

# Server Installation Guide

---

**BigWorld Technology 2.1. Released 2012.**

**Software designed and built in Australia by BigWorld.**

**Level 2, Wentworth Park Grandstand, Wattle St  
Glebe NSW 2037, Australia  
[www.bigworldtech.com](http://www.bigworldtech.com)**

**Copyright © 1999-2012 BigWorld Pty Ltd. All rights reserved.**

**This document is proprietary commercial in confidence and access is restricted to authorised users. This document is protected by copyright laws of Australia, other countries and international treaties. Unauthorised use, reproduction or distribution of this document, or any portion of this document, may result in the imposition of civil and criminal penalties as provided by law.**

# Table of Contents

1. Overview .....	5
1.1. Requirements .....	5
1.1.1. Hardware Requirements .....	5
1.1.2. Software Requirements .....	5
1.2. Reading this document .....	5
2. Simple Installation .....	7
2.1. Recommended Setup .....	7
2.1.1. Supported Distributions .....	7
2.2. Preparation .....	7
2.2.1. Installing CentOS .....	8
2.2.2. Installing the EPEL Repository .....	8
2.2.3. Installing, Configuring and Starting MySQL .....	9
2.3. Installing the BigWorld Technology Server package .....	12
2.3.1. BWMachineD .....	12
2.3.2. BigWorld Server .....	12
2.3.3. BigWorld Server Tools .....	13
2.3.4. Customising your installation .....	14
3. Server First Run .....	15
3.1. Creating a Developer Account .....	15
3.1.1. Account creation example .....	15
3.1.2. Common Mistakes .....	16
3.2. Create a new project .....	16
3.3. Package specific key generation .....	17
3.3.1. LoginApp Keypair .....	17
3.3.2. Customer Server Key .....	18
3.4. Managing the BigWorld Server .....	18
3.4.1. Adding A New User .....	18
3.5. Login to WebConsole .....	19
3.6. Starting a BigWorld server .....	20
4. Cluster Configuration .....	23
4.1. Security .....	23
4.2. Routing .....	24
4.3. Buffer sizes .....	24
4.4. Disabling cron jobs .....	25
A. Hardware Requirements .....	27
A.1. CPU Specifications .....	27
A.2. Duals/Single/Quad/Blades .....	27
A.3. Network Interface Cards .....	28
A.4. Disk Storage .....	28
A.5. Power Supplies .....	28
A.6. Memory .....	28
A.7. NOC Bandwidth .....	29
A.8. VMWare .....	29
B. Installing CentOS 5 .....	31
B.1. Installing .....	31
B.2. Post-installation Setup .....	33
B.2.1. Install updates .....	33
B.2.2. Configure services .....	33
B.2.3. Install build tools .....	34
B.2.4. MySQL server and development libraries .....	34
B.2.5. Changing UIDs .....	34
C. Copying files to Linux .....	37
C.1. USB Thumb Drive / External Hard Disk .....	37
C.1.1. Installing NTFS Support .....	38

C.2. Python SimpleHTTPServer hosted on Windows .....	38
C.3. Windows network share .....	38
D. Creating a custom BigWorld server installation .....	39
D.1. Customising RPMs .....	39
D.2. Manual Installation .....	39
D.2.1. BWMachineD .....	39
D.2.2. Server .....	40
D.3. Server Tools .....	40
D.3.1. Requirements and Caveats .....	40
E. Understanding the BigWorld Machine Daemon (BWMachineD) .....	43
E.1. How BWMachineD Works .....	43
E.2. Configuring BWMachineD .....	43
E.2.1. Creating ~/.bwmachined.conf .....	43
E.2.2. Creating /etc/bwmachined.conf .....	45
F. Troubleshooting .....	49
F.1. Check That BWMachineD is Running .....	49
F.1.1. Troubleshooting BWMachineD .....	49
F.2. StatLogger .....	50
F.3. BigWorld Support .....	50

# Chapter 1. Overview

This document describes how to quickly install the BigWorld Server and Tools in order to see an operational environment up and running in the shortest possible time. This document assumes you have read the Server Overview and understand the basic interaction between BigWorld server processes.

As the majority of this installation procedure is automated, this mechanism of installation may not be suited for more advanced users who are experienced with BigWorld and wish to customise the installation process. For more advanced installation documentation, please refer to Appendix D, *Creating a custom BigWorld server installation* on page 39 .

## 1.1. Requirements

### 1.1.1. Hardware Requirements

The BigWorld server will run on most mainstream PC desktop hardware providing it is 64 bit (x86\_64). For a detailed description of hardware requirements and recommendations for the BigWorld server please refer to Appendix A, *Hardware Requirements* on page 27 .

The BigWorld server tools have a recommended minimum hardware configuration as follows:

- 1GHz CPU
- 1GB RAM
- 30GB Hard disk (for main OS install)
- 120GB Hard disk (for logging)
- 100Mbit NIC

If an external machine is hosting the MySQL server for use with WebConsole and StatLogger, we recommend that machine to have similar specifications. We also recommend that the network link between the machine hosting the WebConsole and/or StatLogger and the link to the MySQL server be low-latency and high capacity for best performance.

### 1.1.2. Software Requirements

CentOS 5 64 bit is the recommended platform for development and production environments, however both RedHat Enterprise Linux 5 and CentOS 5 are supported. For instructions on installing CentOS 5, please refer to Appendix B, *Installing CentOS 5* on page 31 .

Other software dependencies should automatically be met by the RPM dependency list when installing packages from an RPM

## 1.2. Reading this document

While reading this document there are some subtle conventions we use to distinguish between different situations. One of these conventions is the difference we use between commands that are intended to be run as the root user as opposed to a command to be run in a normal user account. This difference is significant in the same way as requiring Administrator access on a Windows machine.

Commands to be run as root will be prefixed with a "#" symbol, while commands to be run as a regular user will be prefixed with a "\$" symbol.

For example, if we wanted to run the command 'cat /proc/1/mem' as the root user we would express it as:

```
# cat /proc/1/mem
```

Whereas if it were to be run as a regular user it would look like:

```
$ cat /proc/1/mem
```

Where possible we try to be explicit with regard to how a command should be run, however this visual indication should assist you if there is any confusion.

IBM provides a useful Linux tutorial, which explains the root user and details the methods of switching between accounts. If you are new to Linux, you should read this tutorial before continuing. It is available at <http://www.ibm.com/developerworks/linux/tutorials/l-basics/section5.html>.

Another convention we use is to italicise parts of commands that you should substitute for the value relevant to your situation or needs.

For example, if we wanted to use the command 'useradd', which is followed by a new username, we would express it as:

```
$ useradd username
```

This can help to prevent ambiguity in some complex or unfamiliar commands that are used in this document.

## Chapter 2. Simple Installation

Installing the BigWorld server consists of installing 3 separate components:

- BigWorld Machine Daemon (BWMachineD)
- BigWorld Server (BaseApp, CellApp, DbMgr, etc)
- BigWorld Server Tools (MessageLogger, WebConsole, etc)

BWMachineD is required by both the BigWorld server and the BigWorld server tools, however both the server and server tools can run independantly of one another and are commonly installed on separate machines to avoid load issues of one service interfering with another.

The simplest path to installing the server and tools is to use the pre-generated RPM packages which are tested and known to work on the supported Linux distributions.

### 2.1. Recommended Setup

The recommended system configuration for a scalable BigWorld server environment is to separate the BigWorld server from the machine that will run the BigWorld server tools.

The isolation of the server tools on a separate machine is recommended in order to ensure that if high load situations occur on any cluster machines, the associated increase in logging and monitoring tasks performed by the server tools do not further degrade performance on any of the active cluster machines.

Due to the potential for log files to grow quite drastically in both a development and a production environment it is recommended that, when creating partitions for the operating system installation, to create a separate partition or have an entirely separate hard disk dedicated for the BigWorld server logs.

For the purposes of an initial installation or small scale development it is perfectly acceptable to install both the server and tools onto the same machine, and this installation guide will assume that the installation is being performed on a single machine.

#### 2.1.1. Supported Distributions

BigWorld only officially supports the 64-bit CentOS 5.x and RedHat Enterprise Linux 5.x distributions.

BigWorld has previously supported other Linux distributions, however in order to better support customers we have consolidated our support list to have a greater focus and reduce variability when resolving customer issues.

CentOS can be downloaded from the CentOS community portal.

RedHat Enterprise Linux is available from the RedHat website.

### 2.2. Preparation

As the BigWorld server is a complex set of programs with a number of dependencies, it is important to ensure that all your system has been installed correctly and prepared before continuing with the primary installation. A quick overview of what is required to install the BigWorld server and tools follows:

- 64 bit x86\_64 hardware
- CentOS<sup>1</sup> 5 64 bit

---

<sup>1</sup> Anywhere CentOS is mentioned, Red Hat Enterprise Linux may be substituted.

- EPEL repository enabled
- MySQL installed, configured and started

The following sections will outline in more detail how to achieve a functional installation of CentOS ready for an installation of the BigWorld server.

## 2.2.1. Installing CentOS

---

A guide on how to install Linux is out of the scope of this documentation as this process will commonly need to be tailored to your circumstances. We have however provided a rough overview of the CentOS installation process which outlines specific packages and configuration options that may be useful for a BigWorld server cluster machine. This overview can be found in Appendix B, *Installing CentOS 5* on page 31 .

### 2.2.1.1. Update System Packages

After any system installation it is a good idea to update all the installed system packages as there may have been important security fixes or other bug fixes which can impact the performance and security of your system since the installation media used for your install was produced.

Run the following command as root:

```
# yum update
```

After performing this update it is a good idea to reboot your computer to pick up any system changes such as kernel updates.

## 2.2.2. Installing the EPEL Repository

---

In order to fully support the server tools under CentOS it is necessary to install some packages that are not available in the default CentOS installation. In order to facilitate this you must enable the Extra Packages for Enterprise Linux (EPEL) repository which is provided by the Fedora Project. The Fedora project manages this repository as they are the base distribution from which Red Hat and CentOS are forked from.

