

# PlayStation®2 Development Environment Setup Guide

© 2001 Sony Computer Entertainment Inc.

Publication date: October 2001

Sony Computer Entertainment Inc.  
1-1, Akasaka 7-chome, Minato-ku  
Tokyo 107-0052, Japan

Sony Computer Entertainment America  
919 E. Hillsdale Blvd.  
Foster City, CA 94404, U.S.A.

Sony Computer Entertainment Europe  
30 Golden Square  
London W1F 9LD, U.K.


The *PlayStation®2 Development Environment Setup Guide* manual is supplied pursuant to and subject to the terms of the Sony Computer Entertainment PlayStation® license agreements.

The *PlayStation®2 Development Environment Setup Guide* manual is intended for distribution to and use by only Sony Computer Entertainment licensed Developers and Publishers in accordance with the PlayStation® license agreements.

Unauthorized reproduction, distribution, lending, rental or disclosure to any third party, in whole or in part, of this book is expressly prohibited by law and by the terms of the Sony Computer Entertainment PlayStation® license agreements.

Ownership of the physical property of the book is retained by and reserved by Sony Computer Entertainment. Alteration to or deletion, in whole or in part, of the book, its presentation, or its contents is prohibited.

The information in the *PlayStation®2 Development Environment Setup Guide* manual is subject to change without notice. The content of this book is Confidential Information of Sony Computer Entertainment.

 and PlayStation are registered trademarks of Sony Computer Entertainment Inc. All other trademarks are property of their respective owners and/or their licensors.

# Table of Contents

<b>About This Manual</b>	<b>v</b>
Changes Since Last Release	v
Related Documentation	v
Typographic Conventions	v
Developer Support	v
<b>Hardware Setup</b>	<b>1</b>
Development System Architecture Overview	1
Features of the DTL-T10000	1
Equipment Required To Use the Programmer Tools	2
Setting up the DTL-T10000	2
DTL-T10000 Setup Notes	2
<b>Software Setup</b>	<b>4</b>
Development Software Overview	4
Installing the Compiler (Tool Chain)	4
Installing Run-Time Libraries	4
Updating dsnet	5
Environment Settings	6
Writing to Flash ROM	6
Restoring the Flash ROM	6
<b>Compiling and Running Sample Programs</b>	<b>8</b>
Changing Access Permission Attributes	8
basic3d: Running an EE Sample Program	8
sqhard: Running an IOP Sample Program	9
sqsoft: Running an EE-IOP Sample Program	9



## About This Manual

This is the Runtime Library Release 2.4 version of the *PlayStation®2 Development Environment Setup Guide* manual.

It describes the steps required to set up the PlayStation 2 software development environment to compile sample programs.

## Changes Since Last Release

- In "Software Setup", a "Restoring the Flash ROM" section has been added.

## Related Documentation

**Note:** the Developer Support Web site posts current developments regarding the Libraries and also provides notice of future documentation releases and upgrades.

## Typographic Conventions

Certain Typographic Conventions are used throughout this manual to clarify the meaning of the text:

Convention	Meaning
<code>courier</code>	Indicates literal program code.
<i>italic</i>	Indicates names of arguments and structure members (in structure/function definitions only).
<b>medium bold</b>	Indicates data types and structure/function names (in structure/function definitions only).
<a href="#">blue</a>	Indicates a hyperlink.

## Developer Support

### Sony Computer Entertainment America (SCEA)

SCEA developer support is available to licensees in North America only. You may obtain developer support or additional copies of this documentation by contacting the following addresses:

Order Information	Developer Support
<i>In North America:</i>	<i>In North America:</i>
Attn: Developer Tools Coordinator	E-mail: <a href="mailto:PS2_Support@playstation.sony.com">PS2_Support@playstation.sony.com</a>
Sony Computer Entertainment America	Web: <a href="http://www.devnet.scea.com/">http://www.devnet.scea.com/</a>
919 East Hillsdale Blvd.	Developer Support Hotline: (650) 655-5566
Foster City, CA 94404, U.S.A.	(Call Monday through Friday,
Tel: (650) 655-8000	8 a.m. to 5 p.m., PST/PDT)

**Sony Computer Entertainment Europe (SCEE)**

SCEE developer support is available to licensees in Europe only. You may obtain developer support or additional copies of this documentation by contacting the following addresses:

