Epsilon BIOS v1.0

Introduction

Epsilon BIOS is a custom flash "replacement" for the Sony PSP which unleashes the full potential of your handheld, allowing you to both use homebrew software and run UMD ISO games from your memory stick on the latest firmware releases while also enjoying the impressive features built into the operating system such as RSS feeds, WMA support etc. Currently the 2.71 firmware release is supported.

It is important to note that Epsilon BIOS is not standalone firmware replacement but more like a bootloader. It works using the dual-firmware system provided by the Undiluted Platinum hardware modification by loading when the PSP is initially powered on, then once running executes and "piggybacks" the real firmware stored in your PSP flash memory. Due to the way this works it is **NOT** possible to use Epsilon BIOS unless your PSP has an U.P. hardware modification installed.

Features

- Runs alongside 2.71 firmware, so you get all the features of 2.71 such as RSS feeds, web browser etc combined with the advantages of homebrew software and ISO loading.
- Allows execution of homebrew software in kernel mode, removing all limits previously in place while running homebrew on firmware versions above 1.50.
- Ultra reliable, near transparent UMD emulation allowing users to run their games from a memory stick with ease, including games which require 2.0+ firmware without rebuilding the ISO or relying on nasty hacks.
- Support for compressed ISO files for UMD emulation, allowing you to fit more games onto your memory stick at one time.
- Built-in recovery menu which can be used to to update your Epsilon BIOS installation or restore your PSP flash contents if it becomes "bricked".

Functional Description

As described above, Epsilon BIOS can be compared to a "bootloader". It is stored on the U.P. flash memory and executed when your PSP is powered on. Once Epsilon BIOS is running it loads and "piggybacks" the real firmware from the PSP flash. Currently only 2.71 firmware is supported, you must upgrade your PSP onboard firmware to 2.71 in order to use Epsilon BIOS. If you attempt to boot Epsilon BIOS with an unsupported firmware version you will be taken to the recovery menu where you can upgrade the PSP firmware. The great thing about Epsilon BIOS is that since the "bootloader" always runs before the real firmware we can apply whatever patches are necessary to disable **whatever protection Sony tries to add in the future**, making a U.P. modified PSP using Epsilon BIOS very future proof.

With Epsilon BIOS you can run homebrew software (EBOOT files) directly from the OS main screen. Kernel mode applications are also supported meaning there are no limits when it comes to homebrew software. Both 1.00 and 1.50 style EBOOT's are supported.

UMD emulation is handled almost transparently. You do not need to launch a separate application in order to load your UMD games from the memory stick, all the ISO's you have stored on your memory stick are listed alongside your homebrew applications in the "Game->Memory Stick" screen and executed from there. Copying new games to your memory stick is easy, simply enable the USB connection and copy ISO's to the "ISOS" directory in the root of your memory stick. Please note that each time you change the contents of the ISOS directory a cache file containing the icons etc for each game must be updated, and this will cause a slight delay while viewing the "Game->Memory Stick" screen.

Compressed ISO's are supported for UMD emulation using our own custom format, "Epsilon ZIP". Using the "Epsilon ZIP Tool" included in the archive you can convert UMD ISO files into EZIP files and vice versa. Simply copy the EZIP files into the same location as normal ISO files in order to play them.

Epsilon BIOS includes a recovery mode which can be used to update the PSP onboard firmware, restore a bricked PSP or upgrade your Epsilon BIOS installation. If there are any problems while booting then you will be presented with the recovery menu. To forcefully enter the recovery menu, hold SELECT+START when you power on the PSP.

Installation

Epsilon BIOS is broken into two distinct parts: the bootloader and the core. The bootloader is programmed to U.P. flash from the PC and is what actually takes control of the system when you first power on your PSP. The bootloader attempts to load the core installed in U.P. flash; if the core has not yet been installed or there is another problem you will be taken to the recovery menu. When you first program the Epsilon bootloader onto your U.P. you will need to install the core separately since it is not bundled inside the bootloader flash image. However, whenever an Epsilon BIOS core update is released you simply copy the update file onto your memory stick then use the recovery menu to update the core. This system is much safer and more user friendly than having to reprogram the U.P. flash from the PC each time you update which would be required if the bootloader and core were integrated.

Installing the Bootloader

1. Turn on the PSP while holding LEFT to enable U.P. programming mode
2. AFTER the PSP has turned on, connect the USB cable to the U.P.
3. Program the bootloader flash image (epsilonBootloader*.flash) to U.P using the flashing tool
4. **Cold restart** the PSP by cycling power

Installing/Updating the Epsilon BIOS Core

Updates and the initial installation of the Epsilon BIOS core are handled through the Epsilon recovery menu. To enter the recovery menu hold SELECT+START while you power on the PSP. The procedure to install/upgrade the BIOS is as follows:

1. Copy EBUPDATE.BIN to the root directory of your memory stick. This can be done via the recovery menu by selecting "Memory Stick USB" or with a card reader, PSP XMB etc
2. Select "Update Epsilon BIOS", then hit X to confirm
3. Once the installation/update is complete the PSP will power off.