To enable the EPEL repository use the following command:

### Note

The EPEL packages aren't kept as up to date as the official releases. If you have trouble downloading a version, try navigating to the directory and searching for the current version of the EPEL package. For example as of the CentOS 5.6 release the EPEL package still refers to 5.4.

Please modify the following URL as required.

As root, run:

```
# wget http://download.fedoraproject.org/pub/epel/5/x86_64/epel-release-5-4.noarch.rpm
# rpm -Uvh epel-release-5-4.noarch.rpm
```

This step has now made the packages from the EPEL repository available to your CentOS installation which will be used when the BigWorld server RPMs are installed.

### 2.2.3. Installing, Configuring and Starting MySQL

Both the BigWorld server and server tools can make use of MySQL for persistent data storage. In order to take advantage of MySQL it must be properly configured for each use case.

To install the MySQL server, run the following command as root:

```
# yum install mysql-server
```

This will install the MySQL server only. In order to interact with the MySQL server to create initial databases and set user access privileges we need to install the MySQL client. To do this run the following command as root:

```
# yum install mysql
```

DBMgr requires that its tables use the InnoDB storage engine, which is not necessarily the default engine used by MySQL. To make InnoDB the default engine, edit the file `/etc/my.cnf`, and add the following entry in the `[mysqld]` section:

```
default-storage-engine=InnoDB
```

#### 2.2.3.1. Start the MySQL server

After installing the MySQL server it is quite common to not have the actual MySQL server running or configured to restart after a reboot. As it is a core component in most environments it is important to check the MySQL server is setup correctly before proceeding.

As root run:

```
# /sbin/chkconfig --levels 345 mysqld on  
# /etc/init.d/mysqld start
```

This has ensured that the MySQL server will be running after the machine is rebooted, and starts up the MySQL server for immediate use. If you are new to Linux and are unfamiliar with the concept of *runlevels* it would be useful to gain a basic understanding of this concept if you intend to maintain a BigWorld server cluster. More information can be found on the Red Hat Documentation website.

#### 2.2.3.2. MySQL Accounts

Both the BigWorld server and the BigWorld server tools have different requirements of the MySQL server as they are both performing unique tasks. As a result of this, we recommend that a separate account is created for the BigWorld server tools, and for every user who will be running a BigWorld server.

The default MySQL installation is configured with one user called `root`. This is not the same as the system root user. In order to create a new MySQL user we log in to MySQL using the MySQL `root` user account as follows:

```
$ mysql -u root
```

This command can be run as any user providing the MySQL root user account has no password set.

### Note

For detailed instructions on MySQL account creation and management please refer to the MySQL documentation website, specifically the *MySQL Server Administration* chapter, section *MySQL User Account Management*.

#### 2.2.3.2.1. BigWorld Server Tools

The BigWorld Server Tools component StatLogger requires a MySQL server to work, while WebConsole by default will use SQLite but can be configured to use MySQL.

The following example illustrates how to create a MySQL user for the server tools with a username of `bwtools` and a password of `bwtools_passwd` and assign the required privileges to allow the server tools to operate normally.

To create a user and grant privileges, run the following command:

```
$ mysql -u root
mysql> GRANT ALL PRIVILEGES ON *.* TO 'bwtools'@'localhost' IDENTIFIED BY
'bwtools_passwd';
mysql> GRANT ALL PRIVILEGES ON *.* TO 'bwtools'@'%' IDENTIFIED BY
'bwtools_passwd';
```

#### 2.2.3.2.2. BigWorld Server User Account

Every user in the network that will run their own BigWorld server will need to have a MySQL account created for that server instance. For example in a development environment with two server developers, Alice and Bob, and a QA team that runs a single server, three MySQL databases would need to be created and assigned to each server user.

It is considered good practice to limit the privileges of a database user to the tasks they will be required to perform. As the DBMgr process (and related tools) require less privileges than the BigWorld server tools, when we create an account for the server we will be more specific in regards to the privileges that user has.

We will need three pieces of information when creating a database for use with the BigWorld server.

- Username (set in `bw.xml` using `<dbMgr><username>`)
- Password (set in `bw.xml` using `<dbMgr><password>`)
- Database name (set in `bw.xml` using `<dbMgr><databaseName>`)

If you are not yet familiar with the `bw.xml` file, don't worry as this will be covered in more detail later in this document.

The following SQL commands provide the basic syntax to use when creating a MySQL account for use with the BigWorld server. Specific examples follow.

```
$ mysql -u root 1
GRANT SELECT, INSERT, UPDATE, DELETE, ALTER, CREATE, DROP, INDEX
ON game_db_name.* TO 'username'@'localhost' IDENTIFIED BY 'password'; 2

GRANT SELECT, INSERT, UPDATE, DELETE, ALTER, CREATE, DROP, INDEX
ON game_db_name.* TO 'username'@'%' IDENTIFIED BY 'password'; 3

GRANT RELOAD ON *.* TO 'username'@'localhost' IDENTIFIED BY 'password'; 4
```

```
GRANT RELOAD ON *.* TO 'username'@'%' IDENTIFIED BY 'password'; 5
```

- 1 Connect to the MySQL server using the MySQL root account. By default no password is required to use this account.
- 2 Grant access for the database table operations SELECT, INSERT, UPDATE, DELETE, ALTER, CREATE and INDEX to the database named *game\_db\_name* for the user *username* using the password *password* when connecting to the MySQL server from the local machine (ie: *localhost*).
- 3 This is the same as the previous GRANT command, however this rule is to give access for the user when connecting to the MySQL server from a remote machine (ie: not connecting from *localhost*). It is possible to restrict the machines from which the user can connect by specifying a more restrictive pattern after the @ symbol.
- 4 Grant access for the RELOAD capability on all databases from the local machine.
- 5 This is the same as the previous GRANT, however as before this rule grants the privilege when connecting from a remote machine.

### 2.2.3.2.2.1. Use Case

In our use case office there are two server developers, Alice and Bob. Alice is involved in the development on a new game project, Parrot Attack and an existing game project Chickens Fight Back. Bob on the other hand is only involved in the development of the new Parrot Attack game.

Alice will need two databases created for her, one for each project she is working on, and Bob will only need a single database for his own development purposes. The following table outlines the information that will be used by the database administrator in order to create the required MySQL database accounts.

Database Name	Username	Password
alice_parrot_attack	alice	1234567
alice_chickens_fight_back	alice	1234567
bob_parrot_attack	bob	bobs secret password1

You can see from this table that both Alice and Bob have been assigned their own database for each game project they are working on. This is to ensure that when Alice starts her own BigWorld server, she doesn't impact any work that Bob may be currently involved in.

Using Bob's example above we would create an account for Bob giving him privileges on his own Parrot Attack database as follows:

```
$ mysql -u root
mysql> GRANT SELECT, INSERT, UPDATE, DELETE, ALTER, CREATE, DROP, INDEX
ON bob_parrot_attack.* TO 'bob'@'localhost' IDENTIFIED BY 'bobs secret
password1';
Query OK, 0 rows affected (0.08 sec)

mysql> GRANT SELECT, INSERT, UPDATE, DELETE, ALTER, CREATE, DROP, INDEX
ON bob_parrot_attack.* TO 'bob'@'%' IDENTIFIED BY 'bobs secret password1';
Query OK, 0 rows affected (0.00 sec)

mysql> GRANT RELOAD ON *.* TO 'bob'@'localhost' IDENTIFIED BY 'bobs secret
password1';
Query OK, 0 rows affected (0.00 sec)

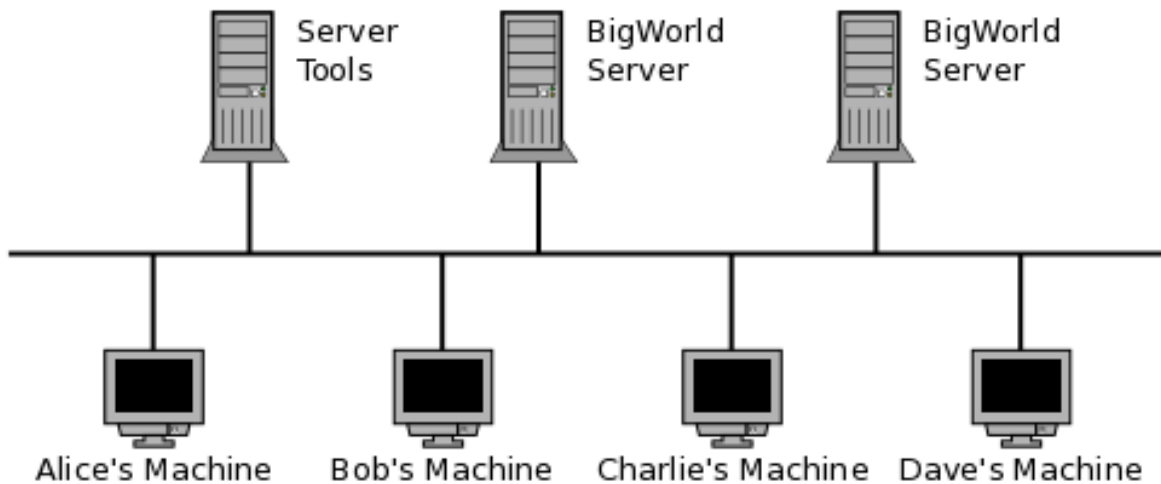
mysql> GRANT RELOAD ON *.* TO 'bob'@'%' IDENTIFIED BY 'bobs secret password1';
Query OK, 0 rows affected (0.00 sec)
```

## 2.3. Installing the BigWorld Technology Server package

Now your system should be ready to install the RPM packages contained in the `rpm` directory of your downloaded package. This guide will assume that you have copied the RPM files into your root user's home directory `/root`. If you are uncertain of how to transfer your files from a Windows machine to your newly installed Linux machine please refer to Appendix C, *Copying files to Linux* on page 37 .

