

# Lesson 1

## SDK Installation and Compiling

### Playstation 3 Development

Sam Serrels and Benjamin Kenwright<sup>1\*</sup>

#### Abstract

A beginners guide to getting started with programming and developing on the Sony Playstation 3 (PS3). This article gives a brief introduction for students on setting up the development kit (DevKit) and working with the console specific tools and software.

#### Keywords

Sony, PS3, PlayStation, Setup, Windows, Target Manager, ELF, PPU, SPU, Programming, ProDG, Visual Studio

<sup>1</sup> *Edinburgh Napier University, School of Computer Science, United Kingdom: b.kenwright@napier.ac.uk*

#### Contents

<b>Introduction</b>	<b>2</b>
<b>1 Development Tools</b>	<b>2</b>
1.1 Playstation 3 Software Development Kit (SDK) . . . . .	2
1.2 Application Specific Interface (API) . . . . .	2
1.3 Sony Playstation 3 Development Kit (DevKit) . . . . .	2
<b>2 Installing and Setting-up</b>	<b>2</b>
<b>3 Compiling Basic Example</b>	<b>3</b>
<b>4 Conclusion</b>	<b>4</b>
<b>Acknowledgements</b>	<b>4</b>
<b>References</b>	<b>5</b>

#### Introduction

**About the Edinburgh Napier University Game Technology Playstation 3 Development Lessons** Edinburgh Napier University Game Technology Lab is one of the leading game teaching and research groups in the UK - offering students cutting edge facilities that include Sony's commercial development kits. Furthermore, within the Edinburgh Napier Game Technology group are experienced developers to assist those students aspiring to releasing their own games for PlayStation. Student have constant access to the Sony DevKits and encourage enthusiastic students to design and build their own games and applications during their spare time.

**Getting Started** Sony's Playstation 3 (PS3) development tools support the ANSI C/C++ coding standard. Hence, it's straightforward to compile and run programs for the PS3. This article introduces the fundamental tools and software necessary to connect, compile, and debug your programs on the PS3. The lessons are intended to give step-by-step explanations in conjunction with screen-captures and code samples to get the students up to speed quickly on developing on the PS3.



**Figure 1. PS3 Development Kit - Sony PlayStation 3 Development Kit DECR-1000A** - What is so good about a PS3 DevKit over a PS3 Debug/Test console? For one thing, you can debug code as if you were running it on your PC, furthermore, you can dump/debug the PS3 Hypervisor and Kernel.

**Additional Reading** In addition to the lesson tutorials, we would recommend reading a number of books on Playstation 3 development and cross-platform coding, such as, Cell Programming for the PS3 [3], Vector Maths and Optimisation for the PS3 [1], and Cross-Platform Development in C++ [2].

**Components** The Playstation 3 development lessons are divided into subsequent parts. The article lessons give the students a chance to work within a commercial console environment. In addition, it allows the students to port their games/applications to the PS3. The different lessons include:

- Setting-Up and Installing
- Running and Debugging on the PS3
- Graphics

- Input (i.e., Controller)
- Sound
- SPU and PPU

## 1. Development Tools

You need access to the following tools:

- Microsoft Windows (x32 or x64)
- Microsoft Visual Studio 2010 or 2013
- Sony's Playstation 3 Software Development Kit (SDK) - available from ps3-edu.scedev.net.
- Sony's Playstation 3 Development Kit (DevKit)

### 1.1 Playstation 3 Software Development Kit (SDK)

The Sony PS3 SDK provides documentation, compilers, and the tools required to deploy and debug on the Playstation 3 development kit.

**Which SDK Version?** The different SDK installers install different versions of the PS3 SDK. You will want to use the most recent version, however an older version may be required in some situations, so the various installers are listed below:

- 2011-06PS3SDKOfflineInstaller.exe - version 3xx - 1.81gb - md5: 01fae94cce9277c1d5e2819ba3d84607
- 2011-10PS3SDKOfflineInstaller.exe - version 3xx - 1.86gb - md5: 49e2a7941b459360bc5d0b493b3867c1
- 2012-02PS3SDKOfflineInstaller.exe - version 400 - 1.96gb - md5: e372f0bb50887526f81fb32032d9bbb4
- 2012-10PS3SDKOfflineInstaller.exe - version 420 - 1.42gb - md5: bf4aa4belf31ae641b178533396982d2
- 2014PS3SDKOfflineInstaller.exe - Version 450 - 1.44gb - md5: 53882f02b3ee20439f88ccbddeee2b6f

Note: When running the PS3 SDK installer, ensure that the extraction path does not contain any ampersands (&).

