==
T H E " U N - O F F I C I A L"
PLAYSTATION DEVELOPMENT FAQ

PSYQ
CONFERENCE
==

Release v1.2
Last Updated: February 29, 1996

DISCLAIMER

This FAQ is to aid in informing the licensed game developer about the development environment provided by Sony Computer Entertainment.

The Development System Tool to which this manual relates is supplied pursuant to and subject to the terms of the Sony Playstation Licensed Developer Agreement.

This FAQ is intended for distribution to and use only by Sony PlayStation Licensed Developers in accordance with the Sony Playstation Licensed Developer Agreement. The information in this manual is subject to change without notice.

The content of this manual is Confidential Information of Sony for the purposes of the Sony PlayStation Licensed Developer Agreement and otherwise.

TRADEMARK INFORMATION

PlayStation and Sony Computer Entertainment names and logos are trade names and/or trademarks and/or copyright artwork of Sony Corporation(or its subsidiaries).

All specific names included herein are trademarks and are so acknowledged: IBM, Microsoft, MS-DOS. Any trademarks not mentioned here are still hypothetically acknowledged.

COPYRIGHT NOTICE

[1.] PSYQ TOOLS

[1.1.]: When compiling a sample, an error occurs with the message, "Can't find how to execute command.".

The following cases may cause the error. Please check them.

* No setting of the environment variable COMPILER_PATH
* A path doesn't lead to the directory where commands eside.
* "\" is used for separating the directories in the environment variable setting.

("/" must be used for the directory separation in PSYQ tool.)

[1.2.]: Why is a PC hung up when starting up the debugger?

If the screen becomes a blackout, the cause will be that the PC is started up without '/r50' option in Japanese mode. Be sure to give '/r50' option if
using in Japanese mode. In addition, '/e' option is also needed.

[1.3.]: How do I view global variables witin a watch window of the debugger?

Hit <alt> g while in the watch window.

[1.4.]: Is there a way to break into a running program within the debugger?

Yes. hit the ESC key. The program will break at the closest pollhost() call. If you don’t have pollhost() calls in your program, it won’t work.

[1.5.]: How do I reload the program and reset the machine from within the debugger?

Hit <alt> R, then load the game using the F10 key, and the ‘download’ item in the file menu.

[1.6.]: How many modules can I link with psylink?

By default the linker allows up to 256 object modules to be linked. Each object module taken from a library counts as 1 module and so making one library with all the object files in it doesn't help. The way to get round the problem is to use the /n switch on the linker command line in order to increase the number of modules that can be linked.

e.g. to increase to 300 modules enter

psylink /n300 .....
if psylink is being called directly or

ccpsx -Xn300 .....
if ccpsx is calling psylink.

[1.7.]:Global register allocation?

It is possible to reserve a register to hold/point to a global variable throughout all the modules of your code. This obviously makes access to the variable very fast but there are a limited number of registers available.

One way to use this on the playstation is to use a register to point to a structure stored in the high speed data cache memory. Accesses to elements of this structure will then be made by displacements from this register rather than by loading the absolute address of the variable. This is similar to the way that the global pointer register optimisation works except that it gives you control of exactly which variables are placed in this memory.

The register used should be one of the saved registers $16 - $23 (s0-s7). If you are not using the GP register optimisation and you specify -G0 when compiling your code then you could use register $28 (gp) for this purpose.

Here is an example :

```
struct fast_data
{
      int i;
      char z[16];
};

register struct fast_data *f asm("$23");

void main ()
{
   ...
   f->i = 3;
   ...
}
```

Here the assignment f->i = 3 will code as

```
    li    $8,3
    sw    $8,0($23)
```

rather than

```
    li    $8,3
    sw    $8,f
```

where the sw instruction expands to

```
    lui   $1,(f+$8000)>>16
    sw    $8,f&$ffff($1)
```

Points to note :

You can't generate a pointer to something held in a register.

It is up to the programmer to initialise the register. This could be done as follows :

```
void main()
{
    asm("li $23,0x1fff0000");
```

or whatever.

The compiler will not make any other use of the register in modules in which the declaration appears. In modules that were compiled without the declaration (e.g. libraries) the compiler may have generated code that makes use of the register. This is not generally a problem since the s0 - s7 registers must be saved and restored by any function that makes use of them. However, the register may not be set correctly in an event handler or call back or any other function which is called through a pointer by code that was not aware of the global register allocation (e.g. library code). It will therefore be necessary to reload the register if the variables need to be accessed in these type of routines. This is similar to the problem with the global register.