comment symbol (`#`):

```
#LoadModule ssl_module modules/mod_ssl.so
```
 9. In the Build Forge `<bf-install>/Apache/conf`, create a directory called `ssl` for storing the keys and self-signed certificates.
 10. In the Build Forge `<bf-install>/Apache/conf`, rename `ssl.conf.example` to `ssl.conf`. (The `ssl.conf` file is the Apache SSL configuration file.)

Create Self-Signed Certificates

1. At a command prompt, change to the directory where you installed OpenSSL (`<openssl_installdir>/openssl`).

2. Move the openssl.cnf file from the `<bf-install>/Apache/conf` directory to the `openssl_installdir>/openssl` directory.

Note

The openssl.cnf file must be available to the openssl program to create keys.

3. Execute the following commands to create buildforge.csr, buildforge.cert, and buildforge.key files.

```
#openssl req -config openssl.cnf -new -out buildforge.csr
#openssl rsa -in privkey.pem -out buildforge.key
#openssl x509 -in buildforge.csr -out buildforge.cert -req -signkey buildforge.key
-days 365
#openssl x509 -in buildforge.cert -out buildforge.der.crt -outform DER
```

4. Copy the key files from the OpenSSL `<openssl_installdir>/openssl` directory to the Build Forge `<bf-install>/Apache/conf/ssl` directory.
5. Open the ssl.conf file and verify that the path and file names for the buildforge.cert and buildforge.key files are correct.

Test the Apache SSL Configuration

Start the Build Forge engine in the foreground using one of the following methods. Look for any Apache start up errors.

- If a Build Forge rc script exists (typically in `/etc/rc.d/init.d`):

```
$ /<path_to_rc_file>/buildforge start
```

- If a Build Forge rc script does not exist:

```
$ <bf-install>/Platform/buildforge.pl &
```

Test the Apache HTTPS Connection to the Web Browser

Launch the Management Console and change the URL:

In the address bar of the Web browser, change the URL http to https:

```
https://<hostname>
```

Configure SSL Support for Tomcat (Quick Report)

The Web browser client and the Tomcat Web server subcomponent of the Management Console communicate over an HTTP connection. Tomcat is required to use the Quick Report reporting feature, a separately licensed option in Build Forge.

This section describes how to configure Tomcat for SSL to secure the HTTP connection. To do this, you must download some additional software, configure Tomcat for SSL, and create self-signed certificates.

These instructions assume that the Tomcat Web server subcomponent is installed on the Management Console host.

Stop the Build Forge Engine

Stop the Build Forge engine if it is running on the Management Console host

Windows Select Start > Programs > IBM Rational Build Forge Management Console > Stop Engine Service.

UNIX/Linux Stop the Build Forge engine by using one of the following options:

- If a Build Forge rc script exists (typically in /etc/rc.d/init.d):

```
$ /<rc_path>/buildforge stop
```

- If a Build Forge rc script does not exist, determine the process ID and use kill command:

```
$ ps aux | grep buildforge
$ kill ${<PID>}
```

Download and Install Java JDK

Java JDK When you installed and configured Apache Tomcat for reporting, you installed a version of the IBM or Sun Java JDK that contains the keytool utility. Verify that \$JAVA_HOME points to the Java JDK installation on the Management Console host so that the keytool is available to the OpenSSL application.

Note

Installing and configuring Apache Tomcat for reporting is a prerequisite to configuring SSL support for Tomcat (Quick Report).

Create Certificate Keystore and a Self-Signed Certificate

Execute the Java keytool command to create a keystore and self-signed certificate. (The keytool is included with the Java 1.5 JDK.)

Windows Run the keytool command from the <JAVA_HOME>\bin directory:

```
%JAVA_HOME%\bin\keytool -genkey -alias tomcat -keyalg RSA -keystore
<path_to_keystore>
```

UNIX/Linux Run the keytool command from the <JAVA_HOME>/bin directory:


```
$JAVA_HOME/bin/keytool -genkey -alias tomcat -keyalg RSA -keystore  
<path_to_keystore>
```

After executing the keytool command, you are prompted for a keytool password. You are also prompted for a key password for the certificate. The keytool password and the key password for the certificate must be the same. (You must provide the keytool password later when you configure Tomcat.)

Configure Tomcat for SSL

Configure your secure socket in the `$CATALINA_HOME/conf/server.xml` file. `$CATALINA_HOME` is the Tomcat installation directory.

Complete these steps for your Windows or UNIX/Linux installation:

1. Open the `server.xml` file in a text editor.
2. Locate the SSL connector element and uncomment the following lines:

```
<!--  
<Connector port="8443" maxHttpHeaderSize="8192"  
maxThreads="150" minSpareThreads="25" maxSpareThread="75"  
enableLookups="false" disableUploadTimeout="true"  
acceptCount="100" scheme="https" secure="true"  
clientAuth="false" sslProtocol="TLS"/>  
</Connector>  
-->
```

3. At the end of the SSL connector element, add the following attributes for the keystore and certificate:

```
keystoreFile="<path_to_keystore_file>"  
keystorePass="<password>"  
algorithm="IbmX509"
```

Note

If you installed the Sun JDK, the default algorithm is SunX509. So, you can delete the line for the algorithm attribute in Step 3.

Test the Tomcat SSL Configuration

Windows Start the Build Forge engine in the foreground:

In Windows, select **Start** → **Programs** → **IBM Rational Build Forge Management Console** → **Start Engine (Foreground)** .

Look for Tomcat start up errors.

Unix Start the Build Forge engine in the foreground (using one of the following options) and look for Tomcat startup errors.

- If there is an rc file (typically in `/etc/rc.d/init.d`):

```
<path_to_rc_file>/buildforge start
```

- If there is no rc file:

```
$ /<bf-install>/Platform/buildforge.pl &
```

Test the Tomcat HTTPS Connection

Use one of the following options to test the Tomcat HTTPS connection to the Web browser.

- Web browser test
1. Open a Web browser.
 2. In the browser address bar, type `https://tomcathost:8443`.

The default Tomcat home page should display.

- Reporting application test
1. Log in to the Management Console.
 2. Select the Reporting tab to verify that it works.
 3. For Quick Report, verify that the protocol is https.

Edit System Settings in Management Console for SSL

After you set up SSL support for the Management Console and set up SSL Support for Tomcat (Quick Report), configure the Management Console system settings.

Important

If you are not setting up the Management Console for SSL, do not change the following configuration settings. It is possible to configure SSL support for Tomcat and Quick Report and not configure it for Management Console. If you do that, it is not necessary to change the system settings.

In the Management Console, select **Administration** → **Settings** and edit the values for the following settings:

System Setting	Value	Description
Quickreport_url	https://<hostname>:8443/quickReport	The console uses this URL to connect to Tomcat and the Quick Report engine.
console_url	https://<hostname>:443	The URL of the console.
console_port	443	The console port used by the Apache Web server.
console_host	<hostname>	The host name of the machine where the console is running.

Silent Install Option for the Management Console on Windows

The silent install uses a response file to install the Management Console and your chosen database as a normal console or in a distributed environment as a primary or secondary console.

The user-generated response file you create contains the keywords and values that would normally be entered during an interactive installation. Using a response file, the installation can proceed without user input.

