

Dear Demo Playable Section developer,

This directory contains bs, a simple bootstrap program based on the one for the full demo disk. There is a makefile for making bs.exe, the source for bs, in case you need to make temporary changes (temporary, mind) and an example cti file showing how to lay out an example program on the emulator to be loaded by and run by bs. Incidentally, bs uses bs.lib, a specially built subset of the full libraries. Bs.c and bs.lib are copyright SCEE.

What does BS do ?

Basically, it boots the machine, displays a crappy icon, and then loads and runs the first executable on the CD, ie the one at track 0, minute 0, second 2, sector 35. By putting your program first on the CD or first in the CTI file, bs will load and run your program. This is how you should test your playable section - call bs.exe psx.exe, and put your game, eg ridge.exe, as the first program on the disk. This is not the way the demo disk will work; the main demo menu program will be the first program on the disk, and it will find the logical position of your game (its own way) and then use bs to load it. But for test purposes, just make bs load your main program file. If you can't do this for some reason, get back to me, or temporarily alter the bs source.

If your code is all ok, and all is well, the dev system should printf to the host:

```
<Some stuff about the bs heap, data locations and so on>
<Followed by the stuff from PadInit()>
<and lastly:>
```

```
BS: Set loc for seek
BS: Program go.
```

At this point, BS has seeked to your program and is loading it. Immediately after this, your program will be run. If something goes wrong, BS will timeout and reboot the machine. This will probably cause repeated reboots, but you never know. Now your game should be running.

The executable will be loaded according to the XF\_HDR data at the start of the .EXE, ie it should have correct info in it or all will be lost; the program is loaded to the address specified in the t\_addr field, and the bss is cleared.

\*\*\* Important bit \*\*\*

The stack pointer is not set properly by cpe2x. However, bs cannot assume where you would like your stack to go ! So, I have provided a program to allow you to set the stack pointer (in the directory setsp). Just type:

```
setsp <my game exe> <mystack address>
```

The address is the top of the stack; remember the stack goes downwards. You should probably put this in your make file. Now bs will know where to put your stack. If you forget this, your code will probably crash as the stack address is usually set to a rubbish value.

\*\*\* oh no ! \*\*\*\*

Once your game stops, the development system should printf:

BS: Returned ok; load Game: 0.

Followed by it loading your game again - ie it loops through the above.

Hopefully:

- A) it should work
- B) if your code is linked with none2.lib (see readme), it should return to BS
- C) Being linked with none2.lib doesn't fuck up your code (very possible)
- D) It should run more than once

Of course, if you write inside 0x80010000 -> 0x80018000, you'll break BS.

Obviously, this isn't as simple as it seems. :-) But it does work, promise.

If you have problems, etc, you know how to reach me, or try the technical support guys.

Allan.

Update 08 October 97 (Dave V)

Included in this directory are old\_bs.lib and new\_bs.lib. You don't really want to use the old\_bs.lib as it's shit if you want to use libcd from a later lib than 3.4. Link the bootstrap with the new\_bs.lib if your demo uses libcd from a later lib than 3.4.