

# **.Pak Resource File Support (PakLib)**

## **Overview:**

The goals of the .pak file format are to:

1. Reduce game loading time
2. Reduce install/uninstall time
3. Make it harder to 'rip' resources from games

A .pak file is a single file that contains images, sounds, music, and most other files. Flash files, however, can not be included.

Pak support is implemented through "p\_" prefixed versions of most familiar file functions such as *fopen*, *fread*, and *FindFirstFile*. Those functions first process through any loaded pak files and then through loose files. If you have file routines built into your game or libraries that you use that make use of file functions not covered by PakLib, you will have to make "p\_" prefixed versions of them.

## **Usage:**

Include "*PakInterface.h*". If you examine the header file, you will see a list of all the "p\_" prefixed functions. Please see the **Reference** section for a full list. Instead of using, for example, *fopen* and *fclose*, you would use *p\_fopen* and *p\_fclose*. This would search all loaded .pak files first for the file. In the event that the file is not found, the standard library implementation (which operates over loose files) will be used, just like usual.

"*PakInterface.h*" externs a *PakInterface\** global named *gPakInterface*. By default, SexyAppBase attempts to load "*main.pak*", which will silently fail if not found. You can add whatever .pak files you want by calling *gPakInterface->AddPakFile(<filename>)*. The method returns *true* on success. The file will be immediately loaded.

To create your .pak file, you need to use the PopPak.exe file located in the *tools* directory. Copy whatever files and directories you want to pack up to a central location. Run PopPak.exe with the /P option and pass in the .pak file to create followed by the directory containing files to pak up. All subdirectories will be included. For example, if your files to pack up resided in C:\paktest\files, and you wanted to create a file named *main.pak*, you would type:

```
PopPak.exe /P main.pak C:\paktest\files
```

You can unpack a file via the /U option.

## Reference:

PakLib has custom versions of most file routine functions that search through any loaded .pak file first, and then loose files (just like normal) if the file requested isn't found.

Functions in **bold** are Windows system-specific functions. The rest are C standard library routines. For reference, the following is the list of "p\_" prefixed functions:

<b>PakLib Function Name</b>	<b>Equivalent Standard Function Name</b>
p_fopen	fopen
p_fclose	fclose
p_fseek	fseek
p_ftell	ftell
p_fread	fread
p_fwrite	fwrite
p_fgetc	fgetc
p_ungetc	ungetc
p_fgets	fgets
p_feof	feof
p_FindFirstFile	<b>FindFirstFile</b>
p_FindNextFile	<b>FindNextFile</b>
p_FindClose	<b>FindClose</b>