To perform a silent installation of the Management Console on Windows, run the following command:

- The name of the user-generated response file is required.
- The command must be issued from the machine on which you are installing the Management Console.

```
winbfconsole-<version>.exe /response_file=<file_name>
```

Create a response file that contains the following keywords and values. The file can be an ASCII.txt file or an XML file. The syntax for the response file is as follows:

- The first line in the response file must be: [response_file]
- Enter keywords and their values on one line.
- Separate keywords and values with an equal sign (no spaces). For example, install_path=C:\Program Files\IBM\Build Forge
- Keywords are case sensitive; enter them as they appear in the following table.
- Values are not case sensitive.

Keyword	Value
accept_license	Enter y or yes to indicate acceptance of the product license agreement. This value is required for all console types.
install_path	Enter the full installation directory path. The default (recommended) installation directory is C:\Program Files\IBM\Build Forge. This value is required for all console types.
license_host	Enter the host name or IP address (IPv4 only) for the Rational License Server. If you do not know the host name and plan to provide it later, do not leave this field blank. Enter any character or value in this field. Leaving this field blank may result in an incomplete and unusable product.

Keyword	Value
	This value is required for all console types.
console_type	<p>Enter normal (the default) as the console type. If you do not provide a value, normal is used. The values are:</p> <p><normal primary secondary></p> <p>The primary and secondary settings are for distributed console environments only.</p>
db_host	<p>Enter or the host name (or alias localhost) or IP address (or alias 127.0.0.1) for the Management Console machine. The IP address may be an IPv4 or IPv6 address as appropriate for the machine where the database runs.</p> <p>This value is required for all console types.</p>
db_type	<p>Enter the database type you intend to use for the Management Console host. The values are:</p> <p><mysql oracle db2 sybase sqlserver supplied></p> <p>Supplied refers to the DB2 Express database that the installer supplies and installs and configures for you.</p> <p>Review required database setup in Pre-Installation Setup.</p> <p>This value is required for all console types.</p>
db_database	Enter the name of the database created for console use.
db_username	<p>Enter the database user name.</p> <p>Create this name before you run the installer. For DB2 and DB2 Express, create a user in Windows (not in DB2).</p> <p>For the other database types, you create a database user name.</p> <p>See Pre-Installation Setup.</p> <p>This value is required for all console types.</p>
db_password	<p>Enter the password that you created for the database user name.</p> <p>This value is required for all console types.</p>
http_server_port	<p>Enter the HTTP port (the default is 80). If you do not provide a value, normal is used.</p> <p>If port 80 is not available or you require a different port, specify a different port number.</p> <p>This value is required for all console types.</p>

Keyword	Value
gdd_location_name	Enter a location name for the console. The location name is displayed in the user interface when you log in. <ul style="list-style-type: none"> • normal - optional • primary - optional • secondary - required (used in projects to designate which console to run on)
master_db_database	For the secondary console only, enter the database name created for primary console use.
master_db_host	For the secondary console only, enter the database host name for the primary console. The host name must be an absolute IP address or hostname, not an alias.
master_db_type	For the secondary console only, enter the database type for the primary console. <mysql oracle db2 sybase sqlserver >
master_db_username	For the secondary console only, enter the database user name for the primary console.
master_db_password	For the secondary console only, enter the password for the database user name for the primary console.

Setting Up Users

This section describes how to set up user accounts Build Forge[®] system.

The following topics are discussed:

- Root User
- Adding User Accounts
- Read-only User for Reports

Root User

The root user (the user whose login name is **root**) has special characteristics within the system:

- **Created upon installation:** the root user is the only default user created by the installation program. The default password is **root** (you should change the password immediately after installation).
- **No license required:** the root user does not consume a user license. No matter how many users are logged in, you can always log in as the root user. (When someone logs in as root, any other user already logged in as root is logged off.)
- **System time zone:** the root user's time zone is the default time zone of the Management Console. The time zone for other users, both in-system and LDAP users, is taken from the root user's time zone by default. Users can set their own time zone after logging in once. All times and logs reported in the system are expressed in the user's time zone.
- **All permissions:** the root user has all available permissions and can edit other user's properties. You cannot remove any access privileges from the root user. Although the root user is not a member of any access groups, the root user can view, edit, or use any data objects in the system.
- **Priority:** The root user is always a priority user.
- **Logging out current users:** the root user can log out users by clicking the **Logout User** button on the **Administration** → **Users** → **<UserName>** page.
- By default, LDAP users are assigned the same time zone as the root user. However, they can edit the time zone assigned to them after they log in once. The system remembers the new preference.

Creating and Editing Users

You can create users and assign properties to them via **Administration** → **Users**. You can also connect your system to an LDAP/Active Directory database to get user information. You manage user security permissions by assigning users to groups, so you must create some users in order to test security features.

Select **Administration** → **Users** to display a list of current users, with a user form below it. The system displays the Name, Login, Email, Limit, Activity (elapsed time since last user activity), and Time Zone of each user.

- To edit a user, click the user's name and edit the properties in the user form, then click **Save**.
- To create a new user, start entering properties in the user form when no user is selected. If a user is selected, click the **Add User** button to clear the form. Click **Save** when you are done editing user information.
- To log out a user, click the user's name and then click **Logout User**.
- To free up a fixed license seat, first log in as root. Click the user's name, then click **Purge Seat**. The console removes the user from the list of IDs counted toward the set of fixed licenses. The user is also logged off if he or she is logged in. For fixed licenses, the console counts the number of users who have ever logged in. Once the limit is reached, no new users can obtain licenses. Existing users must be deleted or purged in order to make room for another user. Purging a seat does not delete the user from the console. If the user logs back in, the fixed license count is increased. If used on a floating-license user, **Purge Seat** has the same effect as **Logout User**.
- To delete a user account, click the user's name and then click **Delete**.

If the Delete button is disabled, a scheduled job is owned by the user account and the user account cannot be deleted. If you want to delete a user account that has scheduled jobs, you must first delete the scheduled jobs.

The user record sets default properties for the user's experience of the system and controls the user's login name, password, and password expiration. The data for a user record can be entered into the system through the Management Console, or derived from an LDAP/Active Directory database.

Note

When you edit a user whose record is derived from an LDAP database, many of the fields on the User page are disabled. You must change these properties in the source database.

To add a new user, click the Add User button, edit the form, then click the Update button.

When you display a user record, three tabs are available:

- **Details** : Use this tab to edit most of the user properties. The available properties are described below.
- **Current Groups** : Displays the access groups the user is a member of, either directly or through a direct group being a member of another group.
- **Change Groups** : Displays the groups the user is a direct member of, and allows you to add the user to groups or remove the user from them.

User Details Form

The screenshot shows the 'User Details Form' with the following fields and values:

- Name: [Empty text box]
- Email: [Empty text box]
- User name: [Empty text box]
- Password: [Empty text box]
- Verified: [Empty text box]
- Limit: Unlimited (dropdown)
- Time Zone: Central Daylight Time (dropdown)
- Date Format: 01/31 02:30PM (dropdown)
- Uses screen reader: No (dropdown)
- Language: English US (dropdown)
- Calendar Start Day Of Week: Sunday (dropdown)
- Priority Login: No (dropdown)
- Password Expires: Yes (dropdown)

For each user, you can set these properties on the **Details** tab:

Name	The display name and label for the user.
Email	The e-mail address where the system can send e-mail notifications for this user.
User Name	The name used to log on to the Management Console.
Password	The password used to log on to the Management Console. The field is not displayed for the user currently logged on. Use this field to enter a new password or change the existing one. Enter the same password in the Verified field.
Limit	Sets the maximum number of jobs the user can launch in a day. When the limit is reached, the system displays messages indicating that the user's run quota has been exceeded. If the value is 0, the system allows the user to run any number of builds.
Time Zone	The user's time zone. The system uses the time zone of the root user as the default time zone for all times posted. By default, in-system users and LDAP users are assigned the same time zone as the root user. Users can edit the time zone assigned to them.
Verified	Repeat the password here to verify it.
Priority Login	If this option is checked, the user becomes a priority user. A priority user can always log in to the system; if there are no more available user licenses, the system logs out the user with the oldest session to make room for the priority user. The root user is always a priority user.
Date Format	Selects the user's preferred display format.
Language	Selects the user's language.