**Compilers** The processor inside the PS3 is based on the *cell broadband architecture*, rather than the standard x86 architecture that your desktop computer uses. This means that a compiler that supports the cell architecture must be used, the SDK comes with two: The SN systems compiler (SNC) and the SCE GCC compiler (GCC). Compiled executables have the extension '.self' - for signed elf.

**GCC** The GNU Compiler Collection (GCC) is a collection of cross platform compilers for various programming languages. The SCE GCC compiler inherits from this standard and behaves just like any gcc compiler, but with PS3 support. The SN systems compiler is based on GCC at its foundation, but it takes a more Playstation 3 specific path, resulting in theoretically better performing code. The baseline is: GCC is more compatible with existing code, SNC produces optimised code at the cost of compatibility.

**Visual studio integration** With the compilers and linkers available in the SDK you could build and deploy with command line scripts, or you could use Visual Studio. Visual studio integration comes in the form of a separate installer from SN systems (VSI2013.exe / VSI2010.exe). This installer adds the functionality to easily create, deploy and debug ps3 projects from within Visual Studio.

**Visual studio integration issues** Visual studio must be ran at least once prior to installing the PS3 SDK + VSI20XX.exe. The Visual studio integration must be installed after the main SDK. When opening an existing project with visual studio, double check the build configurations. Even with a successful install, sometimes VS will default to a win32 build configuration, even in a PS3 project. The VSI installers can only be used with the specified version of visual studio, they are not backwards compatible

- VSI2010.msi (5.42mb md5: c90a5c2aa06...)
- VSI2013.msi (10.3mb md5: f4e2491b466...)

### 1.2 Application Specific Interface (API)

The PS3 SDK provides a set of console specific libraries that are used to compile and run your PS3 executable. Within the PS3 SDK you will find a vast assortment of documentation on the different APIs. This includes information on the six Synergistic Processing Units (SPUs) that give the PS3 hardware specific speed-ups. Understanding and exploiting the six Synergistic Processing Units (SPUs) requires skill and creativity if you intend to exploit the full computational power of the PS3.

**Documentation** After the SDK install, your primary location of samples and documents is the "*C:/usr/local/cell*" directory, you should create a short-cut to this as you will be visiting it often. The best source of information is the main "PS3\_SDKDoc.e" doc, located in "*cell/SDK.doc/en/chm*".

**ProDG Target Manager (TM)** You connect to your PS3 DevKit and debug your program through the ProDG Target Manager (TM) and Debugger. The ProDG TM allows you to connect to the PS3 remotely (i.e., via the IP address), copy the executable across, launch the executable (i.e., even without the debugger), take screenshots, and view the terminal output (e.g., raw text output as you would do with your desktop text output - such as, puts(..) and printf(..) shown in Listing 1).

**ProDG Debugger** The ProDG Debugger runs on the development pc at the same time as your application on the devkit. It can be used to inspect the state of the processing hardware, with data readouts like: the stack contents, registers, memory locations and process/thread states. If there is an error in your application, this is where it would be detected. (Figure 2)

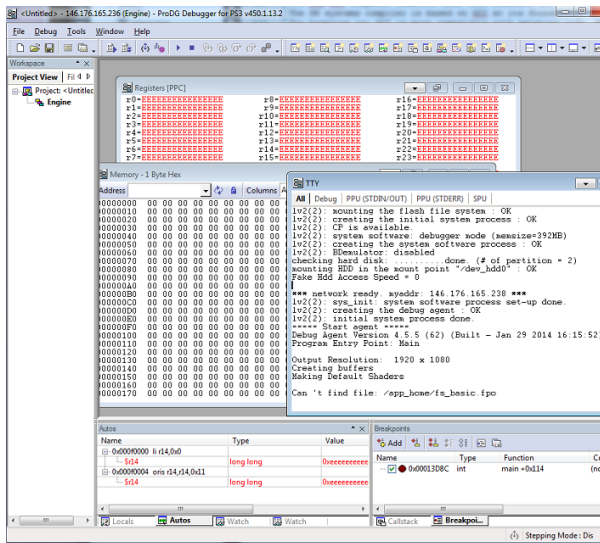
### 1.3 Sony Playstation 3 Development Kit (DevKit)

The Sony Playstation 3 DevKit plays normal games similar to a retail PS3 console. However, the PS3 DevKit possesses various hardware enhancements. For example, extra memory, i.e., 512MB instead of the 256MB retail units and custom firmware (e.g., see Figure 1).

