

[Home](#) >> [PSP Tutorials](#) >> Tutorial m0: Convert FW 1.5 code to FW 3.x code

This is miscellaneous PSP tutorial m0



## M0: Convert FW 1.5 code to FW 3.x code

Hi everybody,

In this tutorial I will discuss the ways of making 1.5 firmware code work on firmware 3.xx. It will not be a very big tutorial but it will be a great one for those with 3.xx firmware. There will be no source or example files. This tutorial has not been made by me but is rewritten by me. I got the tutorial and help from CpuWhiz at the ps2dev forums. So some of the parts can be directly quoted from him. All credits of this tutorial are due to him. The tutorial has 4 parts.



### Part 1: Makefile

Since we create the game/app for a different firmware we have to alter the Makefile. The 3.xx firmwares needs the game/app to be a prx. Here is what we do first:

```
BUILD_PRX = 1
PSP_FW_VERSION = 371
```

These two has to be inserted in the makefile anywhere before the "include \$(PSPSDK)/lib/build.mak" line.



### Part 2: Usermode

Kernel mode only works on firmware 1.5 so we cannot run in kernel mode. The game/app needs to start in user mode. As you have seen in tutorial 0: [setting up the 3D environment](#), I did not discuss the second parameter of the PSP\_MODULE\_INFO function. The second parameter is used to set the kernel or user mode.

```
PSP_MODULE_INFO("3.xx homebrew", 0, 1, 0);
```

With setting the second to 0, we set it to user mode. The last two parameters are to specify the version. This line of code needs to be placed at the top of the main.cpp (or main.c) just like in the tutorials.

This alone is not enough. We also have to set our heap size. CpuWhiz sets his to 20 mb:

```
PSP_HEAP_SIZE_KB(20480);
```

If you want more information about the heap read: [Definition of Heap](#). This line of code needs to be placed below the PSP\_MODULE\_INFO line of code. If you have a compiled PSP SDK from on or after Sep. 30th, 2007 you can also use this function which sets the heap to the maximum size possible:

```
PSP_HEAP_SIZE_MAX();
```

This will allocate as big of a heap as it can. Please note you should recompile your entire toolchain (or at least pspsdk and newlib) to use

this, otherwise your homebrew will crash with a Exception - Bus error (data).



## Part 3: Try running your game/app

With the above alterations you should be able to compile your app the way you always do and copy over the EBOOT.PBP. There is one difference and that is that you no longer have a % folder for the 3.xx firmware. If you are running your application from psplink, you need to run the prx file instead of the elf file or it will not run. At this point your homebrew should run unless you have kernel calls in your code. If your code has kernel calls you will get a 0x8002013C error when you try to start the game/app. Don't panic, move on to part 4. If your homebrew runs, great, skip part 4.



## Part 4: Kernel calls

You need to figure out what is a kernel call and what isn't. To do this, you can use prxtool -f . Here is a example output:

```
$ prxtool -f project.prx
... output left out (it's a lot of output) ...
Import 9, Name UtilsForUser, Functions 1, Variables 0, flags 40010000
Functions:
0x79D1C3FA [0x0008CF34] - UtilsForUser_79D1C3FA
Import 10, Name LoadExecForUser, Functions 2, Variables 0, flags 40010000
Functions:
0x05572A5F [0x0008CF3C] - LoadExecForUser_05572A5F
0x4AC57943 [0x0008CF44] - LoadExecForUser_4AC57943
Import 11, Name IoFileMgrForKernel, Functions 1, Variables 0, flags 00010000
Functions:
0x411106BA [0x0008CF4C] - IoFileMgrForKernel_411106BA
Done
```

If you look at the above output you can see there is a import called IoFileMgrForKernel. This import has one function. Refer to this page: <http://silverspring.lan.st/1.5x/kd/iofilemgr.html>. Search on the page and you will find that 0x411106BA matches the function sceIoGetThreadCwd. You can now search for this function and either (a) replace the kernel call with user mode code -or- (b) move the kernel call into a kernel mode prx and load that kernel mode prx from your homebrew. Option B is out of the scope of this tutorial so search the [ps2dev forums](#) to figure out how to do this. Option A is the preferable solution unless you have to use a kernel call. I went to the main page <http://silverspring.lan.st> and clicked on 1.5x firmware. From this page I found sceIOFileManager in the list and clicked on it to get to the above page. A search function would have been nice.

After this the game/app should work fine. We now have converted our game/app to be able to run on 3.xx firmware versions.