Password Expires	If this option is checked, then the user's password expires after a number of days have elapsed, as specified in the Password Expiration Days system setting.
Uses screen reader	If set to Yes, the interface is enabled to support screen reader features for vision impaired users such as dynamic highlighting and focusing.
Calendar Start Day of Week	Select the day of the week that the Schedule calendar displays first. The default is Sunday.

Creating a Read-only User for Reporting

Have your database administrator create a user with read-only privileges for use in generating reports from the database.

Verifying the Installation

This section describes how to test an installed and configured Build Forge[®] system.

It discusses the following topics:

- Creating a test project
- Setting up a server
- Running projects

Creating a Hello World Project

This topic describes how to create and execute a simple project to verify that Build Forge is set up properly.

1. Create a server authentication, selector, and a server object.
These objects are required before you can create a project.
2. Create a project named HelloWorld.

Select **Projects** . In the Project Details form at the bottom of the main content pane, enter HelloWorld as the **Project Name** and choose a selector. Click **Save Project**.

Creating HelloWorld


3. Add a step to the project named EchoHelloWorld.
 - a. Select the HelloWorld project. The system displays the empty step list for the project and a blank Step Details form.
 - b. In the Step Details form, enter a **Name** of EchoHelloWorld.
 - c. In the Command field, enter a command line that will write “Hello World” to standard output on your chosen server.

For example, the following command line works on Windows[®], Solaris, Linux[®], UNIX[®], and Macintosh OS X systems:

```
echo Hello World
```

Then click the **Save Step** button.

4. Run the project.

Select **Projects** to redisplay the project list, then click the  icon next to the HelloWorld project. The system displays the **Running** tab of the **Jobs** module, with the HelloWorld project listed as running.

Note

If the HelloWorld project is not listed, skip to step 6.

5. Click the **Refresh** link at intervals until the Hello World project disappears from the Running list.
6. Click the **Completed** tab.
7. Click the job tag for the job.

The default job tag for an initial job is BUILD_1. The system displays details for the run, with the step log at the bottom of the main pane.

8. Examine the log for the project.

In most Hello World examples, you would see the "Hello World" text in a console window or pop-up window. The Management Console does its work by sending commands to the agent process on the targeted server; the agent then sends the output from those commands back to the Management Console, which stores them in the logs.

The log has many sections; the relevant one is the final EXEC section. Click on it to display the results of your command:

```
EXEC
274 Jun 13, 2006 - 16:52 start [c:\BuildForgeTests\HelloWorld\BUILD_1@mcsystem] echo
Hello World
275 Jun 13, 2006 - 16:52 Performing variable expansion on command line
276 Jun 13, 2006 - 16:52 Hello World
277 Jun 13, 2006 - 16:52 end [c:\BuildForgeTests\HelloWorld\BUILD_1@mcsystem] echo
Hello World (0)
```

This project demonstrates that you have configured your system correctly, that projects can successfully access a server, run and generate output on a server. The `echo` command can be replaced with any command that can be run on the target server.

Configuring Servers

To make best use of the system's dynamic server selection, you must set up a number of data objects, in a particular order. This topic outlines the minimum requirements for using servers.

1. Create server authentications.

Server Authentications provide login names and passwords for servers. You can apply a server authentication to more than one server, so that you do not need to have unique logins for every server, and when you change a login, you can change it for a set of systems.

2. Create collectors for groups of servers.

Collectors both collect properties from servers, and assign properties to them. You can use a server without a collector; without a collector you must select a server based on default properties like BF_NAME.

3. Create selectors.

Consider creating the following types of selectors:

- **Name-based:** Create one selector for each server, which selects the server based on its host name. This allows you to choose servers by name.
- **Operating-system-based:** Create one selector for each type of operating system in your environment, so that your projects can choose servers by operating system.
- **Capacity-based:** You can get more specific by choosing servers based on available RAM or hard-disk space.

4. Install an agent on each machine that you intend to use as a Build Forge server.

5. Create a server in the Management Console for each machine you intend to use with Build Forge.

6. Test servers.

Use the **Test Connection** button to test your servers after you create them, and review their manifests to make sure that they have the properties you expect.

Server Authentication


Use server authentications to associate login credentials to a server. You can use the same credentials for many servers, and update the credentials globally, by managing a set of server authentications.

A server authentication stores a login name and password as a single named object which you can associate with one or several servers. Use the Server Authentication page to create and edit server authentications.

Running Projects

There are several different ways to launch a project.

This task assumes that you have already created a selector, server, and project.

- While viewing the list of projects, click the  icon in front of any project to start the project immediately.

You cannot use this method if a project has no steps, or if it has any environment variables with the **Must Change On Project** action. Running a project this way uses its default values for selector, class, tags, and environment variables.

- While viewing the project's steps, click **Start Project** .

This method displays the Start Project page for the project, where you can change project parameters, environment variable values, and select steps to exclude from the run:

- Select new values for project parameters.
- Edit the project tag variable values.
- Edit the project environment variable values. If you want your changes saved as the new defaults for these variables, click the **Save Environment** check box.
- Select the Steps tab to display the list of project steps. You can select individual steps to exclude them from this run only.

When you have made your choices, click the **Execute** button to start the project.

- Select **Jobs** → **Start** and then click the project name.

As when you use the **Start Project** button, this method displays the Start Project page.

Start Project Page

The screenshot shows the 'Start Project' page with the following details:

- Job Details** / **Job Steps** tabs.
- Project Parameters:**
 - Selector: Choose local
 - Class: Production
 - Tag Format: BUILD_\$B
 - Tag Example: BUILD_1
- Project Environment:**
 - DotDateWeekday: .date %A
 - DotDateMonth: .date %B
 - DotDateHour: .date %l
 - DotDateYear: .date %Y
- Save Environment:**
- Project Tags:**
 - Editable Tags
 - B: 1

While a project is running, view the **Jobs** → **Running** page to check the project status.

To view job results, select **Jobs** → **Completed** to display the completed jobs. Click the Tag Name to access options for viewing job results.

Integrating with Other Products

This section describes how to integrate the Build Forge[®] system with other products. It integrates with several IDEs (Integrated Development Environments) using plug-ins.

The following topics are discussed:

- Eclipse integration
- IBM Rational Application Development integration
- Microsoft Visual Studio integration

About Plug-ins for IDEs

You can use plug-ins to connect to a Management Console from your integrated development environment (IDE).

The plug-ins allow developers to access Management Console features from a supported integrated development environment (IDE) on a client machine (a developer's personal machine, rather than the Management Console).

