

### Address calculation:

#### Get PSX-EXE/SLUS/SLES address from RAM:

1. Enter the address of the RAM in the field "In->Address".
2. If a game uses the standard index, to the "Index list->Select index", leave "Standard PSX-EXE Index".
3. Press the button "Calculate", in the field "Out->Address" we get the address in an executable file(PSX-EXE ).

Example:

The screenshot shows a software interface for address calculation. It is divided into several sections:

- In:** Contains an 'Address' field with the value '8006203C' and a 'Paste' button. Below it is an 'Instruction' field with the value '0'.
- Out:** Contains an 'Address' field with the value '5283C'. To its right are three buttons: 'Calc', 'Calc & copy', and 'Copy'.
- Operations:** A section with four radio button options:
  - ☒ Get PSX-EXE address from RAM
  - ☐ Get RAM address from PSX-EXE
  - ☐ Get RAM address from load/store instruction
  - ☐ Get LBA from CD image address
- Index list:** Contains a 'Select index' dropdown menu currently showing 'Standard PSX-EXE index'. Below the dropdown are two buttons: 'Delete index' and 'Save index'.

#### Get RAM address from PSX-EXE/SLUS/SLES):

1. Enter the address of the data found in an executable file, to the field "In->Address".
2. If a game uses the standard index, in the "Index list->Select index", leave "Standard PSX-EXE Index".
3. Press the button "Calculate", in the field "Out->Address" we get the RAM address.

Example:

This screenshot shows the same software interface as the previous one, but with different settings:

- In:** The 'Address' field now contains '324BC'.
- Out:** The 'Address' field now contains '80041CBC'.
- Operations:** The radio button for 'Get RAM address from PSX-EXE' is now selected (highlighted with a green dot).
- Index list:** The 'Select index' dropdown still shows 'Standard PSX-EXE index'.

Get RAM address from load/store instruction:

For example, take the instruction: **LH v1, 1A6(v1)**

**v1** – instruction address. ( **v1 = 800385CC**)

**1A6** – instructions index.

1. Enter the address of the instruction to the field “In->Address”.
2. Enter the “instructions index” to the field “In-> Instruction”.
3. Press the button “Calculate”, in the field “Out->Address” we get the RAM address.

*Note: support negative indexes SW v0, -5DA8(s3).*

Example:

The screenshot shows a software interface with two main sections: 'In' and 'Out'. In the 'In' section, the 'Address' field contains '800385CC' and the 'Instruction' field contains '1A6'. In the 'Out' section, the 'Address' field contains '80038772'. Below these sections is a 'Operations' section with four radio buttons: 'Get PSX-EXE address from RAM', 'Get RAM address from PSX-EXE', 'Get RAM address from load/store instruction' (which is selected), and 'Get LBA from CD image address'. At the bottom is an 'Index list' section with a 'Select index' dropdown menu showing 'Standard PSX-EXE index', and 'Delete index' and 'Save index' buttons.

Get LBA from CD image address:

1. We find the required data in the disc image at: \$1a005c
2. Enter the address to the field “In->Address”.
3. Press the button “Calculate”, in the field “Out->Address” we get the LBA of file(in the decimal system) which contains the required data .

*Note: The resulting LBA to be found in the interval LBA files contained on the disc.*

Example:

The screenshot shows the same software interface as above. In the 'In' section, the 'Address' field contains '1A005C' and the 'Instruction' field is empty. In the 'Out' section, the 'Address' field contains '724'. In the 'Operations' section, the radio button 'Get LBA from CD image address' is selected. The 'Index list' section remains the same with 'Standard PSX-EXE index' selected in the dropdown.

The received LBA is in an interval LBA 647 – LBA 819,

Name	Size	LBA
DATA		1 303
MOVIE		139 796
ZZZZ		301 761
SYSTEM.CNF	68	23
SLUS_000.00	174 080	24
LEVEL.BIN	591 120	109
LEVEL2P.BIN	509 472	398
SHELL.BIN	351 912	647
FAILURE.TIM	245 780	819
RETURN.TIM	245 780	940
SUCCESS.TIM	245 780	1 061
VICTORY.TIM	245 780	1 182

It means the required data contain in the file "SHELL.BIN".

#### Getting index:

If in game uses non-standard index, we can get them from PSX-EXE(SLUS/SLES).

1. Press the button "Get index".
2. Choose PSX-EXE (SLUS/SLES).

If index calculation finished successfully, input fields (index name and user index) will be filled automatically.