These instructions assume that you are operating on a Linux command line, either via a text based login, or via a Console or Terminal opened through your Window Manager.

We will start by illustrating a basic office installation



### 2.3.1. BWMachined

The BigWorld Machine Daemon is required on all machines in a server cluster that will run BigWorld processes. As such this is the first program we will install.

```
# cd /root ❶
# yum install --nogpgcheck bigworld-bwmachined-2.1.0.x86_64.rpm ❷
```

- ❶ We first change to the directory to which the RPM files were copied. In this section we assume this is the `/root` directory (adjust this as necessary).
- ❷ Install the `bwmachined` package, making sure to replace the package name with the correct filename if the version numbers are different.

### 2.3.2. BigWorld Server

Installing the BigWorld server is trivial. As root, run:

```
# cd /root
# yum install --nogpgcheck bigworld-server-2.1.0.x86_64.rpm
```

Although the BigWorld server should now be successfully installed, there are still a two steps remaining before a BigWorld server instance can be run. The first step requires us to install the BigWorld server tools which provides you with the ability to start, stop, and interact with a BigWorld server instance. Installing the server tools is discussed in more detail in the following section “BigWorld Server Tools” on page 13 .

A user account also needs to be correctly configured so the server tools are able to locate the appropriate executables and game resources for when starting a BigWorld server instance. This is discussed in passing in “Create a new project” on page 16, and in more detail in Appendix E, *Understanding the BigWorld Machine Daemon (BWMachineD)* on page 43.

### 2.3.3. BigWorld Server Tools

You should now be able to successfully install the BigWorld server tools. Perform the following commands as root:

```
# cd /root
# yum install --nogpgcheck bigworld-tools-2.1.0.x86_64.rpm
```

With the server tools installed there are two more tasks left to do. Firstly we need to configure StatLogger to use our local MySQL server, and secondly we will test that the WebConsole has been started correctly.

#### 2.3.3.1. Configuring StatLogger

StatLogger currently only works with a MySQL database and as such requires a MySQL account to be configured in order to operate. When installing StatLogger from an RPM the preferences file that StatLogger uses will automatically be placed on your filesystem as `/etc/bigworld/stat_logger.xml`. To set your account details you will need to edit this file as the root user and change the `<dbUser>` and `<dbPass>` elements.

To illustrate how these values would be modified we will use the MySQL account details that were created in “BigWorld Server Tools” on page 10. Using this information we would change the `stat_logger.xml` file entries as follows:

```
<?xml version="1.0"?>
<preferences>
  <!--General options-->
  <options>
    <dbHost>localhost</dbHost>
    <dbUser>bwtools</dbUser>
    <dbPass>bwtools_passwd</dbPass>
    <dbPort>3306</dbPort>
```

After configuring StatLogger with your local database account information you will need to start StatLogger and restart WebConsole as follows:

```
# /etc/init.d/bw_stat_logger start
Starting bw_stat_logger:                [ OK ]
# /etc/init.d/bw_web_console restart
Stopping web_console:                   [ OK ]
Starting bw_web_console:                 [ OK ]
```

#### 2.3.3.2. Confirm tools are running

With the server tools installed on your tools machine it is worthwhile ensuring that they have started correctly so that you can be sure nothing has gone awry in the initial part of your installation. To do this we simply run the startup scripts with a status command to ensure they are working as expected.

As root run the following commands:

```
# /etc/init.d/bw_stat_logger status
```

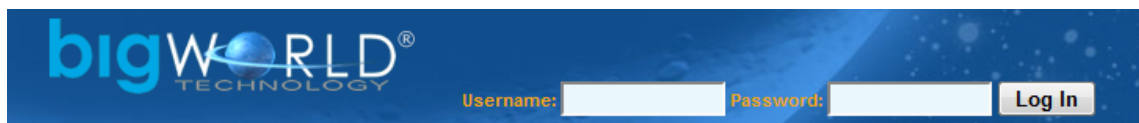
```
Status of stat_logger: running
# /etc/init.d/bw_message_logger status
Status of message_logger: running
# /etc/init.d/bw_web_console status
Status of web_console: running
```

### 2.3.3.3. Connecting to WebConsole

With the server tools installed you should now be able to see the WebConsole page by simply connecting a web browser to it. The URL of WebConsole is the hostname of the machine on which it has been installed at port 8080. For example:

```
http://localhost:8080
```

When connected you should be presented with a page similar to the following:



BigWorld WebConsole

Creating a WebConsole account will be discussed in more detail in *Server First Run* on page 15 .

### 2.3.4. Customising your installation

As development environments progress or a game starts to reach the release stage of its production cycle it may become necessary to customise your installation of BigWorld to your own environment. If this is the case, please refer to Appendix D, *Creating a custom BigWorld server installation* on page 39 for more information.

## Chapter 3. Server First Run

This *First Run* section assumes that the user is completely new to a BigWorld server setup and is working on a freshly installed machine. As such we will step through some basic setup steps such as creating a new user account for the developer to run a BigWorld server with. Feel free to step over any steps you already feel confident with.

The following procedures will also assume that the operations will be performed using a Linux console or terminal. While there are graphical alternatives available for users that have chosen a GUI installation, by describing the text based alternatives we are able to simplify the steps to the core behaviour being performed.

### 3.1. Creating a Developer Account

#### Note

If you are working in a large office environment you may wish to talk to your Systems Administrator about the user configuration mechanism that is already in place in the office. The approach described below may conflict with other distributed account management systems such as LDAP or NIS.

Each developer that will need to run their own server must have a Linux user account created for them. This is to enable them to have a location that stores their individual development files and the configuration files which determine how their server instance will be started.

The following command which is run as the root user is used to create a new user account in Linux.

```
# useradd username
```

Upon issuing this command a new user home directory will be created by default as `/home/username`.

With the user directory created we now need to set a password for the user to ensure they are the only one who is allowed to login. To do this issue the following command as root:

```
# passwd username
Changing password for user username.
New UNIX password:
Retype new UNIX password:
passwd: all authentication tokens updated successfully.
```

With these two steps completed you now have a new user account that you should be able to login with and continue the *First Run* steps.

#### 3.1.1. Account creation example

To fully illustrate the steps described above we will show the entire process of creating a new user account for a new server developer called Alice.

```
# useradd alice
# passwd alice
Changing password for user alice.
New UNIX password:
Retype new UNIX password:
passwd: all authentication tokens updated successfully.
```

### 3.1.2. Common Mistakes

While we haven't discussed multiple computer installations in much detail up to this point, one of the most common mistakes with user accounts can occur when a user account is created on different machines in the network with different numerical user IDs (UID). For example, suppose we had two server machines `host-A` and `host-B` and we performed the following operations:

```
# On host-A
useradd alice
useradd bob

# On host-B
useradd bob
useradd alice
```

Notice that the sequence of operations has been reversed on `host-B` compared with what was performed on `host-A`. As a result the UID that is created for the user `bob` on `host-B` may be the same as user `alice` on `host-A`. If the UIDs conflict for two users, there will be issues with server management and server monitoring.

There are a number of alternatives you can use to resolve this potential issue which will depend on how many machines you will need to use in your network. The most simple approach is to specify a UID when you create the accounts, such as:

```
$ useradd -u 6001 username
```

In this example the UID of 6001 was specified as the user was created, which allows us to specify the same UID when we create the account on another machine. This approach however becomes unfeasible when dealing with a large number of machines, in which case we may look to use a centralised account management solution such as LDAP or NIS. These approaches are more complex and are out of the scope of this document. There are numerous online resources that will guide you through the process of setting up these account management solutions.

## 3.2. Create a new project

With your user account created you can now login and start your first project which will enable you to start a server. Once you have a terminal logged in to your newly setup machine as the user created in “Creating a Developer Account” on page 15 creating an initial example project is quite simple. As part of the server installation a program called `bw_configure` was installed which is used to assist in configuring your user account to run a BigWorld server. By running this script with a new project name it will both create a new project as well as set our configuration file that is used to start a server.

To create a new project run the following command, replacing the string `project_name` with the name of your own project:

```
$ bw_configure project_name
```

For example if the server developer Alice were to create a new project called 'bigworld\_first\_run' she would see the following output:

```
$ bw_configure bigworld_first_run
'bigworld_first_run' project directory not found.
Create 'bigworld_first_run' with tutorial resources [y/N]? y
Creating new project at /home/alice/bigworld_first_run
Generating for chapter 6 - BASIC_NPC
```

```

from /opt/bigworld/2.1/server/bin/res
to /home/alice/bigworld_first_run/res

Writing /home/alice/bigworld_first_run/run.bat
Writing to /home/alice/.bwmachined.conf succeeded

Installation root : /opt/bigworld/current/server
BigWorld resources: /opt/bigworld/current/server/res
Game resources    : bigworld_first_run

```

This command will create a directory that is populated with the resources from the Tutorial. It will also create a new configuration file in your home directory called `.bwmachined.conf`. As a starting point it is useful to understand the basic breakdown of the `.bwmachined.conf` file in case you need to modify it at a later stage.

The `.bwmachined.conf` file is a file that contains a single line that is used by the `bwmachined` process on each machine to determine how to start server processes for a user. The information contained inside this file then has to point to the server binaries to be used as well as the game resources. The breakdown of the file is as follows:

```
server_binary_directory:game_res_directory[;secondary_res_directory]
```

In the `.bwmachined.conf` file that was created for you by the **bw\_configure** program the three paths mentioned above will be set as follows:

- `server_binary_directory` is automatically filled in with the installation path of the BigWorld server binaries.
- `game_res_directory` will most likely be filled out with a directory path that resembles `/home/username/project_name/res`. This `project_name` directory will be where all your additions and modifications will occur: it is where your game will be implemented.
- `secondary_res_directory` is automatically filled in with the BigWorld resource directory. The BigWorld resource directory is generally required for all games as it contains default configuration files and Python libraries that are used by the server processes.