The set of projects that a developer can access through the plug-ins is defined by the access groups that the developer user is a member of. If the steps within a project have different access groups assigned, that can also limit the developer's access. Note that the plug-ins do not allow developer users to change projects or steps, no matter what their access properties, so you can freely place developers in access groups to allow them to launch or view projects.

Each user who wants to use a plug-in needs a license.

This release of the plug-ins supports the Eclipse[™], Rational[®] Application Developer, and Microsoft Visual Studio IDEs.

Plug-ins for Eclipse and Rational Application Developer

Plug-ins provide access to Management Console features from within Eclipse[™] or Rational[®] Application Developer IDEs.

The following plug-ins are available for the Eclipse and Rational Application Developer environments:

Available Plug-ins

Frequency

The Frequency plug-in provides developer self-service features. Through it, a developer can launch and view those projects allowed by his access groups.

When it is installed in your IDE, you can perform the following actions:

- Access one or more Management Consoles to view preconfigured projects
- View build logs
- Launch builds (including test builds using the developer's local files)
- View status of running builds

Reflector

The Reflector plug-in allows developers to run test builds: builds that use files from the developer's local workspace in a remote build configured through the management console. This allows you to test code changes against the actual production build process without checking them in, or see how your local code performs in a test, integration, or release build, without impacting others. You can even apply environment variable overrides, so that different values are used when you run your version of the project.

Eclipse plug-in users have the option to override values for project environment variables. The Job Settings pop-up displays when you start a Build Forge project. Changes to environment variables apply to the job only; default variables values for the project are not changed.

Using the Plug-ins in Eclipse or Rational Application Developer

After you install the plug-ins, you can activate them in the following ways:

- To access Management Consoles to launch jobs and view project logs, use the Frequency plug-in. *Within your IDE*, select **Window** → **Open Perspective** → **Other**. Your IDE displays a dialog box with a list of perspective types; select the Build Forge perspective. The system displays the Console Explorer, Build Info, and Build Log windows. Right-click on the **Console Explorer** and select **New Console** to configure a connection to a Management Console. For more information on using Frequency, see the on-line help provided with the plug-in.

Note

If you need to configure access to an LDAP/Active Directory domain, make sure you use the Build Forge system name for the LDAP domain object, not the actual name of the domain.

- To run test builds, use the Reflector plug-in. *Within your IDE*, configure Reflector by selecting your project and right-clicking. Select **Properties** from the pop-up menu. In the **Properties** dialog's list of properties options, select **Build Forge Project Artifacts**. Configure the dialog with the Build Forge project you want your project to work with, and select files to be uploaded to the system. For more information, see the on-line help provided with the plug-in.

Note

The Reflector plug-in requires the Frequency plug-in.

Installing the Eclipse or Rational Application Developer Plug-ins

Install the plug-ins from your IDE environment by connecting to an update site.

Use the following procedure to install the plug-ins in Eclipse™ 3.02 or later, or Rational® Application Developer version 6.0 or later. Perform these steps from within your IDE.

Note

The Build Forge system must be running.

Note

The Eclipse plug-ins require Eclipse version 3.02 or later. Eclipse is a free, open-source IDE. You can download Eclipse from <http://www.eclipse.org>.

1. Select **Help** → **Software Updates** → **Find and Install** .
2. Click the **Search for new features to install** radio button.
The system displays the **Update Sites to Visit** dialog.
3. Click the **New Remote Site** button.
The system displays the **New Remote Site** dialog.
 - a. Enter “Build Forge Plug-In Update Site” in the name field.
 - b. Enter the following update site URL in the **URL** field, using the hostname of your Management Console machine: `http://<console_host_name>/prism/site.xml`
 - c. Click **OK** .
4. The system displays a list of available plug-ins. Select all the offered plug-ins.

Note

The Reflector plug-in requires the Frequency plug-in.

5. In the **Update Sites to Visit** dialog, select the new **Build Forge** check box and click its + sign to expand it. Verify that the **Build Forge Prism Plug-ins** check box is selected.
6. Click **Next** .
7. Respond to the prompts provided by the installation script.

Plug-in for Microsoft Visual Studio

The plug-in provides access to Management Console features from within Microsoft Visual Studio IDEs. This release supports Visual Studio 2003 and Visual Studio 2005.

You can access the following features via the Visual Studio plug-in:

- Access one or more Management Consoles to view preconfigured projects (as provided by the security settings relevant to your Build Forge login).
- View build logs.
- Launch builds.
- View status of running builds.
- Run *pre-flight* builds: builds that use files from the developer's local workspace in a remote build configured through the management console. This allows you to test code changes against the actual production build process without checking them in, or see how your local code performs in a test, integration, or release build, without impacting others.

Using the Plug-in

After you install the plug-in, start Microsoft Visual Studio. The **Console Explorer** , **Build Info** , and **Build Log** windows appear.

- To access Management Consoles to launch jobs and view project logs, use the plug-in's windows inside Visual Studio. Configure access to a console in the **Console Explorer** ; and view jobs and logs in the **Build Info** and **Build Log** windows. For more information on using Frequency, see the help provided with the plug-in.
- To run pre-flight builds, *within your IDE* configure your project by selecting your project and right-clicking. Select **BuildForge Project Settings** from the pop-up menu. For more information, see the help provided with the plug-in.

Installing the .NET IDE Plug-in

Install the plug-in by running its .msi file and then starting your IDE.

Use the following procedure to install the plug-in on a client system with Microsoft Visual Studio 2003 or 2005 installed.

1. If you have an earlier version of the plug-in installed, uninstall it (using **Start** → **Control Panel** → **Add/Remove Programs**).
2. Exit Visual Studio.
3. Download a copy of the `BuildForgeVSNETSetup.msi` file. Double-click it to start the installation.

If you are installing for Visual Studio 2003, you can choose the installation location. For Visual Studio 2005, the plug-in is automatically installed to the required location.

4. Start Visual Studio.

Your Visual Studio window displays the **Console Explorer** , **Build Info** , and **Build Log** windows which derive from the plug-in. The plug-in appears as a line item on the Add-in Manager tool (**Tools** → **Add-in Manager**) and on the options panel (**Tools** → **Options** → **BuildForge**). By default, the plug-in's windows are displayed whenever you start up Visual Studio.

Note

If you close the plug-in's windows, you can redisplay them by selecting **Tools** → **Options** → **BuildForge** and making sure the check box for each window is selected, then clicking **OK**.

Special Variables for Test Projects

When you run a test build of a project using a plugin, you can use some special environment variables to run commands before and after files from your system are copied to the server.

All commands are run in the project directory:

- PRECMD variables are used to run a command on directories and files that are copied from the developer's machine to the server running the build. The command runs before the project step. Example: this command could check out files from a source control system before they are copied.
- POSTCMD variables are used to run a command on directories and files after a project step has executed. Example: this command could be used to free a checked-out virtual directory (in a source control system that uses such a concept, like Rational ClearCase).

Commands are run on directories and files marked in a Reflector plug-in as Build Forge Project Artifacts. The commands are applied as the directory tree for the reflector is traversed.

Note

Traversal of the directory tree is breadth-first downward for PRECMD commands and reversed for POSTCMD commands. Commands for directories and commands for files are run as appropriate during traversal.

