

IdStorage Manager v1.3 by Chilly Willy

=====

This is my latest attempt to make dealing with the idstorage keys easier, as well as provide extra features. Currently, this app will dump all existing keys, verify the keys against files on the memory stick, display individual keys in hex or ascii, and edit the values of individual keys.

Installation

=====

To install IdStorage Manager on a PSP with custom firmware, make a directory in the GAME (or GAME150 for custom firmwares) directory called idstoragemgr (actually, this name doesn't matter at all) and copy EB00T.PBP, idstorage.prx, and textcolors.bin.

Usage

=====

To use Id Storage Manager, put any key files you wish to verify or write into the idstoragemgr/keys directory. These files must have a file name of the form 0x0041.bin, where the "0041" is the key number in hexadecimal. If you dump the keys on the PSP, the files will be dumped to that same directory in the format described. If you don't understand what I mean, just dump the keys and look at the resulting filenames.

Run IdStorage Manager from the XMB like any other homebrew. You'll see a title screen for a few seconds, then be taken to the status screen. Note that the status screen currently just shows the motherboard, region, and model.

From the status screen, you can perform the other IdStorage Manager functions by pressing the appropriate button. Button mappings are printed at the top of the display. Press the Circle button to exit back to the XMB. Press the Square button to dump the keys to your memstick. Press the Triangle button to view/edit individual keys. Finally, press the Cross button to verify and fix the keys using files on the memstick.

The first time you dump the keys, it creates a folder called 'keys', then dumps the keys into it. If you choose to dump the keys again, the 'keys' folder will be renamed, and a new 'keys' folder created. You can keep dumping the keys until you have dumped them 10,000 times, or you run out of memory on the memstick.

While viewing the individual keys, press the Triangle button to switch the view between hexadecimal and ASCII. Press the Circle button to return to the status screen. Press Right/Up/RTrigger to advance 1/16/256 keys, and Left/Down/LTrigger to go back 1/16/256 keys. Pressing Cross to edit yields another input prompt: you are asked if you wish to cancel, change a byte, delete the current key, or create a key at the current index (the created key will be cleared initially).

When you select to change a byte, the screen changes to the editor display. You'll see the hex representation of the key, and the offset of the byte being edited. Use the D-Pad to change which byte is being edited. Note that there is no cursor (yet). Be sure to keep track of the offset value. Pressing SQUARE adds 16 to the byte being edited, pressing CROSS subtracts 16 from the byte, pressing TRIANGLE adds 1 to the byte, and pressing CIRCLE subtracts 1 from the byte. Press SELECT to exit without saving, and press START to exit after saving the key. You will return to the view key screen.

While verifying keys, it will print which key is being checked and whether it passes or fails the comparison against the file on the memstick. If it fails, it asks you to press the Circle button to skip over this key (hence leaving it alone), or to press the Cross button to write the contents of

the file to the IdStorage key. Once it is done verifying, it will return to the status screen. If you used this to fix keys, you can press the Cross button again to verify the keys, ensuring the fixed keys were written and pass the verification this time.

This version of IdStorage Manager will load the background and text colors from the file "textcolors.bin". If the file is missing, it will default to the old color scheme - white text on a black background. The new colors in the textcolors.bin supplied are white text on a warm blue background. The way to change the colors is to load textcolors.bin into a hex editor and change it as follows:

Byte 0: 00
Byte 1: red intensity for background (00 to FF)
Byte 2: green intensity for background (00 to FF)
Byte 3: blue intensity for background (00 to FF)
Byte 4: 00
Byte 5: red intensity for text (00 to FF)
Byte 6: green intensity for text (00 to FF)
Byte 7: blue intensity for text (00 to FF)
Byte 8: 00
Byte 9: red intensity for alternate background (00 to FF)
Byte 10: green intensity for alternate background (00 to FF)
Byte 11: blue intensity for alternate background (00 to FF)

For example, the contents of the default textcolors.bin is:
00 20 60 A0 00 FF FF FF 00 40 90 D0

Warning

=====

Although I've tried to make this program bug-free and safe to use, it has the possibility of bricking your PSP if you don't use it correctly. Viewing and dumping keys should never cause a problem, but verifying/fixing/editing keys can brick the PSP easily if you don't know what you are doing. Even then, there is a remote possibility the PSP could be bricked by fixing keys. Nothing is 100% safe.

Acknowledgements

=====

This program was made possible by the efforts of many people. I'd like to thank Stapol, Mathieulh, harleyg, Jas0nuk, and Dark_AleX. If anyone else thinks they deserve some credit, let me know. :)