For more details on the `.bwmachined.conf` file, please refer to Appendix E, *Understanding the BigWorld Machine Daemon (BWMachineD)* on page 43.

### 3.3. Package specific key generation

Both the Indie and non-Indie BigWorld server packages require some key files to be generated and placed in the correct location in order to work correctly. Please refer to the section below that is relevant to your package.

#### 3.3.1. LoginApp Keypair

In order to ensure that a Client is connecting to an authentic game server and the username / password sent by the Client are not able to be intercepted from a public network, the initial communication with the BigWorld server to the LoginApp is encrypted using a public keypair.

The BigWorld Technology package ships with a default LoginApp keypair, however since all customers receive the same keypair it is strongly recommended to create your own keypair from the start of your project and store it in your game resource directory rather than in the BigWorld resource directory.

For information on how to go about creating your own custom game keypair please see the Server Programming Guide section “Generating your own RSA keypair”. Once you have created your keypair, place it in to your game resource directory that was created in “Create a new project” on page 16. For example the `loginapp.privkey` would be placed in `/home/username/project_name/res/server` while the `loginapp.pubkey` would be placed in the directory `/home/username/project_name/res`.

### 3.3.2. Customer Server Key

Customers must place the `bigworld.key` file that was provided in the initial package notification email into their game resource tree. This file should be placed either in `bigworld/res/server` or `project_res/server`.

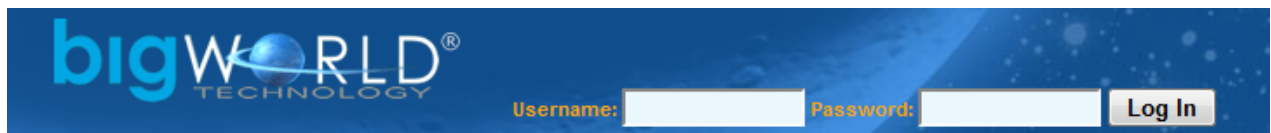
## 3.4. Managing the BigWorld Server

With a project created and with all the relevant configuration files set, we can look to start our server. In order to interact with a BigWorld server cluster we provide both a web interface and a set of command line tools. For most server interacts we recommend using the web interface and we will describe this approach here.

The web interface to the BigWorld server is called WebConsole. It provides a number of features for interacting with a BigWorld server cluster that are more fully outlined in the Server Operations Guide's chapter *Cluster Administration Tools*.

WebConsole is automatically started and runs on port 8080 of the machine on which the `bigworld-tools` RPM was installed. To connect to the WebConsole we would then use a URL such as `http://hostname:8080/` replacing `hostname` with either the IP address of the machine or its hostname depending on how the other machines in your network have been configured. To avoid confusion when testing WebConsole for the first time we recommend using the IP address of the WebConsole machine to confirm that any issues aren't related to hostname or DNS resolution problems.

When connecting for the first time you should be presented with a screen similar to the following:



### BigWorld WebConsole

The user accounts on the WebConsole now need to be created and the default administration login password changed. To do this, log in with username `admin`, password `admin`.

### 3.4.1. Adding A New User

To add a new user, click on the "Add New User" menu item on the left hand side of the page. You will then be presented with a form to input as follows:

Add A New User	
<b>Username</b>	<input type="text"/>
<b>Password</b>	<input type="password"/>
<b>Confirm Password</b>	<input type="password"/>
<b>Server User</b>	<input type="text"/>
<input type="button" value="Add User"/>	

The table below summarises the input fields providing sample input:

Field	Description
Username	The username that for the new user.
Password	The password with which the new user will log in.
Confirm Password	The password from the previous field, retyped to ensure no mistake was made on entry.
Server User	The Linux user this account will be associated with. This will be the Linux user that the BigWorld server processes will be run as.  If you are uncertain of this field, please talk to your system administrator to find out what your Linux user account is.

The following image shows a new user account AliceB being created and associated with the Linux user account alice:

**Add A New User**

**Username** AliceB

**Password** .....

**Confirm Password** .....

**Server User** alice

Add User

Once all the user information has been entered, simply click the Add User button and you will be returned to the main user listing, which will include the new user:

**ADMIN**

**User Listing**  
[Add New User](#)  
[Flush User Mappings](#)

**Users**

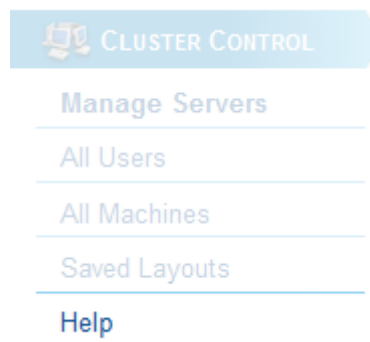
Username	Server User	Action
AliceB	alice	Action... ▾
admin		Action... ▾

## 3.5. Login to WebConsole

Now that a WebConsole user account has been created, you can login and start your first BigWorld server instance. To do this, return to the WebConsole login screen, enter the username and password that you created in the previous section, and press the Log In button. You should now be presented with a screen similar to the following:

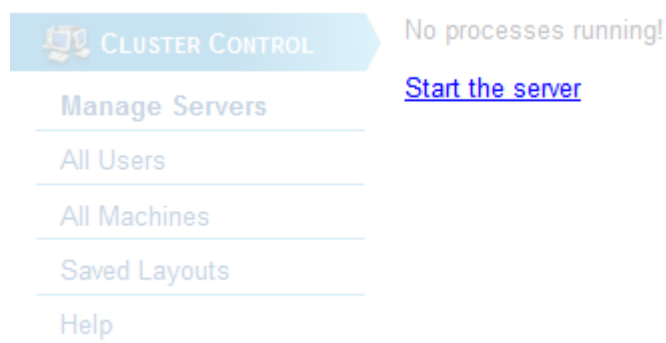


Each blue bar on the left hand side of the page contains a collection of loosely related functionality for interacting with a BigWorld server cluster. For information about each section refer to the 'Help' menu option in each module. For example, in the default display the help option is located as follows:



### 3.6. Starting a BigWorld server

With all the hard work out of the way, you are now only a few mouse clicks away from having your first server up and running. From WebConsole's Cluster Control→Manage Servers module click the *Start the server* link:



You should now be presented with a web page outlining your user information, the separate components of the `.bwmachined.conf` file, and a list of machines on which you can start the server:

Server Environment	
User	alice
UID	3514
MF_ROOT	/opt/bigworld/current/server
BW_RES_PATH	/home/alice/project_name/res:/opt/bigworld/current/server/res

Start	
<input checked="" type="radio"/> On a single machine:	centos01-vm ▾
<input type="radio"/> From a saved layout:	▾
<input type="radio"/> On a group of machines:	(use all machines) ▾
Restrict components by tags:	<input checked="" type="checkbox"/>
<input type="button" value="Go!"/>	

The purpose of this page is to allow you to review both your configuration settings and the machine (or machines) on which you would like to start the server prior to launching the cluster. For our purposes we will use the default settings which should launch the server on a single machine, identified using either your hostname or the IP address of your local machine. To start the server simply press the 'Go!' button. You should now see an interim web page that automatically updates as each server process starts:

Starting The Server				
Component	Dead	Running	Registered	Details
cellappmgr	0	0	1	
baseappmgr	0	0	1	
loginapp	0	0	1	
dbmgr	0	0	1	
cellapp	0	1	0	
baseapp	0	1	0	

The above image shows us all the server processes except the BaseApp and CellApp have started running and are operational. The BaseApp and CellApp processes are the last to start as they rely on all the other server processes to operate correctly.

You should now have an active server instance which presents you with a web page similar to the following:

Processes for alice					
Process Name	Machine	CPU Load	Memory Usage	PID	Action
<a href="#">cellappmgr</a>	<a href="#">localhost</a>	0.0%	7.5%	1742	Action... ▼
<a href="#">baseappmgr</a>	<a href="#">localhost</a>	0.0%	7.5%	1743	Action... ▼
<a href="#">loginapp</a>	<a href="#">localhost</a>	0.0%	7.5%	1745	Action... ▼
<a href="#">dbmgr</a>	<a href="#">localhost</a>	0.0%	68.6%	1744	Action... ▼
<a href="#">cellapp01</a>	<a href="#">localhost</a>	3.9%	7.5%	1747	Action... ▼
<a href="#">baseapp01</a>	<a href="#">localhost</a>	0.0%	66.3%	1746	Action... ▼

[Start more processes](#) | 
 [Restart the server](#) | 
 [Stop the server](#) | 
 [Save this server layout](#) | 
 [View log](#)

At this point you should be ready to move on to the Tutorial which will start to lead you through the process of creating your own game and understanding the programming concepts involved in interacting with BigWorld.

Good luck with your development!

## Chapter 4. Cluster Configuration

There are a number of issues within a network cluster that can influence the behaviour of a BigWorld server. This chapter outlines these issues along with the steps that should be taken both to deal with them and to ensure the optimal performance of your BigWorld cluster environment.

### 4.1. Security

In a BigWorld Server cluster, not all machines are connected to the public internet, but those that are must be secured well.

This is most easily achieved by using a firewall to block incoming packets. In general, the approach for machines with public IP addresses should be to block all incoming packets on the interface with the public IP.

The exception is that LoginApps and BaseApps need to allow UDP traffic to the ports on which they listen. The ports to be used are defined in the `res/server/bw.xml` file using the options `loginApp/externalPorts/port` and `baseApp/externalPorts/port`. For details on these options, see the Server Operations Guide's chapter *Server Configuration with bw.xml*, sections "BaseApp Configuration Options" and "LoginApp Configuration Options".