<code>_PRISM_DIR_PRECMD</code>	Specifies a command to be run on directories as they are encountered during tree traversal. The command is run once for every directory that contains at least one file. If the <code>\$1</code> token is used in the command, it is replaced the name of the directory the command is running on. Only the first occurrence of <code>\$1</code> is handled this way.
<code>_PRISM_FILE_PRECMD</code>	Specifies a command to be run on files as they are encountered during tree traversal. The command is run once for every file. If the <code>\$1</code> token is used in the command, it is replaced the name of the file the command is running on. Only the first occurrence of <code>\$1</code> is handled this way.
<code>_PRISM_DIR_POSTCMD</code>	Specifies a command to be run on directories as they are encountered during tree traversal. is run once for every directory that contains at least one file. The command is run once for every directory in the project that contains at least one file. If the <code>\$1</code> token is used in the command, it is replaced the name of the directory the command is running on. Only the first occurrence of <code>\$1</code> is handled this way.

`_PRISM_FILE_POSTCMD` Specifies a command to be run on files as they are encountered during tree traversal. The command is run once for every file. The system replaces the first `$1` in the command with the file name.

Upgrading an Existing Version

This section describes how to upgrade an existing version to version 7.0 or later.

The following topics are discussed:

- Upgrading Management Console on Windows
- Upgrading Management Console on UNIX and Linux
- Upgrading Agents
- Upgrading Client APIs
- Migration Script Reference

Upgrading the Management Console on Windows

This procedure describes how to upgrade prior versions of the Management Console on Windows to the latest version.

Before proceeding, check the following list:

- *Version:* Determine whether your current installation is version 3.8 or 7.0 or later (with any iFix).
- *Version 3.8 with included MySQL 3.2.3 database:* **Important** - You must install MySQL 5.0 and migrate Build Forge data to MySQL before upgrading. You can then use MySQL 5.0 as your Build Forge database.
- *Changing platforms:* If you are changing to a new machine and/or operating system along with the upgrade, treat the upgrade as a new installation, then export data from your old system and import it to the new one.
- *Preserving data:* Do not uninstall your existing version of Build Forge. It must be upgraded in place. Uninstalling the existing installation destroys the project and log data in your database.
- *Preserving custom interfaces/adaptor templates:* If you have custom interfaces (now known as adaptor templates), rename the `<bf-install>\interface` directory before performing the upgrade. Your custom work can be easily applied to the new adaptor templates.
- *Required disk space:* If you are migrating from Build Forge version 3.8 with MySQL, to version 7.0 or later with the DB2 Express, allow at least triple the current database size plus 2 GB.
- *Database:* The database server must be running in order to complete the upgrade.
- *Notification Templates:* In version 7.0.1, the URL in the notification templates changed. If you are upgrading from 7.0, your notification templates might not work. If you experience an error, see [Troubleshooting Common Problems](#).

Upgrading from Version 7.0 or Later on Windows

These procedures apply for any version 7.0 or later installation, including those with an iFix applied. The iFix must have been correctly installed.

Before you start:

- Stop the Tomcat server. Execute the shutdown command in the Tomcat bin directory:

```
cd <bf-install>\$CATALINA_HOME\bin
shutdown
```

- If you are running Build Forge in the foreground, stop the engine. If Build Forge is running as a service, the installer stops it automatically.
- Copy the installation image (`mc-<version>-Windows.exe`) to the Build Forge Management Console host.

To start the upgrade, double-click the installation image file. The installer starts. Respond to the following prompts.

Note

Default installation values are presented; they may not match the values in your `buildforge.conf` file. Enter the values in your *existing* `buildforge.conf` file.

1. Welcome

Click **Next**

2. License Agreement

Click **I Agree**

3. Choose Install Location

Enter the path of your current Build Forge installation.

4. Build Forge Management Console Configuration

Enter the following information, then click **Next** .

- License Host - Enter the name or IP address of the license server host.
- Primary/Secondary Settings - Click **Normal** .

5. Database Configuration

Choose **Connect to existing Database** , then provide the following information. Click **Install** when done.

- DB name - name of the database

- DB type - type of database (one of DB2, MySQL, Oracle, SQL Server, Sybase)
- DB host - host name or IP address of the host where the database is located
- Username - login name of the database user that Build Forge uses
- Password - password for the login name

6. Installing

Wait for the installer to finish. Click **Finish** .

7. Post-Install Steps

Perform the post-installation steps in [Post-Installation Setup](#).

Upgrading from Version 3.8 on Windows

To upgrade from version 3.8 to version 7.0 or later, you must perform the tasks in the following topics:

1. Run the Windows Installer.
2. Run the bfmigrate script to update your database.
3. If you are migrating from a version of Build Forge that used the included MySQL database, run the bfdbmigrate script.

Run the Windows Build Forge Installer

Before you start:

- If you are upgrading from Build Forge 3.8 with the included MySQL 3.2.3 database, before you run bfmigrate you must install MySQL 5.0 and migrate your Build Forge data. Migration to MySQL 5.0 must include setting up a user for Build Forge to use to connect to the MySQL 5.0 instance.
- If you intend to migrate your database from MySQL to DB2 Express, the Build Forge installer creates a database named **build** and a Windows user **build** to use to log in to DB2 Express; you supply a password.
- If you are running Build Forge in the foreground, stop the engine. If Build Forge is running as a service, the installer stops it automatically.
- Copy the installation image (`mc-<version>-Windows.exe`) to your current Build Forge Management Console host.

Double-click the installation image file to start the installer. Respond to the following prompts.

Note

Default installation values are presented; they may not match the values in your `buildforge.conf` file. Enter the values in your *existing* `buildforge.conf` file.

1. Welcome

Click **Next**

2. License Agreement

Click **I Agree**

3. Choose Install Location

Enter the path of your current Build Forge installation.

4. Build Forge Management Console Configuration

Enter the following information, then click **Next** .

- License Host - Enter the name or IP address of the license server host.
- Primary/Secondary Settings - Click **Normal** .

5. Database Configuration

- If you are upgrading from version 3.8 with the provided MySQL database and intend to use the DB2 Express database provided with version 7.0 or later, choose **Use IBM Rational Supplied Database** , then provide the following information.

The **DB Name** and **DB Type** fields are filled in. The default database name **build** is the required database name; do not change it.

- **DB host** - use `localhost` or the host name of the current host.
- **Username** - the default is `build`; you may change the user name
- **Password** - supply a password for the user name
- *In all other cases*, choose **Connect to existing Database** , then provide the following information. Click **Install** when done.
 - **DB name** - name of the database
 - **DB type** - type of database (one of DB2, MySQL, Oracle, SQL Server, Sybase)
 - **DB host** - host name or IP address of the host where the database is located
 - **Username** - login name of the database user that Build Forge uses
 - **Password** - password for the login name

6. Installing

- Wait for the installer to finish. Click **Finish** .

Run the bfmigrate Script

Run bfmigrate to migrate a 3.8 database schema to the version 7.0 or later database schema. For additional details, see [Migration Script Reference](#).

After you run bfmigrate:

- Log on to the Management Console and check your project information to verify that the migration was successful.
- Optionally, delete your 3.8 database schema by running the bfmigrate -d command.

Important

If you are upgrading from Build Forge 3.8 with the included MySQL 3.2.3 database, before you run bfmigrate you must install MySQL 5.0. Migration to MySQL 5.0 must include setting up a user for Build Forge to use to connect to the MySQL 5.0 instance.