## 2. Installing and Setting-up

We give a step-by-step guide to getting the PS3 SDK installed and having you connect and be able to run an executable on the PS3 dev kit.

1. Install Visual Studio 2010 or 2013, run it once, then close it.
2. Run PS3 SDK Installer (see Figure 5)
3. Install the required version of the Visual studio Integration



**Figure 2. ProDG Debugger** - Used to inspect the inner workings of the processors as the application is running

(VSI20XX.msi)

4. Add the PS3 DevKit IP to your Target Manager (Figure 7)
5. Right click on the PS3 IP in your Target Manager and select Connect (e.g., Figure 8)
6. Connected and ready to compile and run your compiled programs on your PS3 DevKit

**Connected** At this point, you should be able to connect to the PS3 DevKit remotely via the IP address. You can restart the PS3 or launch executables via the Target Manager. Furthermore, when you run an executable on the PS3 you can capture text output via the tty window or take screenshots on-the-fly.

### 3. Compiling Basic Example

An uncomplicated PS3 project inside the SDK you can open and compile quickly to check that your installation is correct and that you can compile and generate a PS3 executable is the 'TTYEcho' project (i.e., in folder: "C:/Program Files (x86)/SN Systems/PS3/examples/TTY/TTYEcho/"). Within this directory you should find a 'TTYEcho.vcxproj' and 'TTYEcho\_vs2010.sln'. Open the project in Visual Studio (e.g., open Visual Studio 2012, select open project, then navigate to the folder TTYEcho and open the project). If your PS3 SDK software installed correctly, the project should open within Visual Studio without a problem.

**Errors** If you encounter this error message: "Platform 'PS3' referenced in the project file 'xxx' cannot be found. Please make sure you have it installed under '%VCTargetsPath%\Platforms\PS3'" Then the sdk has not intergrated into Visual Studio correctly. Find and run VSI2010.msi  
If you encounter the error message: 'The selected file is a solution file, but was created by a newer version of this application and cannot be opened'.

**TTYEcho** Once you've opened the project you can see it has a single '.c' file called 'Echo.c', as shown in Listing 1. The

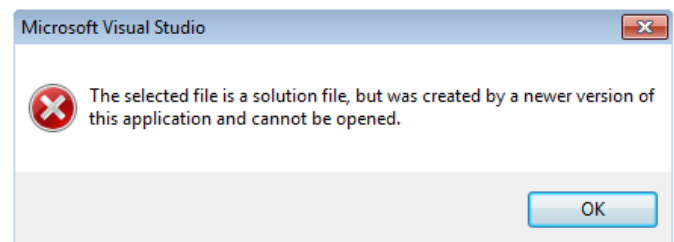
source code is uncomplicated and conforms to the basic 'C' standard. When you compile the program you should get 'Build Succeeded'. Navigating to the project directory, you will notice a folder called 'PS3\_PPU\_Debug'. Within this directory you should notice the compiled files, such as Echo.o, TTYEcho.log. However, most importantly, you should see 'Echo.ppu.self', which is the executable you copy and launch on the PS3 using the Target Manager.