Using the Linux firewall configuration tool, **iptables**, we can add a rule to drop all incoming traffic on the external interface. In the following examples, we assume the external interface is `eth1`. For example:

```
# /sbin/iptables -A INPUT -i eth1 -j DROP
```

On the machines running LoginApps, we can add a rule to allow traffic through on the login port, using the default port of 20013, as illustrated below:

```
# /sbin/iptables -I INPUT 1 -p udp -i eth1 --destination-port 20013 -j ACCEPT
```

Similarly, for machines running BaseApps, we add similar rules to allow traffic through on the BaseApp external port as specified in the `baseApp/externalPorts/port` options.

We use `-I INPUT 1` in the new rule instead of `-A INPUT` because **iptables** applies the first rule in the chain that matches an incoming packet. Therefore, we need to insert the rule for accepting login packets before the rule for rejecting all UDP traffic on `eth1`.

For a production server, you should disable all networking services apart from SSH from trusted IP addresses.

BWMachined requires the ability to broadcast on the internal interface and receive back its own replies on UDP ports 20018 and 20019. The internal interface is denoted here as `eth0`. The firewall rules should accommodate this requirement.

For example:

```
# /sbin/iptables -I INPUT 1 -p udp -i eth0 -m multiport --destination-ports 20018:20019 -j ACCEPT
```

For details about the ports used by a BigWorld server see *Security*.

You must ensure that the firewall rules are restored each time the machine boots. You can use the following command to save the **iptables** configuration:

```
# /etc/init.d/iptables save
```

## 4.2. Routing

A large number of tools and server components in BigWorld Technology rely on being able to send IP broadcast packets to the default broadcast address (255.255.255.255), and for them to be routed correctly.

This will happen by default on a machine with only one network interface (i.e., machines on the internal network only, such as CellApp machines, DBMgr machines, etc.).

For machines with two network interfaces (i.e., BaseApp and LoginApp machines), we need to make sure that packets sent to the broadcast address are routed via the internal interface.

We can make sure this is done correctly by making an entry in the kernel routing table with the **ip** command. This command may not be installed by default. You can install this utility by running the following command as root:

```
# yum install iproute
```

In the example below, we once again assume that `eth0` interface is the internal network. To add a default broadcast route, run the following command as the root user:

```
# /sbin/ip route add broadcast 255.255.255.255 dev eth0
```

This command will only add the route to the current routing table, and will not apply after rebooting your machine. In order to ensure this route is applied whenever the `eth0` interface is brought online run the following command as the root user:

```
# echo "broadcast 255.255.255.255 dev eth0" > /etc/sysconfig/network-scripts/  
route-eth0
```

This command will create the file `/etc/sysconfig/network-scripts/route-eth0` if it doesn't already exist.

## 4.3. Buffer sizes

Some of BigWorld's network components require socket buffers that are generally larger than system defaults. In order for these components to work properly, the amount of memory allocated for these buffers must be increased. This involves values: the maximum read and write buffer sizes, and the default write buffer size.

If you installed BWMachined using the RPM package, these values should have been automatically added or updated in the `/etc/sysctl.conf` file, and you can skip this section.

The size allocated for socket buffers is determined from kernel settings, which can be modified on the fly using the **sysctl** command. For example, the maximum size of read buffers can be increased to 16 MB with:

```
# /sbin/sysctl -w net.core.rmem_max=16777215
```

However, to make these changes persistent, we strongly recommend defining higher values in the `/etc/sysctl.conf` file. The entries for the relevant settings should have the following values:

```
net.core.rmem_max = 16777216
net.core.wmem_max = 1048576
net.core.wmem_default = 1048576
```

## 4.4. Disabling cron jobs

Cron is a system daemon which enables tasks to be scheduled for running at pre-determined intervals, such as hourly, daily, weekly, etc. Cron refers to these periodic tasks as *jobs*. These jobs may adversely affect the performance of a running server due to the kind of functionality they perform. For example it is quite common for cron jobs to update the **locate** database which involves performing a recursive directory listing on the entire machine. These kinds of jobs can involve reading from each part of a hard drive, effectively causing a flush of the disk cache Linux has in memory. This can cause a momentary lapse in performance of server processes, as disk swapping starts to occur on the host machine.

We recommend that BigWorld Server machines have resource-intensive (CPU, memory or disk) cron jobs disabled in production environments. You can achieve this (with various levels of granularity) by disabling these cron jobs.

Cron jobs can be disabled by clearing the executable bit on the relevant job. For example, to disable the job run by `/etc/cron.daily/makewhatis.cron`:

```
# chmod -x /etc/cron.daily/makewhatis.cron
```

Re-enabling cronjobs can be done by setting the executable bit using the reverse operation:

```
# chmod +x /etc/cron.daily/makewhatis.cron
```

- System cron jobs are stored in the following locations:
  - `/etc/cron.d` (contains cron jobs for system services)
  - `/etc/cron.hourly` (for hourly cron job scripts)
  - `/etc/cron.daily` (for daily cron job scripts)
  - `/etc/cron.weekly` (for weekly cron job scripts)
  - `/etc/cron.monthly` (for monthly cron job scripts)

Also remove any unnecessary user-level cron jobs. These can be listed per-user using:

```
$ crontab -l
```

### Note

We do not recommend completely disabling the cron service as facilities such as log rotation and some security mechanisms may rely on the cron service being active.

# Appendix A. Hardware Requirements

## Table of Contents

A.1. CPU Specifications .....	27
A.2. Duals/Single/Quad/Blades .....	27
A.3. Network Interface Cards .....	28
A.4. Disk Storage .....	28
A.5. Power Supplies .....	28
A.6. Memory .....	28
A.7. NOC Bandwidth .....	29
A.8. VMWare .....	29

The appendix is intended as a quick summary of the required hardware specification needed in order to run a BigWorld. The following list should be considered as the minimum set of required hardware:

- 1GHz CPU (non-mobile CPU preferred)
- 256MB RAM
- 8GB Hard Disk
- 100Mbps Network Interface Card

The following list outlines BigWorld's recommended hardware requirements for an optimal price vs performance solution.

### A.1. CPU Specifications

In general, use the fastest CPU's that you can. Faster CPU's mean more entities per CPU and fewer machines.

As far as which CPU to buy, look at the normal kinds of things for servers: big L1 and L2 cache sizes, and fast front-side buses are always better than small L1 and L2 cache sizes and slow front-side buses.

Multiple processors will help you out due to lower network traffic and fewer machines, until such time as the processors are generating too much data to get to the network card (or over the PCI bus to the network card).

If you have multiple processors, then make sure that you have one server component running on each CPU.

### A.2. Duals/Single/Quad/Blades

In general CPU density in each box is a price decision. Blade machines are expensive, but if you are paying expensive rates for your NOC it may work out to be cheaper. If we ignored the NOC cost, we recommend dual CPU machines.

An example Blade setup would be as follows:

#### BladeCenter LS20 885051U

- Processor: Low Power AMD Opteron Processor Model 246 (Standard)
- Memory: 4 GB PC3200 ECC DDR RDIMM (2 x 2 GB Kit) System Memory
- IBM eServer BladeCenter™ Gigabit Ethernet Expansion Card
- SCSI Hard disk drive 1 : 73GB Non Hot-Swap 2.5" 10K RPM Ultra320 SCSI HDD

### BladeCenter 86773XU

- Optical device: IBM 8X Max DVD-ROM Ultrabay Slim Drive (Standard)
- Diskette drive: IBM 1.44MB 3.5-inch Diskette Drive (Standard)
- Power supply modules 1 and 2: BladeCenter 2000W Power Supplies one and two (Standard)
- Management modules: BladeCenter KVM / Management Module (Standard)
- Switch module bay 1: Nortel Networks Layer 2/3 Copper GbE Switch Module for IBM eServer BladeCenter
- Switch module bay 2: Nortel Networks Layer 2/3 Copper GbE Switch Module for IBM eServer BladeCenter

A setup with 10 LS20 blades in a 86773XU BladeCenter would cost approx \$60k USD.

## A.3. Network Interface Cards

1Gbps NICs are recommended. The accurate way to determine what NIC is required is to measure the inter-server traffic for your game. If the traffic is reaching 25% of the cards capacity we recommend using a faster card (i.e. if the traffic is more than 25Mbps use a 1Gbps NIC). Note most 100Mbps NIC cards cannot handle more than 50Mbps sustained throughput.

## A.4. Disk Storage

For the general machines (CellApps, BaseApps, and the various managers) RAID disk setups are not recommended. None of these machines use the disk sub-system extensively (and are certainly not disk bound). Use standard drives, that are big enough to store the entire world data (typically in the order of 1 to 10G). If a drive dies it can be replaced and the data copied from the master.

The machine with the master copy of the data should use a RAID 5 system for speed and data integrity. Hot-swap drives will facilitate easy replacement when drives fail. The database server also needs to use RAID 5 for data integrity and Logical Volume Management for snapshotting. We also recommend using 10k or 15k RPM drives (SATA or SCSI). The database for a game stores backup copies of all entities. This database can be large, potentially 10G to 1TB, but this should be measured during development. This can be estimated by multiplying the number of entities by their size.

## A.5. Power Supplies

We recommend the use of dual redundant PSUs for the database machine, and the master data server. All other machines can use standard single PSU's, since a failure of these machines is not critical. It is cheaper to let the BigWorld fault tolerance system do the work on the software side.

## A.6. Memory

To calculate memory requirements for the CellApp we recommend the following:

- Around 32MB to 128MB, for Linux to run comfortably (depends on how well you have stripped back the kernel and system services).
- Around 32MB for a CellApp or BaseApp to run comfortably with no entities on them and no spaces loaded.
- Enough RAM for your entities, and for your world geometry (remember that the cell needs to load up enough geometry to cover for the AoI for all entities it is supporting). This amount will depend on how dense your meshes are, and how much data is stored with each entity. 2GB of RAM would not be unusual for an average game.

The BaseApps would typically require around 512M, depending on how much entity data is stored. All other machines require around 512M.

## A.7. NOC Bandwidth

This is easy to calculate. Multiply the number of players by the desired bandwidth per player. Outgoing bandwidth is generally higher than the incoming bandwidth.