Order Information	Developer Support
<i>In Europe:</i> Attn: Production Coordinator Sony Computer Entertainment Europe 30 Golden Square London W1F 9LD, U.K. Tel: +44 (0) 20 7859-5000	<i>In Europe:</i> E-mail: ps2_support@scee.net Web: <a href="https://www.ps2-pro.com/">https://www.ps2-pro.com/</a> Developer Support Hotline: +44 (0) 20 7859-5777 (Call Monday through Friday, 9 a.m. to 6 p.m., GMT)

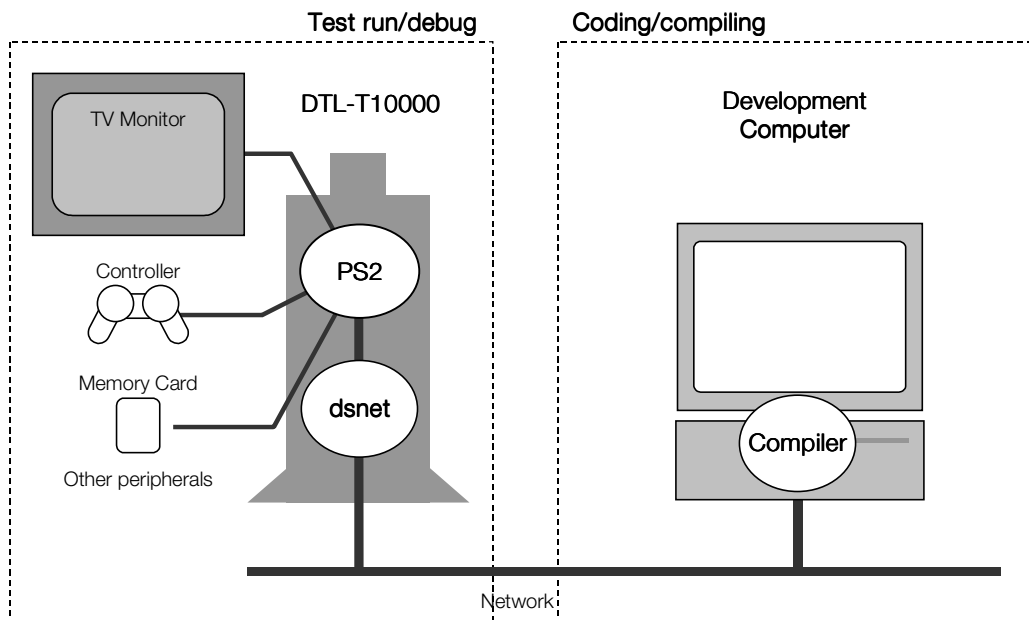
## Hardware Setup

This manual describes the steps required to set up the PlayStation 2 software development environment to compile sample programs. By performing the series of operations described here, the user can confirm that the development environment has been properly set up.

### Development System Architecture Overview

The following is a schematic drawing showing the development system architecture.

**Figure 1: Architecture of the development system**



The development system is based on the DTL-T10000/DTL-T10000H PlayStation 2 software development unit (these are both referred to as the DTL-T10000, below). The DTL-T10000 PlayStation 2 software development unit is what is generally referred to in cross-platform development as a target box. The DTL-T10000 is equipped with features corresponding to those of the actual PlayStation 2 as well as administrative and communication functions (dsnet) to allow programs to be test run and debugged. The platform that supports these functions is a single board PC/AT-compatible that runs Linux and is installed in the DTL-T10000.

The DTL-T10000 is a unit designed specifically for test running PlayStation 2 software; the user does not directly log in to the DTL-T10000 to code programs. Perform coding and compiling on a separate computer (known as a development computer) that is connected to the network.

### Features of the DTL-T10000

The DTL-T10000 is a development tool for developing PlayStation 2 software and has the following features:

- It is equipped with the features of the PlayStation 2 and allows efficient software development.
- It comes with a communications processor that allows it to send and receive software and data with a development computer over a network. This allows programs to be run, checked, and debugged efficiently.

- It can be connected directly to a network, so installation and upgrading is simple. This also reduces the load on the development computer.

## Equipment Required To Use the Programmer Tools

### DTL-T10000

The DTL-T10000 contains a single-board PC (MMX Pentium 233 MHz) serving as the communications processor, but make a separate development PC available to provide a compiling/debugging environment and make it possible to remotely access the DTL-T10000 through a network.