**Listing 1. Echo.c**

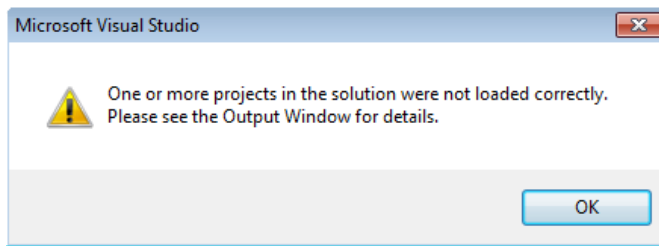
```
1 #include <stdio.h>
2 #include <stdlib.h>
3
4 int main(void)
5 {
6     char ch = '\0';
7
8     puts("EchoTTY: Hit Q to quiet!\n");
9
10    do
11    {
12        int n = fread((void*) &ch, 1 sizeof(ch), stdin);
13        if ( n&& ch!=EOF)
14            putchar(ch);
15    } while (ch != 'Q' );
16
17    puts("EchoTTY: Quitting!\n");
18    return EXIT_SUCCESS;
19 }
```

**Executable Output** You won't see any sexy graphics on screen when you run the TTYEcho program. In fact, the screen will remain black. However, after you launch the executable in the Target Manager. Still within the Target Manager, select the 'TTY' output window. You will see the PS3 load the executable and then print out the text 'EchoTTY: Hit Q to quit!' - proving your code has executed on the PS3. The program is now sitting and waiting for the input in the while loop (as shown in Listing 1).

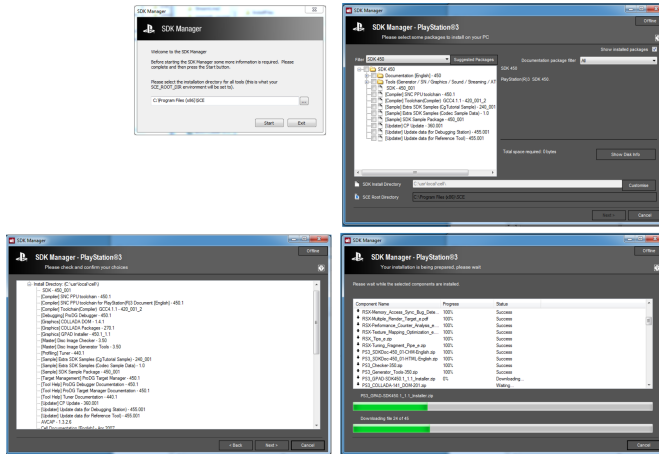
Congratulations - you have successfully compiled and run your first PS3 program. You are now ready to move forwards and start compiling more complex programs to take advantage of graphics, sound, and the controller input.



**Figure 3. Error Opening PS3 Project Within Visual Studio** - Wrong version of the PS3 SDK and Visual Studio.  
C:\Program Files (x86)\SN Systems\PS3\examples\TTY\TTYEcho\TTYEcho.vcxproj:  
warning : Platform 'PS3' referenced in the project file 'Echo' cannot be found. Please make sure you have it installed under '%VCTargetsPath%\Platforms\PS3'.



**Figure 4. Error Opening PS3 Project With Wrong Version of Visual Studio - Wrong version of the PS3 SDK and Visual Studio.**



**Figure 5. Installing PS3 SDK -**

## 4. Conclusion

In summary, if everything went well, you should be setup and ready for developing on Sony's Playstation 3. The PS3 SDK and Visual Studio integration should be working. You should have successfully compiled and launched a 'signed' PS3 executable on the PS3 DevKit and viewed the TTY output in the Target Manager.

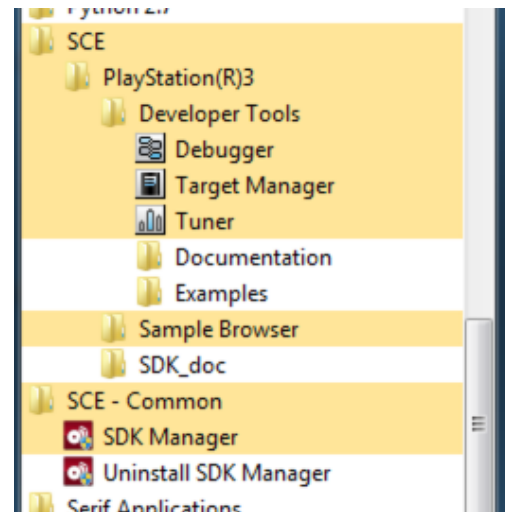
## Appendix

### Task Manager

**Screen Captures** If you're not able to touch the PS3 physically, you can still connect and execute your program remotely. Furthermore, if you want you can take still video capture images of the screen using the Task Manager (e.g., Figure 9).