1. Make sure the database is running.
2. Run the bfmigrate script.

From the Build Forge installation directory, run the following command.

```
bfmigrate -m
```

3. Wait for successful completion

The migration script can take an hour or more to run. A successful run will end with the following message:

```
Migration: 4156: Successfully migrated v3.8 data to v7.x
```

If migrate fails, undo the migration, then try again. To undo:

```
bfmigrate -u
```

Run the bfdbmigrate Script

Stop: Use these procedures only to upgrade for the following case.

- **From:** Build Forge version 3.8 using the included MySQL database
- **To:** Build Forge version 7.0 or later using the included DB2 Express database

Do not complete this task if you set up a non-provided database of any type with Build Forge version 3.8.

Run the `bfdbmigrate` script to migrate MySQL data (Build Forge 3.8) to DB2 Express data (Build Forge 7.0 or later).

1. Make sure the database is running.

The `mysqld` process should be running.

2. Run the `bfdbmigrate` script.

From the Build Forge installation directory, run the following command:

```
bfdbmigrate
```

3. Wait for successful completion

The script creates a schema in the DB2 Express database. The script generates a series of messages as it runs. A successful run will end with the following message:

```
Migration: 5148: Successfully migrated from old db to new
```

A successful migration does not generate any error messages.

4. Stop the `mysqld` process.

Use Windows Task Manager, **Processes** tab. Check for the **mysqld** process. If it is running, select it and click **End Process**.

5. Stop and then restart the Build Forge engine.

6. Change the license server name to the Rational License Server name.

The migration placed your old license key in the new database.

- a. Select **Administration > System**. A list of system settings is shown.
- b. Find the License Server system setting.
- c. Click the License Server name to edit its value. Enter the host name of the Rational License server where your Build Forge licenses are installed.
- d. Click **Save**.

7. **Post-Install Steps**

Perform the post-installation steps in [Post-Installation Setup](#).

Upgrading the Management Console on UNIX or Linux

This procedure describes how to upgrade prior versions of the Management Console on UNIX/Linux to the latest version.

Before proceeding, check the following list:

- *Version:* Determine whether your current installation is version 3.8 or 7.0 or later (with any iFix).
- *Changing platforms:* If you are changing to a new machine and/or operating system along with the upgrade, treat the upgrade as a new installation, then export data from your old system and import it to the new one.
- *Preserving data:* Do not uninstall your existing version of Build Forge. It must be upgraded in place. Uninstalling the existing installation destroys the project and log data in your database.
- *Preserving custom interfaces/adaptor templates:* If you have custom interfaces (now known as adaptor templates), rename the `<bf-install>/interface` directory before performing the upgrade. Your custom work can be easily applied to the new adaptor templates.
- *Required Disk Space:* If you are migrating from Build Forge version 3.8 with MySQL 5.0, to Build Forge version 7.0 or later with DB2 Express, allow at least triple the current database size plus 2 GB.
- *Database:* The database server must be running in order to complete the upgrade.
- *Notification Templates:* In version 7.0.1, the URL in the notification templates changed. If you are upgrading from 7.0, your notification templates might not work. If you experience an error, see [Troubleshooting Common Problems](#).

Upgrading from Version 7.0 or Later on UNIX or Linux

These procedures apply for any version 7.0 or later installation, including those with an iFix applied. The iFix must have been correctly installed.

Before you start:

- (7.0.1 and later) Stop the Tomcat server. Execute the shutdown command in the Tomcat bin directory:

```
cd $CATALINA_HOME/bin
./shutdown.sh
```

- Stop the Build Forge engine.

1. Download the Build Forge tarball (for example, `mc-<version>-unix.tar.gz`) from IBM or copy it from installation media to a temporary directory on your current Build Forge host.

2. As a non-root user, backup your current Build Forge directory:

```
mv buildforge bf-old
```

3. As a non-root user, create an installation directory:

```
mkdir /usr/local/buildforge
```

4. As a non-root user, change to the installation directory:

```
cd /usr/local/buildforge
```

5. Copy the downloaded file from tmp to the current installation directory:

```
cp /tmp/mc-<version>-unix.tar.gz .
```

6. Unzip and extract the file contents to the installation directory:

```
$ gunzip -c mc-<version>-unix.tar.gz | tar xvf -
```

Note

Note that tar is set to use *stdin* for the pipe (using the - parameter)

7. Run the `bfinstall.pl` Perl script with the `--with-dbtype` parameter for your database type.

Values are:

- DB2: `---with-dbtype=db2`
- MySQL: `--with-dbtype=mysql`
- Oracle: `--with-dbtype=oracle`

The following example installs for DB2:

```
$ ./bfinstall.pl --with-dbtype=db2
```

Note

The script asks you to confirm the installation of PHP and Perl you are using, then validates the versions and installed components of PHP and Perl. You may see messages passed through from your compiler as well as output from the script itself.

8. Restart the engine.
9. Set permissions for Apache. See [Set Permissions for Apache](#).
10. Copy the war files for the reporting application and for services-layer APIs for Java and Perl.

See [Configure Tomcat for Build Forge](#)

11. Download and unzip the JDBC driver for your database. Copy the jar file only to `$CATALINA_HOME/common/lib`. See [Install JDBC Drivers](#).
12. (7.0.1 and later) Stop the Tomcat server:

```
cd $CATALINA_HOME/bin
./catalina.sh stop
```

13. (7.0.1 and later) Delete the `quickReport` and `rbf-services` directories from the Tomcat webapps directory:

```
cd $CATALINA_HOME/webapps
rm -r quickReport
```

```
rm -r rbf-services
```

14. (7.0.1 and later) Restart the Tomcat server:

```
cd $CATALINA_HOME/bin  
  
./catalina.sh start
```

Upgrading from Version 3.8 on UNIX or Linux

To upgrade from Build Forge version 3.8 to version 7.0 or later, you must perform the tasks described in this topic.

- Run `bfinstall.pl` to upgrade
- Run the `bfmigrate` script to update your database.
- Complete post-upgrade tasks.

Important

If you are running Build Forge version 3.8 on MySQL, you must install MySQL 5.0 and migrate Build Forge data to MySQL 5.0 before running `bfmigrate`. (Migration to MySQL 5.0 must include setting up a user for Build Forge to use to connect to the MySQL 5.0 instance.)

Run `bfinstall.pl` to Upgrade Build Forge

1. Stop the Build Forge engine.
2. Download the Build Forge tarball (for example, `mc-<version>-unix.tar.gz`) from IBM or copy it from installation media to a temporary directory on your current Build Forge host.
3. As a non-root user, backup your current Build Forge directory:

```
mv buildforge bf-old
```

4. As a non-root user, create an installation directory:

```
mkdir /usr/local/buildforge
```

5. As a non-root user, change to the installation directory:

```
cd /usr/local/buildforge
```

6. Copy the downloaded file from `tmp` to the current installation directory:

```
cp /tmp/mc-<version>-unix.tar.gz .
```

7. Unzip and extract the file contents to the installation directory:

```
$ gunzip -c mc-<version>-unix.tar.gz | tar xvf -
```

Note

Note that tar is set to use *stdin* for the pipe (using the - parameter)

8. Run the `bfinstall.pl` Perl script with the `--with-dbtype` parameter for your database type.

Values are:

- DB2: `---with-dbtype=db2`
- MySQL: `--with-dbtype=mysql`
- Oracle: `--with-dbtype=oracle`

The following example installs for DB2:

```
$ ./bfinstall.pl --with-dbtype=db2
```

Note

The script asks you to confirm the installation of PHP and Perl you are using, then validates the versions and installed components of PHP and Perl. You may see messages passed through from your compiler as well as output from the script itself.