### Development computer

Any standard personal computer can be used for the compile/debug environment. The following list of specifications should serve as a guideline. The recommended environment is RedHat Linux 5.2. If another version of Linux is to be used, make sure that it uses Libc6.

**Table 1: Development computer specifications**

Item	Specification
CPU	Pentium II 400 MHz or higher
Memory	128MBytes min.
PC/AT monitor	17 inches min.
Graphics	SVGA

## Setting up the DTL-T10000

For additional information on setting up and configuring the DTL-T10000, please refer to the instruction manual that comes with the DTL-T10000. The main procedure is as follows:

1. Connect the TV monitor
2. Connect the unit to the network (see notes below)
3. Turn on the main power supply switch on the back, start by pushing the button on top.
4. Run a web browser on the development computer to access the DTL-T10000  
(An administration tool window will be displayed)
5. Use the administration tool screen to set the internal clock, enter network settings, etc.
6. Install "Packages" using the administration tool screen

## DTL-T10000 Setup Notes

### Connecting to the Network

The setup procedure is a little different depending on whether there is a DHCP server on the network.

#### If a DHCP server is on the network

If there is a DHCP server on the network, the IP address and subnet mask will be configured automatically when the DTL-T10000 is connected to the network and turned on. However, this will result in the DTL-T10000 having a different IP address each time, so we recommend using a fixed IP address to allow efficient development. To use an automatically assigned IP address permanently (fixed IP address), consult the administrator of the DHCP server.



**If a DHCP server is not on the network**

If there is no DHCP server on the network (i.e., each device is assigned a permanent IP address), then a local network should be set up and the default IP address should be changed to an IP address that works in the user's environment. The development computer is set up with an IP address of 192.168.0.2 and a subnet mask of 255.255.255.0. The development computer and the DTL-T10000 can then be connected to each other directly using a crossover cable or through straight cables going through a hub that is not already connected to an existing network. The local network will be set up when the development computer and the DTL-T10000 are turned on.

**Installing "Packages"**

Of the DTL-T10000 internal software, dsnet and the flash ROM are updated as necessary. However, the dsnet tools are installed and updated using the administration tools screen. The most recent version will be distributed along with the development software, so updates should be performed using this. The specific procedures will be described later.

## Software Setup

---

### Development Software Overview

SCE provides development software in the form of compilers, libraries, sample programs, and other tools. The main ones are listed below.

- C/C++ compiler  
ee-gcc and iop-gcc. Based on gcc and customized for the EE and the IOP.
- Libraries
- Sample programs
- Various tools

The latest versions of these are available on the developer support website, so download them as necessary.

### Installing the Compiler (Tool Chain)

Download the following packages from the developer support website.

- Tool Chain EE 2.96-ee-001003 and Tool Chain ee-001003-update-01
- Tool Chain IOP 2.8.1

### Segmented Downloads

Since the size of a package is large, it can be broken into smaller files and downloaded. After downloading the entire module, use the cat command to concatenate each of the files.

Example:

```
$ cat ee0010.0* > ee0010.tgz
```

### Uncompress / Install

Gain root privilege. If a previous version already exists, we recommend saving it before performing the installation.

Example:

```
# mv ee0010.tgz /usr/local/
# cd /usr/local/sce/ee
# mv gcc gcc.old (Renamed previous gcc)
# cd /usr/local
# tar zxvf ee0010.tgz
:
$ ee-gcc -v (Confirm version)
:
gcc version 2.96-ee-001003-01
```

### Installing Run-Time Libraries

The run-time libraries are frequently updated. Download the latest version from the developer support website. Use the following procedure to install the downloaded library.

## Uncompress / Install

1. Development software is installed under /usr/local/sce. When updating, it is safer to save the previous environment by renaming the directory and installing from scratch.

2. As root, unpack in /usr/local.

```
$ su
:
# cd /usr/local
# mv sce sce_old
# tar xzvf tlib_XXX.tgz
:
# ls -l -t
sce
readme_j.txt
readme.txt
t10000-XXXX.bin
:
:
```

## Notes on Installing Software

Development software assumes it is installed in /usr/local/sce, so when installing always unpack using /usr/local as the current directory. If installing elsewhere, create a symbolic link as user root from /usr/local to the other location (e.g., /home/xxx/sce).

```
# ln -s /home/xxx/sce /usr/local
```

## Copying IOP Libraries and Header Files

IOP library header files and static libraries must be copied to a standard directory. A script to do this is provided, so execute sce/iop/install/Install.sh.

