Archived:BamFS

Archived: This article is archived because it is not considered relevant for third-party developers creating commercial solutions today. If you think this article is still relevant, let us know by adding the template {{ReviewForRemovalFromArchive|user=~~~~|*write your reason here*}}.

Tool No longer available

BamFS 1.1

BamFS - Bambalam J2ME filesystem merger

Usage: BamFS [-options] directory outfile.bfs [outfile.java]

Options:

```
-n Natural sort files by filename
-e:*.ext,*.ext Exclude files with extensions
-p:Package Specify java package name to be used
```

- What is it?

BamFS is a tool to merge all your J2ME data resources into a ".bfs" data pack file. It also produces a static java class source file that you include in your project to get access to the merged files.

- Why?

Having your J2ME resources packed in one big datafile greatly improves your Jar Compression rate.

- How do I use it?

Run BamFS with your resource directory and an outfile as arguments and it will produce a datapack file and a java source file. Copy the datapack ".bfs" file to the root of your java source directory and copy the java source to your package source directory (if you have a package structure, you can state a package to use using the -p:package option).

Then you can simply access your files by using the static functions included in the generated java file.

How do I write the code to load data or images from the packed resources?

Lets say you named your BamFS class to Files. Then you can use:

byte[] Data = Files.getFile(Files.LEVELDATA_BIN); (the LEVELDATA_BIN constant is the generated id for the original file "leveldata.bin")

Or you can load images directly: Image boat = Files.createImage(Files.BOAT_PNG); (loads the packed png "Boat.png" into a new image called boat)

You can also access the packed resources as inputstreams using the getFileAsStream(int FILE_ID) function.

If you're loading a lot of resources at the same time, you can speed up the process by preloading the whole data pack into memory. Do it by calling "Files.preload_files()". Don't forget to free the preloaded pack when you're done loading your data by calling the "Files.free_preloaded_files()" function.

- I still don't get it!

Check out the example source for a midlet using the BamFS filesystem included!

BamFS was created using Bamcompile - a great tool to make windows utilities using PHP.

(courtesy : Anders Hammar, supremej2me@bambalam.se)

(C) Copyright Nokia 2013. All rights reserved.

(C) Copyright Nokia 2013. All rights reserved.