## A.8. VMWare

It is possible to use VMWare for single developer testing purposes, however VMWare is not recommended as a scaleable production configuration due to the timing latency that can be introduced. It is also important to note that if you are intending to use a VMWare image that the architecture of the package you are creating should be same as the intended machine you will run the image on. While this may seem counter-intuitive, this is important as the cross-architecture emulation significantly slows down the server and will generally causing process deaths due to a lack of responsiveness.

# Appendix B. Installing CentOS 5

## Table of Contents

B.1. Installing .....	31
B.2. Post-installation Setup .....	33
B.2.1. Install updates .....	33
B.2.2. Configure services .....	33
B.2.3. Install build tools .....	34
B.2.4. MySQL server and development libraries .....	34
B.2.5. Changing UIDs .....	34

### Note

Even experienced users should skim the following sections to make sure that required packages are installed.

You may wish to refer to the Centos documentation (<http://www.centos.org/docs/5/>) for additional notes and guidelines on installing and configuring CentOS.

## B.1. Installing

- Boot the computer using the installation DVD, or other media (for example PXE boot). See the CentOS documentation for further details.

You may need to select the CD/DVD ROM drive as a bootable device in the BIOS when installing from DVD.

- This installation guide is based on the graphical installer. Press ENTER at the first boot screen to select Install in graphical mode.

If you are having trouble with video card drivers, then you can reboot and try the text-only installer.

- If this is the first time that the CD/DVD has been used, then it is worthwhile to use the built-in test option.

The test will take around 15 minutes. If you do not want to test, just select Skip.

- **Language and keyboard type.**

The language selected will be used for the installation procedure as well as being the default language of the installed system.

- **Installation method**

Choose Local CDROM if you are using DVD to install, or you can choose your local CentOS mirror.

- **Disk partitioning**

You can partition the disk as you see fit. The Remove all partitions on selected drives and create default layout option should work for most situations. You can modify the default layout by checking the Review and modify partitioning layout checkbox.

### Note

If the machine will host the database server (running MySQL and DBMgr), and you are using secondary databases, you will need to use LVM partitions and you will also need to allocate some free space for the LVM snapshot. You can add unallocated space on one of your logical drives when reviewing the partitioning layout. See “Database Snapshot Tool” for more details about the snapshotting tool.

- **Boot loader configuration**

Select the appropriate options for your machine (by default, the bootloader will be installed to the MBR).

- **Network configuration**

Ensure that you have at least one network device listed, and that IPv4 is enabled for it.

In production, for BaseApp and LoginApp machines, there should be two network interfaces, one for external traffic, and the other for internal server traffic. In development, these can be the same.

The hostname can be specified manually or it can be set from DHCP.

If you are not using DHCP, you will need to enter the default gateway and DNS addresses.

- **Time zone selection**

Select your time zone.

We recommend that you leave the system clock as UTC as per the default.

- **Setting the root password**

You will need access to the `root` account to install some BigWorld Server components, make sure you remember this password.

- **Package selection**

If this is a production machine, we recommend that you have `Desktop - Gnome` unchecked. You can leave the other options unchecked, the specific packages that the BigWorld server requires will be installed later on in this guide.

In development, you may wish to use the machine as a desktop development machine, in which case you can choose to install whichever packages you require for development, such as the `Desktop - Gnome` package group.

- **First boot configuration**

The installation program will format the disk partitions, install the base system and system packages. After this process is complete, you will be asked to reboot the machine. On first boot, you will be prompted for further configuration.

- **Authentication**

This tool sets up how your OS will look up user account information.

BigWorld components assume that the username to UID mapping is unique across the network i.e. two users with the same name on two different machines will have the same UID and vice versa. If you are creating accounts with the same name on multiple machines, please ensure that they all have the UID by manually specifying their UID.

Furthermore, the BigWorld server also assumes that server components started by the same user (as identified by their UID) belongs to the same server instance, even when those components are running on different machines. To run multiple BigWorld Server instances, multiple user accounts are needed.

You can set up remote account information servers such as LDAP. We recommend using LDAP during development to ensure that every machine in the cluster has the same user set. Typically, each developer user has an account where they can run their own servers independent of other users.

Refer to the OpenLDAP documentation for further information on how to configure an LDAP service to authenticate users.

- **Firewall configuration**

For a development machine, the firewall should be disabled. The default firewall blocks all UDP traffic, which prevents the BigWorld Server from operating. You can disable the firewall by setting the `SecurityLevel` option to `Disabled`.

For a production machine, you will need to setup specialised firewall rules for your specific security requirements. Guides for BigWorld Server specific firewall settings are given in the *Cluster Configuration* on page 23 section of this document.

The BigWorld Server is known to work with the default SELinux settings (enforcing).

- **System services**

For production machines, in order to avoid unexpected load spikes on your system from background services, we recommend that you disable any non-essential services. You can configure which services are started up at boot time.

Services that are recommended to be disabled include:

- cups
- bluetooth
- yum-updatesd

- **Finishing the installation**

Once exiting the first boot configuration screen, you will be presented with a login prompt. Login as the root user to continue the installation.

## B.2. Post-installation Setup

### B.2.1. Install updates

Although not strictly required, it is a good idea to install the latest updates. You can update the packages by running the following command as root:

```
# yum update
```

### B.2.2. Configure services

In order to avoid unexpected load spikes on your system from background services, non-essential services should be disabled. The service configuration can be modified by running the following as root:

```
# firstboot --reconfig
```

This will bring up the same configuration menu that appears after the OS has been installed. Select the option `System services`, and uncheck those services that you don't wish to start at boot time. See above in "Installing" on page 31 for a list of recommended services to disable.

### B.2.3. Install build tools

---

To build the BigWorld Server, GCC and Make must be installed. These should be the default compiler and make utility on your Linux installation.

```
# yum install gcc-c++ make
```

### B.2.4. MySQL server and development libraries

---

This step is required if you wish to use BigWorld WebConsole and StatLogger, or use DBMgr with MySQL support.

The BigWorld Server is compatible with MySQL 5.0 and MySQL 5.1. Support for MySQL 4.x has been deprecated.

If you have not already installed the MySQL server, you can do so by running the following command as the root user:

```
# yum install mysql-server
```

If you intend to use DBMgr with MySQL support, you must also ensure you have the MySQL client development package, as this is needed to rebuild DBMgr with MySQL support.

Install the MySQL client development package by running the following command as the root user:

```
# yum install mysql-devel
```

### B.2.5. Changing UIDs

---

A requirement of BWManaged is that all machines in your cluster must have the same user account information, in particular, that the numerical user IDs (UID) for a particular username are the same on every machine.

When setting up your cluster, one system in your cluster may end up with a mapping of UIDs to usernames that is different from other systems in your cluster. This is especially likely if you are not using LDAP or a similar tool to synchronise login names.

If you are using the GNOME desktop environment on these machines, there can be problems when you change the UID for a username. This section outlines the steps to change the UID of a username and avoid these problems. If all the machines were set up such that each user account has the same UID on each machine, you can skip this section.

1. Make sure no one is logged in graphically.
2. If you are in graphical mode, press `CTRL + ALT + F1` to switch to a text console.
3. Log in as `root`.
4. Choose the new user ID and group ID for your user, making sure that the new user ID is not being used by any other user, and similarly that the group ID is not used by another group. You can check by looking

through `/etc/passwd` and `/etc/group`. By convention, the user's primary group ID and user ID are the same, though this need not be the case.

5. Change the user ID and group ID of the user and the user's primary group by invoking the following commands:

```
# groupmod -g <new GID> <groupname>
# usermod -u <new UID> -g <new GID> <username>
```

6. Confirm that the new user has the new UID and GID:

```
# id <username>
```

7. Issue the command below to remove any invalidated user state:

```
# rm -rf /tmp/*<username>*
```

### Note

`rm` is the command to remove files, and the `-rf` flags direct a recursive search, and force deletion of all directories and files that match the name, without prompting you. These flags should therefore only ever be used with extreme care.

The asterisks represent sets of wild-card characters in the search for files to remove. For example, for the user Alice, this command would remove a file called `/tmp/mapping-Alice` if it were to exist.

8. You will now need to change the ownership of the home directory to the new user and group:

```
# chown -R <username>:<groupname> /home/<username>
```

9. Press `CTRL + ALT + F7` to return to the graphical login if required.

# Appendix C. Copying files to Linux

## Table of Contents

C.1. USB Thumb Drive / External Hard Disk .....	37
C.1.1. Installing NTFS Support .....	38
C.2. Python SimpleHTTPServer hosted on Windows .....	38
C.3. Windows network share .....	38

For people unfamiliar with using Linux, even a simple task such as copying files from a Windows machine to a Linux machine can be daunting. This section aims to assist this process by providing a number of (hopefully) convenient alternatives that can be used to assist in transferring files between machines.

There a number of approaches that can be used for copying files from Windows to Linux. These include but are not limited to:

- USB Thumb Drive / External Hard Disk
- Python SimpleHTTPServer hosted on Windows
- Windows network share

Below is a brief outline of each of the mentioned methods. We encourage you to investigate each approach independently in order to feel comfortable performing these operations whenever required without assistance from BigWorld.

The following examples will assume you are copying the RPM files from a standard BigWorld package which are located in the `rpm` directory at the top level of your package.

It doesn't matter where you copy the files on the Linux machine. However, the commands listed in the *Simple Installation* on page 7 section are given under the assumption that you copy the files to `/root`.

### Note

In Linux the entire file system is organised in a single hierarchy. The location `/` refers to the root level and subdirectories are listed after that. For example, to switch to the `/usr/include` directory, you can use the `cd`, or 'change directory' command in the following way:

```
$ cd /usr/include
```

If you are new to Linux, you should read the article at [www.freeos.com/articles/3102](http://www.freeos.com/articles/3102), which describes the Linux file system in more detail.