## Updating dsnet

dsnet on the DTL-T10000 can be updated. Go to "T10000 administration tools" using a web browser on the development machine and specify the following rpm package for updating. Specific instructions are also provided in the T10000 instruction manual.

1. Check to see if the following file is included in the development software that was installed.  
/usr/local/sce/rpm/dsnet-X.X.XX-X.i386.rpm  
("X.X.XX-X" indicates the version number)
2. Use a web browser to access the DTL-T10000.
3. Select "Packages" from the Settings menu.
4. Enter the host name or IP address of the development computer, and a user name and password for logging into the development computer. Enter "/usr/local/sce/rpm" in the "Directory" field.
5. When a list of packages is displayed, select the file shown above and click the "Install" button.
6. Select "Shutdown" from the Settings menu and reboot the DTL-T10000.

## Environment Settings

### Path settings

The following directories must be added to the path.

Add the following to the lines setting up the path in the environment settings file of the shell (.bash\_profile or .cshrc).

- usr/local/sce/bin
- usr/local/sce/ee/gcc/bin
- usr/local/sce/iop/gcc/bin

### Environment Variable Settings

By specifying the T10000 hostname (or IP address) in the DSNETM environment variable, dsedb, etc. can be run with no need to specify the -d option.

Make the following additions to the shell environment settings file (e.g. .bash\_profile, cshrc).

- for .bash\_profile(.bashrc): `$ export DSNETM=hostname`
- for .cshrc: `setenv DSNETM hostname`

## Writing to Flash ROM

Of the DTL-T10000's internal software, the Flash ROM, which corresponds to the kernel of the PlayStation 2 is distributed in its newest release as an image file accompanying the development software. The Flash ROM can be updated using the following procedure.

1. Check to see that the following file is included in the development software that was installed.  
    /usr/local/t10000-XXXX.bin  
    ("XXXX" indicates the version number)
2. Run dsflash.  
    \$ dsflash /usr/local/t10000-XXXX.bin  
    \*\*\* Updating Flash ROM ...  
    \*\*\* Resetting...  
      
    MPU-4.0 board DSW602
3. When the program finishes and displays "Complete!", run dsreset.  
    \$ dsreset  
    dsreset (Version X.X.XX...)  
    \$

### Confirming the Flash ROM Version

Starting dsibd will display the message shown below. Confirm the underlined portion.

```
CPUID=15, ROMGEN=2001-0219, CACH_CONFIG=1edd8, 8MB, IOP mode,
FlashROM boot <20010219-213228,ROMconf,t10000-relXX.bin:11824>
```

## Restoring the Flash ROM

If a write failure occurred because dsflash processing was interrupted for some reason and the contents of the Flash ROM are destroyed, the kernel will not be able to start and tools such as dsflash / dsreset / dsedb will not be usable. If this happens, follow the procedure below to restore the Flash ROM.

1. Set the TOOL/WS switch located at the lower left of the DTL-T10000 front panel to WS.

2. Restart the DTL-T10000.  
It should start up in WS mode. The TOOL/WS LED is located at the top of the front panel of the DTL-T10000. TOOL should be lit regardless of the mode.
3. Verify that dsedb starts and the prompt appears correctly, then quit.

Example:

```
$ dsedb -d hostname
*** Resetting...
EE DECI2 Manager version...
  CPUID=2e12, BoardID=4126, ROMGEN=...., 128M
*** DBGP Version ...
:
dsedb S> quit
$
```

4. Execute the following command to rewrite the Flash ROM in TOOL mode.

```
$ dsflash -mpu4shadow t10000-XXXX.bin
```

5. Verify that "Complete!" is displayed and that dsflash terminates.
6. Return the TOOL/WS switch to TOOL.
7. Restart the DTL-T10000. It should start up in TOOL mode.
8. Verify that dsedb starts and the prompt appears correctly.

Do not execute [dsflash -mpu4shadow] in TOOL mode. If you do, the Flash ROM will be rewritten for WS mode, but if the rewrite were to fail, then it would not be possible to manually restore the Flash ROM. If both the TOOL and WS contents are destroyed, the Flash ROM will need to be repaired.

## Compiling and Running Sample Programs

The procedure for building sample programs and running them on the DTL-T10000 is described below. If the sample programs run properly, the development environment is set up correctly.