9. Restart the engine.

Run the bfmigrate Script

These procedures apply for any version 7.0 or later installation, including those with an iFix applied. The iFix must have been correctly installed.

Run the `bfmigrate` Perl script to update the Build Forge data in your database.

1. Run the `bfmigrate` script.

From the Build Forge installation directory, run the following command:

```
bfmigrate.pl -m
```

2. Wait for successful completion

The migration script can take an hour or more to run. A successful run will end with the following message:

```
Migration: 4156: Successfully migrated v3.8 data to v7.x
```

If migrate fails, undo the migration, then try again. To undo:

```
bfmigrate -u
```

Complete Post-Upgrade Tasks

1. Set permissions for Apache. See [Set Permissions for Apache](#).

2. Copy the war files for the reporting application and the services-layer APIs for Java and Perl.
See [Configure Tomcat for Build Forge](#).
3. Download and unzip the JDBC driver for your database. Copy the jar file only to \$CATALINA_HOME/common/lib. See Install JDBC Drivers.

Upgrading Agents

Upgrade agents by installing new agents over the old ones.

You upgrade an agent by installing a new agent on top of the old one. Follow the installation instructions for installing an agent.

- Version 7.0 and version 3.8 agents must be upgraded. Install version 7.0.1 or later agents over them.
- Version 7.0.1 or later agents can be made self-updating.

Migration Script Reference

The migration scripts convert version 3.8 data to the new schema and create new data objects.

Overview

Two migration scripts are provided to upgrade from a 3.8 installation:

- `bfmigrate` performs initial conversion for any 3.8 installation. It has several command-line options, at least one of which must be used. The file is `bfmigrate.exe` for Windows, `bfmigrate.pl` for UNIX or Linux.
- `bfdbmigrate` performs conversions on a MySQL database (provided with Build Forge version 3.8) to a DB Express database (provided with Build Forge 7.0 and later). It has no options.

The migration scripts are in the Build Forge installation directory. Execute them from a command line.

bfmigrate Reference

Syntax:

```
bfmigrate -[h|v|m|d|u|r]
```

This command converts a 3.8 database to the new schema.

Note

Your database must be running before you run the `bfmigrate` command. If your 3.8 installation used MySQL on Windows[®], for example, you can issue the following commands to start MySQL (and if MySQL is already running, the system ignores the command).

```
cd "c:\Program Files\BuildForge\Apache\mysql\bin"
mysqld --standalone
```

Options

- h Displays help text.
- v Displays the version number.
- m Migrates a 3.8 database to version 7.0 or later schema. The system renames existing tables (`bf_users` becomes `b3_users`, and so on), then creates a new set of `bf_` tables. The system converts all data in the 3.8 tables to version 7.0 or later formats, as well as creating some new data objects. The migration produces a stream of status messages as it converts tables; some tables may take 30 minutes or more to convert, depending on the size of your database. When the migration is complete, the script displays the following message:

```
Migration: 4156: Successfully migrated v3.8 data to v7.0
```

Note

After a migration, you must update the **License Server** system configuration setting, which will be set to your old 3.8 license key. Set it to the machine name for your license server.

- d Deletes tables saved from a 3.8 installation, by deleting all tables in the database whose names start with `b3_`.
- u Undo a migration by dropping version 7.0 or later tables. You are left with a database with no 7.0 or later tables; if you preserved your 3.8 tables, they remain, but are not used (because they have been renamed to `b3_` names). The engine automatically creates new, blank 7.0 or later tables when it is next started, so the result is an empty version 7.0 or later database.
- r Roll back a migration. This option drops the 7.0 or later tables, then renames the `b3_` tables to `bf_` names. After a rollback, you can retry an upgrade from version 3.8 to version 7.0 or later.

bfdbmigrate Reference

Windows only. Use this command only after performing a `bfmigrate` conversion, if and only if your 3.8 installation used the MySQL database installed by the system (if you installed your own, separate MySQL database, do not use this script). To run it, issue the following command from your installation directory:

```
bfdbmigrate
```

The script creates a schema in the DB2 Express database. As with the `bfmigrate` script, you should wait until the script completes. The script generates a series of messages as it converts various tables, ending with the following message, and a successful migration does not generate any error messages.

Migration: 5148: Successfully migrated from old db to new

Data Created by bfmigrate

In addition to migrating your existing data objects, the script derives some new ones from your database. You can modify these data objects after migration. The script:

- Creates collectors for each of your existing servers, and names them after your servers. Any servers that are members of a pool get a `_POOL` property in their collectors, with the appropriate pool name as the `_POOL` value.
- Creates selectors for each of your existing servers, naming them after your servers.
 - If the server had a `_POOL` property in 3.8, its selector includes a required property (`_POOL == <poolname>`) and an optional property (`BF_NAME == <servername>`). Such a selector can select any server in the pool, but tends to prefer the named server.
 - Otherwise, the selector includes a single required property `BF_NAME == <servername>`, which specifies that the selector can only choose a server with the specified name.
- Creates a server authentication for each unique username/password combination used with your 3.8 servers. Because the username/password for accessing a server are a separate data object in version 7.0 and later versions, the script separates these from the server object and lists them appropriately in version 7.0 and later database. The newly created server authentications have the same names as the server records they are derived from. If you used the same username/password combination for more than one server, the system makes only one server authentication, naming it after the first server that used the combination.

Upgrading APIs

If you are using Java client API or a Perl client API scripts, you must also upgrade your client APIs to the current version.

For instruction how to download and set up your current API packages, See "Working With APIs" in the Build Forge online Help.

Troubleshooting Common Problems

Upgrading Agent on Solaris requires running `pkgrm` command

Use the `pkgrm BFAgent` command to remove the existing Solaris Build Forge agent before running the `pkgadd` command.

URL for version 7.0 notification templates might not work in later versions

The notification template URL provides a link in the notification e-mail to the build report in Build Forge.

In version 7.0.1, the URL in the notification templates changed. Consequently, when you upgrade from 7.0 to any later version, the notification template URL links might not work.

If you experience a link error, make the following changes to the notification templates:

1. Select Project > Templates.
2. Click on the notification template to display its properties on the Details tab.
3. In the Body field, locate the URL for the template. The URL should be similar to the one in the following example:

```
http://${CONSOLEHOST}:${CONSOLEPORT}/fullcontrol/index.php?mod=projectruns&action=edit&bfid=${PID}&bfid=${BID}&bfid=${UID}
```

4. Replace the following 7.0 URL elements with the appropriate 7.0.1 (and later) URL elements:

7.0 URL elements	7.0.1 (and later) URL elements
projectruns	jobs
&	&

Administration

This section describes administrative operations for the Build Forge[®] system.

The following topics are discussed:

- Managing the engine
- Managing the database

Managing the Engine

This section describes how to pause, start, or stop the engine (`bfengine.exe` / `bfengine.pl`).

Note

If you have any problems getting the engine to run, run it in the foreground so that you can see the status and error messages it generates. On Windows, select **Start** → **All Programs** → **IBM Rational Build Forge** → **Start Engine (Foreground)** to run the engine in the foreground, which causes it to display output in a console window. When the engine is running in the foreground, you can stop it by closing its console window.

Pausing the Engine

If you want to act directly on your Build Forge[®] database, to back it up, restore it, or change the schema in some way, you should pause the engine before you make any changes.