## C.1. USB Thumb Drive / External Hard Disk

When using this approach you may need to pay attention to the filesystem type of the drive you are using. A FAT32 filesystem is natively supported by Linux, while NTFS is not. Quite often, drives that have been formatted under Windows will have an NTFS filesystem rather than FAT32. If this occurs in your drive, see "Installing NTFS Support" on page 38 .

This approach is perhaps the easiest for copying a small number of files between machines on a once-off basis. Once you have copied the files from your Windows machine to the USB device and inserted it into the

Linux machine the device should be auto-mounted. If you are logged into a graphical account you should see the device appear as a new icon on your desktop.

Command line only users may need to perform more steps to discover where the device was mounted. The **mount** command should provide you with a list of all device-to-directory mappings to enable you to discover the directory in which the files are located.

### C.1.1. Installing NTFS Support

To install NTFS support for your CentOS installation, firstly ensure that you have installed the EPEL repository as outlined in “Installing the EPEL Repository” on page 8 and then run the following command as root:

```
# yum install ntfs-3g ntfsprogs
```

## C.2. Python SimpleHTTPServer hosted on Windows

This approach assumes that you have installed Python on your Windows machine and have the Python executables in your \$PATH environment variable.

After installing or downloading your Bigworld Technology package, open a command line window by clicking on Run in the Start menu, and typing **cmd** into the prompt. Navigate to the rpm directory in your new installation and type:

```
C:\BigWorld\rpm> python.exe -m SimpleHTTPServer
rving HTTP on 0.0.0.0 port 8000 ...
```

You can now use the **wget** program or a Web browser on your Linux machine to copy files via HTTP. If you are unsure of your Windows machine IP address you can use the **ipconfig** program on the command line. For example, to copy the 2.1 bwmachined RPM file you would use the following command, replacing 10.40.3.145 with your own IP address:

```
$ wget http://10.40.3.145:8000/bigworld-bwmachined-2.1.0.x86_64.rpm

--2011-11-05 11:16:28-- http://10.40.3.145:8000/bigworld-
bwmachined-2.1.0.x86_64.rpm
Connecting to 10.40.3.145:8000... connected.
HTTP request sent, awaiting response... 200 OK
Length: 1096149 (1.0M) [application/octet-stream]
Saving to: `bigworld-bwmachined-2.1.0.x86_64.rpm'

100%[=====] 1,096,149 4.93M/s in 0.3s

2011-11-05 11:16:29 (11.8 MB/s) - `bigworld-bwmachined-2.1.0.x86_64.rpm' saved
[1096149/1096149]
```

## C.3. Windows network share

Copying files from a Windows network share can be extremely convenient, but depending on your network setup it may be problematic to initially setup. In order to use this file-copying mechanism, ensure that you share your windows files using Advanced network file sharing, not the default simple file sharing. More detailed instructions on configuring a network share on Windows can be found in the Server Programming Guide's chapter *Shared Development Environments*.

For simplicity's sake, to retrieve the files on the Linux machine you should use a graphical login and navigate to your Windows machine using the Places→Network Servers menu options.

# Appendix D. Creating a custom BigWorld server installation

## Table of Contents

D.1. Customising RPMs .....	39
D.2. Manual Installation .....	39
D.2.1. BWMachined .....	39
D.2.2. Server .....	40
D.3. Server Tools .....	40
D.3.1. Requirements and Caveats .....	40

### Note

This appendix does not apply to Indie package customers.

As a game advances through its development process it may become necessary to perform a custom installation of the BigWorld server. This may be due to BigWorld server binaries having been regenerated, or production environment machines needing to alter the installation location from the default provided in the RPM files.

This chapter outlines some of the approaches that can be used to customise the installation of each BigWorld server component. Whichever approach you choose to use, we assume that you have obtained the BigWorld Technology package correctly. This is to ensure that there are no line ending issues, such as having Windows CRLF line endings when attempting to use scripts from Linux, which assume LF only line endings.

This chapter will also assume that you have a solid understanding of Linux (in particular the CentOS file system hierarchy) and are confident in navigating the environment on the command line to locate files as required.

## D.1. Customising RPMs

BigWorld distributes the RPM .spec files and Python scripts that are used for building the officially shipped RPMs. This enables you to customise the RPMs for your own environment should you need to.

Should you wish to use this approach of customisation, please refer to the Server Operations Guide's chapter *RPM* which fully outlines both how the BigWorld RPM build scripts work, and approaches that can be used for automating the distribution of the RPMs within a large network environment.

## D.2. Manual Installation

### D.2.1. BWMachined

Out of all the BigWorld server components, BWMachined is perhaps the easiest to customise the installation of due to its simplicity.

To install BWMachined manually you can run the `bwmachined2.sh` script as the root user in the `big-world/tools/server/install` directory as follows:

```
# ./bwmachined2.sh install
```

This operation will perform the following steps:

- Stop any existing installed BWMachined daemon, and uninstall it if it does exist. This will not uninstall an RPM package of BWMachined however.
- Create the BWMachined init script in `/etc/rc.d/init.d/`.
- Create the symbolic links in `/etc/rc[1-5].d/`. It is setup to stop in `rc1.d/` and start in `rc[2-5].d/`. These can be changed manually if desired.
- Copy the BWMachined executable to the directory `/usr/sbin/`.
- Launch the executable in daemon mode as if the init script were called with the `start` argument.

## D.2.2. Server

---

A custom installation of the BigWorld server is not constrained by any installation scripts or file system locations. All that is necessary is that the user(s) that will run the BigWorld server on each machine in the cluster have access to the server binaries and the game resources. This means that it is possible to install the BigWorld server binaries either in a user's home directory, or in any other location on a host.

Because it is generally enforced that the `.bwmachined.conf` file is the same between machines for an individual user in a BigWorld network cluster, it is preferable to ensure that the installation location of the BigWorld server binaries is consistent on each machined. This also ensures that ongoing maintenance of the machines is much easier.

If you choose to manage your own BigWorld server binary installation you may wish to refer to some of the options mentioned in the document Release Planning's section "Distributing Game Resources".

## D.3. Server Tools

The server tools perhaps benefit the most from being installed from an RPM due to the complexity of sub-systems and configuration that is involved in installing them correctly. The following instructions describe dependency list as well as a basic installation guide although we no longer support a manual installation of the server tools.

### D.3.1. Requirements and Caveats

---

#### D.3.1.1. Requirements

- BWMachined installed and operational on the server tools machine
- Dedicated user account for the server tools
- MySQL Database
- Python 2.4 or higher (default for RedHat / CentOS)
- TurboGears v1.x (default for RedHat / CentOS)

#### D.3.1.2. Dependencies

WebConsole by default uses an SQLite database for managing all persistent data such as user information, preferences, etc.

The statistics-collecting process, StatLogger, relies on a MySQL database server for storing process and machine statistics.

### D.3.1.3. Installation process

The installation process will not be outlined in a step-by-step manner as has been done for the RPM installation process. If you are attempting a custom installation of the BigWorld server tools it is assumed that you have enough system understanding and experience with BigWorld to perform each step described below.

A custom installation of the BigWorld server tools requires you to perform the following steps:

- Install the EPEL repository as outlined in “Installing the EPEL Repository” on page 8 .
- Install the `MySQL-python` RPM package as distributed with the BigWorld package to prevent any WebConsole and StatLogger connection issues.
- Install all dependencies.

Even when performing a custom installation we recommend that you install the dependency chain via the default CentOS repositories used for the BigWorld RPMs, as it will ensure that the versions of installed packages are correct.

The following yum packages should be installed:

- `python-setuptools`
- `python-sqlobject`
- `TurboGears`
- `yum update` your distribution to ensure all security and bug fixes have been applied.
- Create a user account within your domain which will be used to run the tools.

For example, create the user `bwtools` that is able to be resolved on all machines in the server cluster and has a unique user ID on all cluster machines.

- Ensure the MySQL server is running and enabled for all appropriate runlevels.
- Create a MySQL server account for the BigWorld server tools.
- Install the BigWorld server tools into the BigWorld tools users directory.
- Set the SELinux security context for `bwlog.so`.
- Update the WebConsole configuration file.
- Update the StatLogger configuration file.
- Update the MessageLogger configuration file.
- Copy the service scripts from `bigworld/tools/server/install` into the directory `/etc/init.d` and set the service runlevels.
- Start MessageLogger, StatLogger and WebConsole.
- Configure log rotation for the server tools log files.
- Verify that all services are running after a server restart.

# Appendix E. Understanding the BigWorld Machine Daemon (BWMachineD)

## Table of Contents

E.1. How BWMachined Works .....	43
E.2. Configuring BWMachined .....	43
E.2.1. Creating ~/.bwmachined.conf .....	43
E.2.2. Creating /etc/bwmachined.conf .....	45

## E.1. How BWMachined Works

For information about the role of BWMachined please refer to the Server Overview's chapter "BWMachined".

In order for the BWMachined process to act as an agent for starting and locating server processes for users within a network cluster it is necessary to have a mechanism for BWMachineD to be able to locate game resources on a per user basis on each machine it is running on.

The approach chosen for BWMachineD was to have a configuration file in each user's home directory that intends to launch any BigWorld processes. This configuration file directs BWMachineD as to where it can locate the BigWorld server binaries as well as the game resources to use on each machine. While it is uncommon this approach allows each machine to potentially have a different installation location for the server binaries and game resources.

When interacting with the BigWorld server tools and requesting to launch a server, a message will be sent to BWMachineD which in turn queries the configuration file associated with the requesting user. The paths specified in the configuration file are then used by the BWMachineD process when it attempts to launch the server processes. For more details regarding the layout of the BWMachineD configuration files please refer to the following section "Configuring BWMachined" on page 43 .

## E.2. Configuring BWMachined

BWMachined plays a crucial role in the ongoing operation of your cluster environment, so it's important to understand how it can be configured and which configuration options are relevant for your server environment.

There are two configuration files which are relevant to the operation of BWMachined:

- ~/.bwmachined.conf

This file is used to specify options relating to how an individual user working within a BigWorld cluster should find the server resources required to operate on any available cluster machines.

