Server Build Guide

BigWorld Technology OSE. Released December 2014.

BigWorld Pty Ltd, Level 2, 1 Smail Street
Ultimo NSW 2007, Australia
www.bigworldtech.com

- Overview
- Requirements
- Checkout the BigWorld Technology Package
- Compiling the BigWorld Server
- Installing the BigWorld Server
- BigWorld Server Components
- Further Reading

Overview

This document describes how to configure a build environment required for building the server and associated tools along with the process for compiling the server and related components.

Unless you are performing specific modifications to the BigWorld server processes, it is recommended to use the official shipped binaries.

Requirements

Hardware Requirements

The BigWorld Server will compile on most standard "desktop" PCs.

The minimum system requirements expected for compiling the server are:

- 64 bit Intel / AMD CPU
- 512 Mb RAM

Linux Distribution Requirements

BigWorld supports compiling and running the server on four Linux distributions.

These distributions are:

- RedHat Enterprise Linux 5 and 7 (http://www.redhat.com)
- CentOS 5 and 7(http://www.centos.org)

ⓘ Please be aware that we currently have not tested and are not supporting RHEL 6 / CentOS 6.

For more information on how to install CentOS, please refer to the Server Installation Guide.

Software Requirements

All packages listed below, unless otherwise noted, are expected to be the default package installation from a RedHat or CentOS distribution. Packages from third-party repositories are not supported unless specifically mentioned.

The following software packages are required for compiling the server. Note, rather than installing each of these packages individually, you can install the `bigworld-devel` package. `bigworld-devel` includes all of the packages listed below .

- GNU C / C++ compiler (packages: `gcc`, `gcc-c++`)
- GNU make (package: `make`)

- On CentOS 5: MySQL development files (`mysql-devel`)
  On CentOS 7: MariaDB development files (`mariadb-devel`)
- Python development files (package: `python-devel`)
- Supporting libraries for Python libraries (packages: `sqlite-devel`, `readline-devel`, `gdbm-devel`, `bzip2-devel`, `ncurses-devel`, `binutils-devel`)
- SDL development files, for SDL example client (packages: `SDL-devel`, `SDL_image-devel`)

> ⓘ The SDL_image-devel package is currently not available on CentOS 7 . If you wish to use the SDL-client on CentOS 7, you will need to manually install the SDL_image package.

It is also recommended (but not required) to have the following packages available:

- GNU Debugger (packages: `gdb`)

Installing Required Software

All required packages can be installed simply by using the system package management program '`yum`'. To install a package using `yum`, you would use a command such as:

```
$ yum install <package_name>
```

For example, to install the GNU C and C++ compilers you would issue the following command as the root user, following the prompts where appropriate:

```
$ yum install gcc gcc-c++
```

Checkout the BigWorld Technology Package

You will need to checkout the BigWorld Technology package from from the official BigWorld repository. We recommend you place the source code in a regular user account (ie: not a root/privileged user account).

The directory name you checkout into is completely up to you, although we recommend naming it after your project.

Compiling the BigWorld Server

Once your build environment has been installed, compiling the server is a trivial operation.

> ⓘ Never compile the server as the `root` user.

- Change directory to your BigWorld checkout, for example:

  ```
  $ cd /home/builduser/bigworld_pristine
  ```

- Change directory to the BigWorld source code:

  ```
  $ cd programming/bigworld
  ```

- Run 'make':

  ```
  $ make
  ```

The BigWorld server source code is located in the directory `programming/bigworld/server`, with individual server components located under subdirectories.

If required, individual server components can be rebuilt by running `make` from within that component's source directory. For example in order to rebuild the DBAppMgr you could issue the following command:

```
$ cd programming/bigworld/server/dbappmgr
$ make
```

Installing the BigWorld Server

For details on how to install the BigWorld Server and related components, please refer to the Server Installation Guide.

BigWorld Server Components

Directory	Content Description
programming/bigworld/	Top level BigWorld Technology source directory.
examples/	Source code for small client and server examples.
cellapp_extension/	Examples of how to extend cell entities with C++ (EntityExtra/Controllers).
examples/client_integration/	Examples of integrating other clients with the BigWorld server.
c_plus_plus/	Example clients where game logic is in C++.
python/	Example clients where game logic is in Python.
simple/	Simple example Python client.
lib/	Top level directory for all library code.
server/	Container directory for all Server specific source code.
baseapp/	BaseApp server component (source code not available in standard packages).
baseappmgr/	BaseAppMgr server component.
cellapp/	CellApp server component (source code not available in standard packages).

Directory	Content Description
cellappmgr/	CellAppMgr server component (source code not available in standard packages).
dbapp/	DBApp server component (see also lib/db_storage_* directories for database back-end specific implementations).
dbapp_extensions/	Database specific engine drivers to be loaded at runtime by DBApp.
dbappmgr/	DBAppMgr server component.
loginapp/	LoginApp server component.
reviver/	Reviver server component.
tools/	Container directory for C++ based server tools.
bots/	Bots server process for simulating automated client connections.
bwmachined/	BWMachined daemon for server process communication and operation.
clear_auto_load/	ClearAutoLoad program to remove any any auto loading entities from the Entity database prior to startup.
consolidate_dbs/	ConsolidateDBs process for aggregating secondary databases from the cluster on startup or shutdown of a BigWorld server.
message_logger/	MessageLogger server component for receiving log messages from server components and writing them to a permanent log file.
snapshot_helper/	Snapshot helper assistant program for taking LVM snapshots of a database.

Directory	Content Description
sync_db/	SyncDB server process for updating the entity database structure to match the current entity definition state.
transfer_db/	TransferDB server process for taking and transferring snapshots of the primary and secondary databases.

Further Reading

For more information about the BigWorld Server, please refer to the following documents:

- Server Overview
- Server Installation Guide
- Server Programming Guide
- Server Operations Guide