How To Upgrade BigWorld Versions

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

Table of Contents

Chapter 1. Introduction

This document details the steps required to upgrade to the current BigWorld Technology version. Customers upgrading from earlier versions should first upgrade their assets to the previous BigWorld Technology version and only then use this document to upgrade to the latest version.

Note

For details on BigWorld Technology file formats see the document File Grammar Guide.

Chapter 2. Merging BigWorld Into Your Repository

After you have downloaded the latest BigWorld version, it will be nescessary at some point to migrate the official BigWorld release into your own repository. This chapter aims to outline some issues you may encounter in that process and how you can avoid or alleviate them.

The best approach to use for merging two branches together is to use a fresh checkout of both branches so that no changes are lost due to confusion about their origin. Ideally any modifications that are specific to your project will be located in project specific libraries or directories that are not modified by BigWorld. This will help to speed up the merge process and help to make the type of changes that you need to deal with much simpler.

2.1. Python Bytecode

If you have either committed the Python generated bytecode `.pyc` files, or have a copy of them in the directory you are merging into, please ensure that you remove these files prior to merging. As the version of Python has been updated, this will help to remove any conflicts that may arise from mis-matching files. To remove all bytecode files at once you can perform the following commands:

For Windows users:

```
del /s *.pyc
```

For Linux users:

```
find . -name '*.pyc' | xargs rm -f
```

Chapter 3. Upgrading using Asset Processor

3.1. Upgrading using the Asset Processor

The Asset Processor Python script will automatically upgrade your resource tree to the latest BigWorld Technology format. Instructions on using this script can be found in the text file `bigworld/tools/misc/asset_processor/asset_processor.txt`.

For more details about setting up the Asset Processor to deal with font related changes in the latest version, please see the chapter *File Format Changes* on page 11 .

Chapter 4. File Format Changes

This chapter describes changes made to BigWorld Technology's file formats between this version and the previous version.

4.1. XML Format

The `DataSection.asWideString` API now expects a UTF-8 string and not a Base64 string. This change is automatically done by Asset Processor to all of the GUI files. Customers which are using wide strings in other XML files and which are using this API to parse these XML files should convert their string from Base64 to UTF-8. XML values using the "!" prefix and an ASCII string will also be converted.

4.2. Font Format

The font system has been changed to utilise a glyph-caching system, rather than baking entire font sets out to a texture map. This has been added as part of internationalisation support. In particular, the Font files no longer need a `<generated>` section, and they no longer store their `.dds` files on disk.

A new rule has been added to the Asset Processor; running this will change your font files to the new format, thereby enabling the glyph cache for all fonts.

If you have any fonts that you do not want changed, for example if you are using artist-modified font textures, then add the names of the font files you wish to exclude from the process to `bigworld/tools/misc/asset_processor/FontGlyphCacheUpdate.py`

Once the AssetProcessor has been run, you will be able to see in the output log which fonts were modified. You can then delete their associated `.dds` files from your source control repository.

Chapter 5. Client Script Changes

Customers should update their Python scripts based on the below changes:

5.1. Input Handling

5.1.1. PyKeyEvent

Keyboard, mouse button, and joystick button event information is now encapsulated within the `PyKeyEvent` class. Python scripts will need to be updated in the following locations:

- The personality script event handler `handleKeyEvent` now takes a single parameter, which is a `PyKeyEvent` instance.
- `GUI.handleKeyEvent` now takes a single parameter, which should be a `PyKeyEvent` instance.
- The `handleKeyEvent` and `handleMouseButtonEvent` methods on `SimpleGUIComponent` scripts need to be adjusted in the same way so that they take a single `PyKeyEvent` parameter.
- `BaseCamera.handleKeyEvent` now takes a `PyKeyEvent` .

See the Client Python API for details on the `PyKeyEvent` class.

5.1.2. PyMouseEvent

Mouse movement events are now encapsulated within the `PyMouseEvent` class. Python scripts will need to be updated in the following locations:

- The personality script event handler `handleMouseEvent` now takes a single parameter, which is a `PyMouseEvent` instance.
- `GUI.handleMouseEvent` now takes a single parameter, which should be a `PyMouseEvent` instance.
- The `handleMouseEvent` methods on `SimpleGUIComponent` scripts need to be adjusted in the same way so that they take a `PyKeyEvent` rather than passing deltas as individual parameters.
- `BaseCamera.handleMouseEvent` now takes a `PyMouseEvent` .