- /etc/bwmachined.conf

This file is primarily used to specify settings relating to how the machine on which BWMachined is running should operate within the cluster environment.

### E.2.1. Creating ~/.bwmachined.conf

When starting a server and related components there are two important pieces of information that are required:

- Where to find the BigWorld server executable files.

- Which directories contain the game resources to use with the server.

The preferred way to specify these settings is in the file `~/ .bwmachined.conf`.

### Note

For users not familiar with Linux:

- The `~` (tilde) indicates the user's home directory, for example a user called `johns` would generally have a home directory located at `/home/johns`
- The period character before the filename indicates it is a hidden file which causes it to not be displayed by many directory listing applications including `ls` (unless the `-a` option is specified).

Below is an example of a `~/ .bwmachined.conf` file for a user `johns`. This file needs to be manually created when the user account is created. Note the different places where a semi-colon and a colon are used:

The preferred way to specify these settings is in a file `~/ .bwmachined.conf`.

### Note

For users not familiar with Linux:

- The `~` (tilde) indicates the user's home directory, for example a user called `alice` would generally have a home directory located at `/home/alice`
- The period character before the filename indicates it is a hidden file which causes it to not be displayed by many directory listing applications including `ls` (unless the `-a` option is specified).

Below is an example of a `~/ .bwmachined.conf` file for a user `alice`. This file needs to be manually created when the user account is created. Note the different places where a semi-colon and a colon are used.

```
# .bwmachined.conf
# Format: BW_ROOT;BW_RES_PATH:[BW_RES_PATH] ...
/opt/bigworld/current/server;/home/alice/fantasydemo/res:/opt/bigworld/
current/server/res
```

The path before the semi-colon should point to the root directory of the installed files. You will generally have the `bigworld` directory underneath this root directory. The paths after the semi-colon (separated by colon characters) specify the resource paths that will be used to find resources used by your game.

When starting a BigWorld server using the standard server tools, such as **control\_cluster.py** or WebConsole, BWMachineD is responsible for launching the server binaries and uses the information located in `~/ .bwmachined.conf` as well as the architecture of the host system to determine how to launch the server for the requesting user.

### Note

Each user that needs to run a server within your cluster environment will need a `~/ .bwmachined.conf` file created and configured for them.

## E.2.2. Creating `/etc/bwmachined.conf`

The global configuration file `/etc/bwmachined.conf` is used for setting options that define how BWMachineD will operate on the host it is running on. For example, if you have multiple machines in your cluster, and during development you wish to isolate certain machines into groups for developer usage, this would be applied in the global configuration file. The following list provides a quick summary of host based configuration options that may be applied in the file `/etc/bwmachined.conf`:

- User defined categories
- Reviver configuration
- BigWorld server timing method
- Interface configuration for multi-interface hosts

### Note

The configuration file is only read when BWMachineD starts. It will have to be restarted if you want it to acknowledge your changes.

### E.2.2.1. Reviver Configuration

When the Reviver process starts, it queries the local BWMachineD process, and will only support the components that have an entry in a special user defined category called `[Components]`. An example configuration specifying that the Reviver should support all server components would be defined as below:

```
[Components]
baseApp
baseAppMgr
cellApp
cellAppMgr
dbMgr
loginApp
```

### Note

BaseApp and CellApp will not be restarted by Reviver, the `[Components]` entries are used by the WebConsole and `control_cluster.py` to determine which processes should be started by BWMachineD on that host. This list however is only a hint for the server tools and the processes may still be started on that host if required.

### Note

If the `[Components]` category does not contain any entries, then Reviver will support all server components.

### E.2.2.2. Timing Method

By default, time services is provided by the `clock_gettime` system call. The default is the recommended timing method and works for all supported platforms, for further options, please refer to the Server Operations Guide's chapter *Clock*.

### E.2.2.3. Machine groups

The membership of the machine running BWMachined can also be optionally specified in `/etc/bwmachined.conf`. For more details on what machine groups are used for, and how to specify them, refer to the Server Operations Guide's chapter *Machine Groups and Categories*.

### E.2.2.4. Internal Interface Configuration for Multi-interface Hosts

The BigWorld machine daemon is used for both internal machines and out-facing machines such as those that BaseApps and LoginApps are run on. The protocol that BigWorld components use for discovery of server processes, and process startup registry involves a UDP broadcast to the machine daemons.

The machine daemon must determine the interface to receive these broadcasts from. By default, the machine daemon will determine which interface is the internal interface by sending a broadcast packet on each interface and waiting for this broadcast packet to be returned. The interface which receives the first broadcast packet is assumed to be the internal network.

For the out-facing machines with more than one interface, this may result in the incorrect interface being chosen. In these situations, it is best to check your broadcast routing rules<sup>1</sup> and consider adding firewall rules to block receiving from BWMachined ports (20018 and 20019) on these other interfaces.

For example, if `eth1` is not your internal interface:

```
# /sbin/iptables -I INPUT 1 -p udp -i eth1 -m multiport --destination-ports
20018:20019 -j DROP
```

#### Note

In most circumstances it is better to block all ports and then only open the ports that are needed. See “Security” on page 23 .

In rare cases, the `[InternalInterface]` configuration option can be set with either the dotted-decimal address or the name of the interface that is connected to the internal network.

For example, using the name of the interface:

```
[InternalInterface]
eth0
```

Using dotted-decimal notation:

```
[InternalInterface]
192.168.0.1
```

If this option is not specified, auto-discovery of the internal interface will be performed. If the option is specified, but no interface is found that matches the value set in `[InternalInterface]`, then an error will be logged to syslog and the machine daemon process will terminate.

This option may now also be specified in the `[bwmachined]` section of `/etc/bigworld.conf` as follows:

```
[bwmachined]
```

<sup>1</sup>For more details, see “Routing” on page 24 .

```
internal_interface = <value>
```

# Appendix F. Troubleshooting

## Table of Contents

F.1. Check That BWMachined is Running .....	49
F.1.1. Troubleshooting BWMachined .....	49
F.2. StatLogger .....	50
F.3. BigWorld Support .....	50

While we attempt to prevent problems from occurring during the installation process, some will inevitably arise. This appendix aims to outline some of the more common failure cases that can occur when installing the BigWorld server and tools.

### F.1. Check That BWMachined is Running

- To check if the daemon is running, use **control\_cluster.py**<sup>1</sup> utility. To use **control\_cluster.py** to check the status of BWMachined, issue the following command:

```
$ bigworld/tools/server/control_cluster.py cinfo
```

The machines correctly running BWMachined should be displayed, as in the example below:

```
shire          10.40.3.37          0 processes    0%, 0% of 2000MHz (4% mem)
```

Make sure that the process has an address that is on the internal network. If it does not, make sure that your broadcast route is set correctly.

If your machine is not listed, but has BWMachined running, then you might need to check your firewall rules. For more details, see section “Security” on page 23 in this document.

You can also run the following command to check the relationship between bwmachined processes in the cluster.

```
$ bigworld/tools/server/control_cluster.py checkring
```

#### F.1.1. Troubleshooting BWMachined

Some of the most common problems that can occur when installing a BigWorld service are a result of BWMachined not running after the initial installation. The following outlines the most common issues encountered and how to resolve these problems. These steps should be checked in order.

- Is BWMachined running?

```
/sbin/service bwmachined2 status
```

If BWMachined is running, you may have to adjust firewall rules in order to allow UDP broadcast messages to be sent and received around the server cluster. For more information please refer to “Security” on page 23.

<sup>1</sup>For more details, see the Server Operations Guide’s section “Server Command-Line Utilities”.

If BWManaged is not running, continue with the following step.

- Was BWManaged able to start at all?

BWManaged is quite often able to successfully startup but will then fail when attempting to send broadcast messages out to the network. In order to determine what happened on startup we need to investigate the system logs (syslog) which can be found by default in `/var/log/messages`. Output from a successful startup should appear as follows:

```
/opt/bigworld/2.1/bwmanaged/sbin/bwmanaged2: --- BWManaged start ---
/opt/bigworld/2.1/bwmanaged/sbin/bwmanaged2: Host architecture: 64 bit.
/opt/bigworld/2.1/bwmanaged/sbin/bwmanaged2: Using gettime timing
/opt/bigworld/2.1/bwmanaged/sbin/bwmanaged2: Broadcast discovery receipt
from 10.40.3.145.
/opt/bigworld/2.1/bwmanaged/sbin/bwmanaged2: Confirmed 10.40.3.145 (eth0)
as default broadcast route interface.
```

The key line here is the final line that confirms the default broadcast route. If you do not see this line and in its place see an error message you will most likely need to adjust the default broadcast routing rules. For more details see “Routing” on page 24 .

If you are having trouble finding BWManaged output in your syslogs, try restarting BWManaged which should generate new logs as it starts.

## F.2. StatLogger

Some of the most common issues encountered with StatLogger include:

- Incorrect or no MySQL database connection info provided.

This can be resolved by following the instructions outlined in “Configuring StatLogger” on page 13 .

- Old versions of Python MySQLdb module being used.

This issue will generally manifest itself as StatLogger failing to reconnect to the MySQL server overnight when the StatLogger service is not being heavily loaded.

This issue can be resolved by installing the Python MySQLdb module provided by BigWorld as outlined in “BigWorld Server Tools” on page 13 .

## F.3. BigWorld Support

When all else fails and you are uncertain of how to procede, contacting BigWorld support is the next step. BigWorld provides a number of support methods.

For Indie package customers, using the forums is the best place to ask questions. The forums are located at <https://forum.bigworldtech.com>. These forums are a community interaction with other Indie customers who have most likely encountered the same issues as you. BigWorld staff do monitor and respond to these forums.

For non-Indie customers, using the PerlDesk ticket system available at <http://support.bigworldtech.com> is the recommended method of asking questions. This ticketing system allows us to track questions and ensure that they are responded to in a timely manner.