### Changing Access Permission Attributes

After installing, the `sce/ee/sample` and `sce/iop/sample` directories will be read-only. To build sample programs, change the access permission attributes in the following manner.

```
$ su root
Password:xxxxxxx  <= enter root password
# cd /usr/local/sce/ee
# chmod -R a+w sample      <= make sample directory and subdirectories
world-writable
# exit
$
```

Make the same changes for the `iop` directory.

### basic3d: Running an EE Sample Program

The procedure for building and running a program that is run on the EE will be described, using `sce/ee/sample/basic3d/vu1` as an example.

#### 1. Building the Sample Program

Build using 'make' or 'make all'.

The sample programs use general purpose Makefiles so they can be used for user programs as well by rewriting the `TOP=` and `OBS=` lines, etc.

```
$ cd /usr/local/sce/ee/sample/basic3d/vu1
$ make all
```

#### 2. Running the Sample Program

If the `DSNETM` environment variable described above is set up properly, the sample program can be run with 'make run'. If the variable is not set, the program can be run by indicating the T-10000 hostname, as shown below.

```
$ dsedb -d hostname -r run main.elf
```

The elf file is sent to the T-10000 via the network. If the program runs properly, it will produce the following output.

```
$ dsedb -d hostname -r run main.elf
*** Resetting...
EE DECI2 Manager version...
CPUID=..., BoardID=..., ROMGEN=..., 128M

Loading program (address=...size=...)...
Loading XXX symbols...
Entry address =...
GP value    =...
```

Operate the controller connected to the DTL-T10000.

To stop the sample program, press Ctrl+C from the console of the development computer. An image will stay on the monitor but this is normal.

You can build all the samples by executing make in the sce/ee/sample directory.

Depending on the sample program, there is a possibility that a build may not work just by executing make run. Refer to the explanatory file readme.txt located in each sample directory.

## sqhard: Running an IOP Sample Program

The procedure for building and running a program that runs on the IOP will be described, using sce/iop/sample/sound/sqhard, a sample program that performs MIDI playback using the hardware synthesizer.

### 1. Building sqhard

To build the program, use the make command from the directory containing the sample program.

```
$ cd /usr/local/sce/iop/sample/sound/sqhard
$ make
```

### 2. Running sqhard

Run the built program using 'make run'

```
$ make run
```

Once the module specified in the makefile is loaded, a tune will be played. To stop while the tune is playing, press Ctrl+C. If the tune continues to play, run dsreset.

Almost all sample programs can be built by executing make run although you should read the explanatory file readme.txt located in each sample directory to verify the execution method.

Generally, the procedure to execute a program on the IOP is as follows.

### 1. Reset target

Execute the following command to prevent file requests from the EE from loading modules.

```
$ dsreset -d hostname 0 2
```

### 2. Load and start program

```
$ dsistart -d hostname programname
```

or

```
$ dsidb -d hostname
```

(Start message will display)

```
dsidb> mstart programname
```

### 3. Reset target

Execute the following command to return to a state in which programs can execute on the EE.

```
$ dsreset -d hostname 0 0
```

## sqsoft: Running an EE-IOP Sample Program

The procedure for building and running a program that runs on the EE and IOP will be described, using sce/iop/sample/sound/sqsoft, a sample program performing MIDI playback using the software synthesizer. The program operates together with sce/ee/sample/sound/sqsoft, which runs on the EE side.

### 1. Building the IOP-side module

## 10 Compiling and Running Sample Programs

First, the program module to be run on the IOP is built.

```
$ cd /usr/local/sce/iop/sample/sound/sqsoft
```

```
$ make
```

### 2. Building the EE-side program

Next, the program to be run on the EE is built.

```
$ cd /usr/local/sce/ee/sample/sound/sqsoft
```

```
$ make
```

### 3. Checking sound element data

Check to see that sound element data files are present.

```
$ ls /usr/local/sce/data/sound
```

### 4. Running the sample program

Run the EE-side program that was built.

```
$ make run
```

main.elf is sent to the DTL-T10000 and executed, the IOP-side module is loaded, sound element data is loaded, and a tune is played. Press Ctrl+C from the console on the development computer to stop the tune. If the tune continues to play, run dsreset.

The method used to execute sample code depends on the individual sample. For details, read the readme.txt files in the sample directories.