Example:

User index

Index name

SLUS\_007.57

User index

17800

Get index

Clear index

Add index

### Searching index:

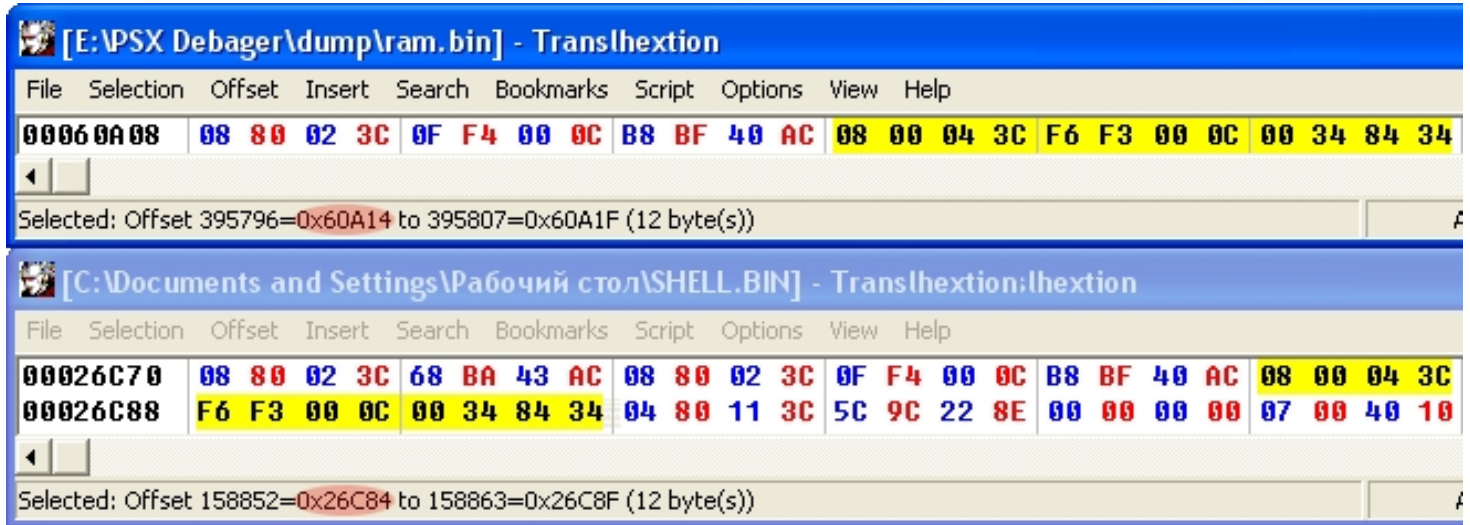
Not always, all code of game contained in the executable file PSX-EXE(SLUS/SLES).

If the necessary code located in other file, it need to calculate a non-standard index.

Otherwise converting RAM <-> PSX-EXE will be incorrect.

To get a non-standard index, you need to find the identical data in the executable file and RAM.

Then subtract data address in the RAM, from data address in executable file.



Data address in the RAM: \$60A14

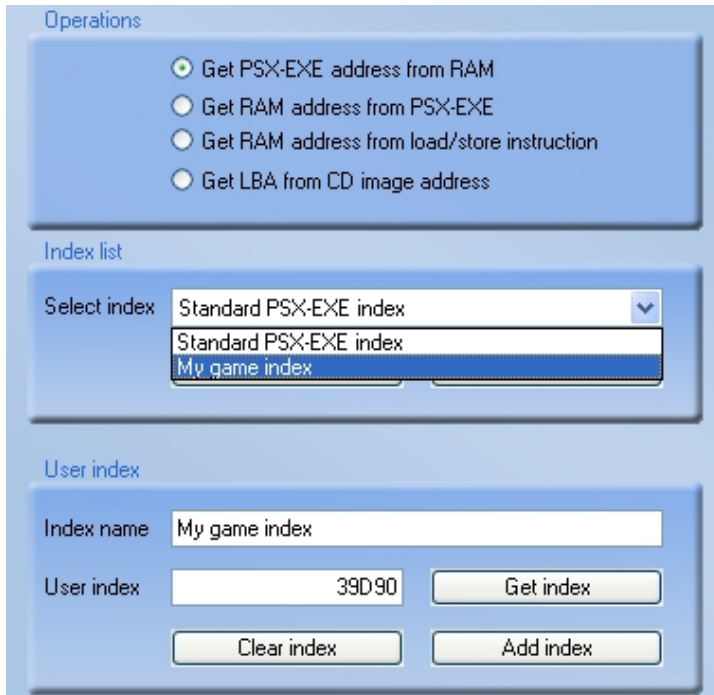
Data address in executable file: \$26c84

$\$60A14 - \$26c84 = 0x39D90$

Index: 0x39D90

### Using index:

1. We write our index to the field "User index".
2. Press the button "Add index".
3. Select index in "index list ->Select index".
4. If you want to retain the index, push "Save index" button.



*Note: You can open index editor by the minimize button, on window system menu.*