

Component packaging

Each developer has the reversed Internet domain name reserved for their component-names, so if your company has registered company.com, you can use com\company\ and put whatever subdirectories you wish there. For example:

```
com\company\games\thefirstgame\  
com\company\public\graphicsengine\levels\
```

All components are packed as .jar files, and there are two kinds of components.

1. Source components

Only contains .class files. The class files should be in the same package-structure as the component directories (for example com.company.thefirstgame) and should be in the same sub-directory in the .jar file (ie. com\company\thefirstgame\).

It is recommended that this .jar file is saved in the same structure as the base package of it (com\company\thefirstgame\code.jar if the base package for the component is com.company.thefirstgame), but this is not required.

2. Binary components

Contains files of any kind, and represents a directory including all sub-directories. For example, it could contain the data for a level (level.map) and be called level1.jar. When this component is downloaded, it is unpacked to a directory of the same name as the component, so if the component was called com\company\level1.jar, the GF client will unpack the contents of level1.jar into com\company\level1\.

When the component has been downloaded and unpacked, the developer can use them as normal files.

Binary components can be marked as persistent to prevent deletion of them when the GF client is closed. If they are marked as persistent, the GF client will also check if the components already exist before downloading it, and abort the download if it does.

A GameDescriptor can contain a list of several components. If it contains several source components, they will all be loaded into the same classloader, so different developers can share components in an efficient manner. If it contains several binary components, they will be unpacked as normal.