To pause the engine, select **Administration** → **System** and locate the **Pause Build Forge Engine** property (use the **Filter** box). The system displays a form for editing the **Pause Build Forge Engine** setting. Change the property to Y and click the **Save Config** button.

The system pauses the engine, but any currently-running projects continue. You must wait for any currently active jobs to complete to ensure that no data is being written to the database.

To reactivate the engine, change the value of the property back to N, and click the **Save Config** button again.

Note

When the engine is paused, your database can still be updated from the console and the services layer. To prevent this, you must also shut down all Build Forge services, including Apache and Tomcat. For instance, before performing a database backup, you must pause the engine and shut down all Build Forge services.

Starting and Stopping the Engine

You may want to stop the engine when you perform a backup of its database or when you install an updated version of the system. The following sections describe how to start and stop the engine on Windows[®], Linux[®] or Solaris.

Starting and Stopping the Engine on Windows

On Windows:

- At **Start** → **Programs** → **IBM Rational Build Forge Management Console** , choose either
 - **Start Engine Service**
 - **Stop Engine Service**

Control Panel: You can also use the **Administrative Tools** > **Services** control panel to start or stop the **IBM Rational Build Forge Management Console** service.

Running in the Foreground: If you have any problems getting the engine to run, run it in the foreground so that you can see the status and error messages it generates: **Start** → **Programs** → **IBM Rational Build Forge Management Console** → **Start Engine (Foreground)** . As the Management Console runs, log output is shown in a console window. To stop the engine in this mode, close its console window.

Starting and Stopping the Engine on Linux or Solaris Systems

If you have an rc file installed (typically in /etc/rc.d/init.d):

```
$ /<path to rc file>/buildforge start
```

```
$ /<path to rc file>/buildforge stop
```

If you do not have an rc file installed, start engine using the following:

```
$ /<path to system installation>/Platform/buildforge.pl &
```

Stop it by determining its process ID and then issuing a kill command:

```
$ ps aux | grep buildforge  
$ kill ${<PID>}
```

Managing the Database

This section describes issues that are important when setting up the Management Console database, especially if you want to change default configurations.

Configuring an ODBC Connection on Windows

To connect to a database via ODBC on Windows[®], you must provide the appropriate connection information when you install the Management Console. Before you perform the installation, however, a connection to this database must be properly configured in **Start** → **Settings** → **Control Panel** → **Administrative Tools** → **Data Sources** as a System DSN.

Database Size

The size of the database depends on the number of projects you run and the amount of log data you keep in the Management Console.

Large customers who run more than 8,000 processes per month and have been running the Management Console for one year have databases approaching 8GB of storage.

Support

This section describes how to get support for the Build Forge[®] system.

The following topics are discussed:

- Contacting IBM support
- Executables list
- Notices

Executables List

The following table lists the various executable files that make up the IBM[®] Rational[®] Build Forge[®] system, and describes how they work together. These files have .exe extensions on Windows[®] and .pl extensions on UNIX or Linux[®]. On UNIX/Linux, the executable files are in the `<bf-install>/Platform` directory.

Note

You can display the version number for any executable file in the system by running the file with a -v option. For example:

```
C:\Program Files\IBM\Build Forge>bfproject -v
bfproject.exe 7.0.351
```

Executable	Service?	Description
bfdbmigrate.exe/bfdbmigrate.pl	N	Migration script used to convert a default database from version 3.8 (MySQL) to DB2 [®] after the schema has been converted to version 7.0. See your <i>Installation Guide</i> for instructions. This script is intended to be used <i>after</i> using <code>bfmigrate.exe/bfmigrate.pl</code> .
bfproject.exe/bfproject.pl	N	Launched by <code>buildforge.exe</code> to start a job.
bfengine.exe	Y	Starts <code>buildforge.exe</code> and web server. Windows only.
bfexport.exe/bfexport.pl	N	Utility used to export data from the database.
bfbomexport.exe/bfbomexport.pl	N	Utility used to export the BOM from the database.
bfimport.exe/bfimport.pl	N	Utility used to import project data into the database.
bfmigrate.exe/bfmigrate.pl	N	Migration script to upgrade a 3.8 database to version 7.0. See your <i>Installation Guide</i> for instructions.
bfpurge.exe/bfpurge.pl	N	Launched by <code>buildforge.exe</code> to purge builds.
bfrefresh.exe/bfrefresh.pl	N	Launched by the system to update the manifests for servers.
bfsched.exe/bfsched.pl	N	Launched by <code>buildforge.exe</code> to check database for scheduled jobs and launch them when appropriate.

Executable	Service?	Description
bfshutdown.exe/bfshutdown.pl	N	Utility application; launch it to shut down all Build Forge [®] processes.
bfstepcmd.exe/bfstepcmd.pl	N	Launched by bfproject for long-running steps, to create a separate process for them.
buildforge.exe/buildforge.pl	N	Manages build, purge, and sched processes.
console_uninst.exe	N	Uninstallation program for the Management Console. Windows only.
bfdispatch.exe/bfdispatch	Y	Agent service. Windows only.
bfagent.exe/bfagent	N	Agent executable

Contacting IBM Customer Support for Rational Products

If you have questions about installing, using, or maintaining this product, contact IBM Customer Support as follows:

The IBM software support Internet site provides you with self-help resources and electronic problem submission. The IBM Software Support Home page for Rational products can be found at <http://www.ibm.com/software/rational/support/>.

Voice Support is available to all current contract holders by dialing a telephone number in your country (where available). For specific country phone numbers, go to <http://www.ibm.com/planetwide/>.

Note

When you contact IBM Customer Support, please be prepared to supply the following information:

- Your name, company name, ICN number, telephone number, and e-mail address
- Your operating system, version number, and any service packs or patches you have applied
- Product name and release number
- Your PMR number (if you are following up on a previously reported problem)

Downloading the IBM Support Assistant

The IBM Support Assistant (ISA) is a locally installed serviceability workbench that makes it both easier and simpler to resolve software product problems. ISA is a free, stand-alone application that you download from IBM and install on any number of machines. It runs on AIX, (RedHat Enterprise Linux[®] AS), HP-UX, Solaris, and Windows[®] platforms.

ISA includes these features:

- Federated search
- Data collection
- Problem submission
- Education roadmaps

For more information about ISA, including instructions for downloading and installing ISA and product plug-ins, go to the ISA Software Support page.

IBM Support Assistant: <http://www.ibm.com/software/support/isa/>.

Notices for IBM Rational Build Forge

This information was developed for products and services offered in the U.S.A.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:

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

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

*IBM World Trade Asia Corporation
Licensing
2-31 Roppongi 3-chome, Minato-ku
Tokyo 106-0032, Japan*

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law:

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT,

MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Corporation
Department BCF
20 Maguire Road
Lexington, MA 02421
U.S.A.

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement or any equivalent agreement between us.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurements may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

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 and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

All statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies,

brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs.

Trademarks

IBM, the IBM logo, Rational, ClearCase, ClearQuest, and DB2 are registered trademarks of International Business Machines Corporation in the United States, other countries, or both.

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

Intel, Intel logo, Intel Inside, Intel Inside logo, Intel Centrino, Intel Centrino logo, Celeron, Intel Xeon, Intel SpeedStep, Itanium, and Pentium are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

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.

Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

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

For other IBM trademark attributions, see <http://www.ibm.com/copytrade.shtml> [<http://www.ibm.com/legal/copytrade.shtml>]