Notes Regarding Homebrew

As the majority of homebrew software currently available is designed to run on the 1.50 kernel we decided that for compatibility reasons it would be best to have Epsilon BIOS load the 1.50 kernel instead of 2.x when running homebrew software. This is possible since the Epsilon BIOS bootloader is actually based on the 1.50 firmware so when running homebrew software the kernel is loaded from U.P. flash rather than PSP onboard flash. The only known issue with this method relates to wireless network configuration - since the 1.50 kernel does not support WPA encryption you will need to configure your PSP to use WEP if you wish to use WIFI enabled homebrew software.

Recovery Menu

To forcefully enter the recovery menu hold SELECT+START while you power on the PSP. The Epsilon BIOS recovery menu from the 1.0 bootloader has the following menu selections:

1. **Memory Stick USB** – Enables the USB connection between the PSP and PC for transferring files. This is the same as the USB connection in the PSP XMB.
2. **Update Epsilon BIOS** – Installs an Epsilon BIOS core update from the memory stick. The update file must be named EBUPDATE.BIN and placed in the root directory of the memory stick.
    1. **Launch Firmware Updater** – Launches an official Sony firmware updater EBOOT stored on the memory stick at /PSP/GAME/UPDATE/EBOOT.BIN. You may use this feature to both upgrade and downgrade your PSP **onboard** firmware version. Please take note first of the following important facts:
        1. Epsilon BIOS releases are only compatible with certain firmware versions. If you flash your PSP to an unsupported version you will be taken to the recovery menu by the bootloader until such time as a supported firmware version is installed.

    2. In order to downgrade your firmware certain files must be modified in your PSP onboard firmware so the official Sony updater *thinks* you have a very early firmware version installed. There are always risks involved with modifying your onboard firmware and such an action can be potentially hazardous to your data. Please note that when upgrading the firmware version no files need to be edited so this warning does not apply.

1. **Restore onboard NAND** – Used to “un-brick” a PSP by programming a known good flash dump (such as a dump of 1.0 or 1.50 firmware) to your PSP onboard NAND flash. The flash image must be a file called “nandImage.flash” in the root of the memory stick, in the same format used by the UP flasher tool (512bytes user + 16bytes extra for each page, interleaved).
2. **Shutdown PSP** – self-explanatory :)

Version Information

Epsilon BIOS v1.0 - Bootloader v1.0, Core v1.0 Required onboard firmware: 2.71

FAQ

Q: Why do I get a 0x86660000 error when I try to launch a Sony firmware upgrade?

A: Epsilon BIOS blocks you from trying to install firmware versions that are not supported by the installed core. If you wish to upgrade to a new firmware, you might first need to upgrade Epsilon BIOS to a version which supports said firmware.

Q: From firmware 2.6 and up, PRX files are protected with a new encryption method. How did you figure out how to decrypt these files?

A: As most people will now be aware, the discovery of the 2.6 kmode exploit lead to decryption of modules using the new encryption method. However when we started working on this the kernel mode exploit was unknown so we took a different approach to reach our goal, one that doesn’t rely on exploits so should allow us to easily hack new firmware releases in the future once Sony changes the encryption method again. Here’s how we did it - warning: this is a bit technical, which unfortunately is required to give a proper answer. Since we couldn't get a dump of kernel memory from a PSP running the 2.6 firmware, the only way to figure out how to decrypt the 2.6 PRX files was to disassemble the IPL and see how this decrypted the files while the PSP is booting. Unfortunately, Sony used a clever trick in the 2.6 IPL to prevent hackers disassembling it. They read out some data from the reset vector and use it to decrypt the main portion of the IPL code. The problem here is that by the time we can run code on the PSP, any attempt to read out this data will be in vain as it gets scrambled inside the IPL. However, through some hardcore trickery we found a way to dump the data at the reset vector which enabled us to decrypt the main portion of the IPL code and then use this to figure out how the 2.6 PRX files were encrypted. The same encryption method and keys are used in 2.7 and 2.71, so when 2.7 came out we had this dumped and decrypted very quickly. There is nothing left now they can use to hide the IPL so when the 3.0 firmware eventually comes out its highly likely the encryption will have changed again but it shouldn’t take too long to figure it out. Sorry to give you the bad news Sony.. the hackers win another round, you cannot hide your firmware from our eyes anymore ;)

Greetz/Thanks

- malloc for originally finding that 1.0 PSP's could run unsigned code
- nem for his Hello World demo, the FIRST real homebrew PSP program
- Undiluted Platinum team for their modchip
- adresd, chip, mrbrown, ooPo, rinco, Tyranid, Warren and everybody else involved in the development of PSPSDK.
- Fanjita and Ditlew for their work on the 2.0+ EBOOT loaders
- Edison Carter for his GTA cheat device which also revealed a way to run unsigned code on firmware above 2.0
- JiniCho, Jonny, malloc, Raphael, argandona and Optixx for the many incarnations of PMP
- YoYoFr, Laxer3a, and Thunderz for snes9x TYL
- Shine and Nevyn for LuaPlayer
- Humma Kavula for UMD Emulator
- The many other people creating awesome homebrew software for the PSP

For some awesome PSP related websites, check out:

- ps2dev.org
- maxconsole.net
- psphacker.com
- psphacks.net
- psp-news.dcemu.co.uk
- psp-spot.com
- pspupdates.qj.net
- Xavboxpsp.com