See the Python Client API documentation for details on the `PyMouseEvent` class.

5.1.3. PyAxisEvent

Joystick axis events are now encapsulated within the `PyAxisEvent` class. Python scripts will need to be updated in the following locations:

- The personality script event handler `handleAxisEvent` now takes a single parameter, which is a `PyAxisEvent` instance.
- `GUI.handleAxisEvent` now takes a single parameter, which should be a `PyAxisEvent` instance.
- The `handleMouseEvent` methods on `SimpleGUIComponent` scripts need to be adjusted in the same way so that they take a `PyKeyEvent` rather than passing deltas as individual parameters.
- `BaseCamera.handleAxisEvent` now takes a `PyAxisEvent`.

See the Python Client API documentation for details on the `PyAxisEvent` class.

5.1.4. handleCharEvent

`handleCharEvent` has been removed as character events are now included as part of the `PyKeyEvent` class. Any character input logic needs to be moved to `handleKeyEvent` by checking to see if the `character` member of `PyKeyEvent` is `None` or a valid Unicode string.

5.1.5. SimpleGUIComponent.mouseButtonFocus

The `SimpleGUIComponent.mouseButtonFocus` now controls a number of events related to the mouse which was previously controlled by the `SimpleGUIComponent.focus` property. These are:

- `SimpleGUIComponent.handleMouseButtonEvent`
- `SimpleGUIComponent.handleClickEvent`

In order for these events to be called on the component scripts, be sure to set the `mouseButtonFocus` attribute to `True`, either via the scripts or within the `.gui` XML files.

5.1.6. SimpleGUIComponent.handleMouseButtonEvent

The `SimpleGUIComponent.handleMouseButtonEvent` will now only be called on one component, which is the top most component located beneath the mouse that has `mouseButtonFocus` set to True.

Previously, `mouseButtonFocus` would be called on the top most component as well as each other component that the mouse is under (whether or not it was hidden by another overlapping component).

Note

Keep in mind that `handleKeyEvent` will still be called for mouse buttons (i.e. it is treated as if it is any other key button press), as it did in previous versions (as long as `handleMouseButtonEvent` returns `False`).

5.2. Interpretation of Pitch

Previously, the interpretation of a pitch value was inconsistent throughout the BigWorld API. This has now been made consistent. A positive pitch is now consistently interpreted as nose pointing down and a negative pitch as nose pointing up. This has affected a number of APIs.

5.2.1. Direction Cursor

The meaning of `minPitch` and `maxPitch` have been swapped. `minPitch` is now the most that the pitch can point upwards and `maxPitch` is the most that the pitch can point downwards.

If the direction cursor setting are read out of `engine_config.xml`, the pitch related settings in the directionCursor section may need to be updated.

5.2.2. Cursor Camera

If the pitch of the cursor camera is now queried, it will now be the inverse of what it used to be.

5.2.3. TrackerNodeInfo

The `minPitch` and `maxPitch` arguments of `TrackerNodeInfo` have the opposite direction than the used to.

Chapter 6. Server Changes

6.1. Server Hardware Support

32 bit Linux distributions are no longer supported. Only 64 bit environments are now supported. This change was made as most hardware that has been sold recently has been 64 bit, and the increased memory availability on 64 bit servers is more suited to the requirements of the BigWorld server.

While there should be no game script related changes needed as a result of this change, any code you have implemented on a 32 bit system will need to be reviewed to ensure it works safely under a 64 bit environment.

6.2. DBMgr source code

The DBMgr source code has undergone a cleanup and refactor which has involved separating the bulk of the code into three libraries. The core DBMgr code is still located in `bigworld/src/server/dbmgr`, however the XML and MySQL implementations along with the generic DBMgr interface have been moved into `bigworld/src/lib`. Specifically the layout of the DBMgr code is as follows:

- `bigworld/src/lib/dbmgr_lib`

  This directory contains the interface required for DBMgr to communicate with specific database implementations, and vice-versa.

- `bigworld/src/lib/dbmgr_mysql`

  This directory contains the MySQL implementation for DBMgr, SyncDB and ConsolidateDBs.

- `bigworld/src/lib/dbmgr_xml`

  This directory contains the XML implementation for DBMgr.

The most significant change that occurred along with the source code moving location has been the cleanup and separation of the MySQL implementation. Most classes are now contained in their own file, and DBMgr tasks along with property data type mappings have been separated in their own subdirectories.