**Listing 2.** The Target Manager screen capture facility needs to know the PS3 graphical output parameters (e.g., dimensions and colour pitch). We can determine this in code using the following code snippet.

```
1 {
2     cellGcmSetDisplayBuffer(0, fbOffsetFront, colorPitch, ←
        resolution.width, resolution.height);
3
4     #define BYTES_PER_PIXEL 4
5     printf("=====\n");
6     printf("Buffer ID: \t0x%.8X\n", 0);
```



**Figure 6. PS3 Tools - After PS3 installer has finished, you should notice extra options within the start menu.**

```
7     printf("Buffer Address: \t0x%.8X\n", fbOffsetFront+0←
        xC0000000);
8     printf("Display Width: \t%d\n", resolution.width);
9     printf("Display Height: \t%d\n", resolution.height);
10    printf("Color Pitch: \t%d\n", colorPitch)←
        BYTES_PER_PIXEL);
11    printf("=====\n");
12 }
```

**Listing 3. Output from Listing 2**

```
1 =====
2 Buffer ID: 0x00000000
3 Buffer Address: 0xC0000000
4 Display Width: 720
5 Display Height: 480
6 Color Pitch: 768
7 =====
```

**.self File Format** The 'signed' elf (i.e., .self) PS3 executable format specification is available online for those that want to investigate the low-level details. For example, the first seven bytes of a compiled '.self' executable if you open it in a hex-editor are: 'SCE0000'.

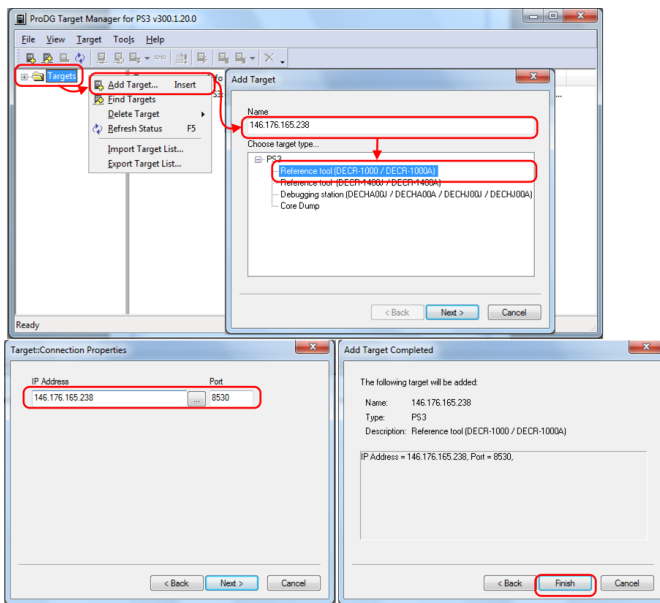
**stdio and stdlib** All of the standard C/C++ libraries have been ported across to the PS3 - hence, it's very easy to port across basic C/C++ code to the PS3 (e.g, sprint, fopen, write, puts).

## Acknowledgements

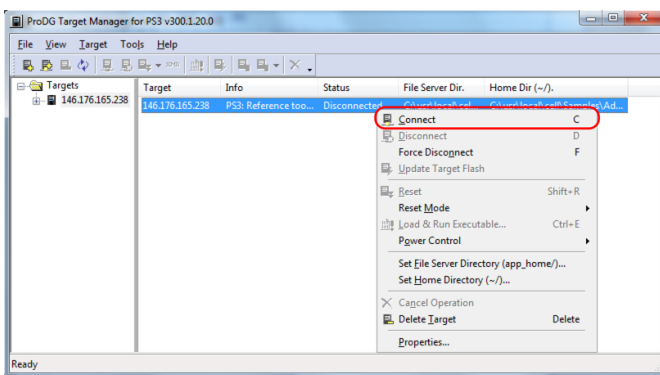
We would like to thank all the reviewers for taking time out of their busy schedules to provide valuable and constructive feedback to make this article more concise, informative, and correct. However, we would be pleased to hear your views on the following:

- Is the article clear to follow?
- Are the examples and tasks achievable?
- Do you understand the objects?
- Did we miss anything?





**Figure 7. PS3 Tools - Target Manager: Start - SCE - Playstation(R)3 Development Tools - Target Manager**



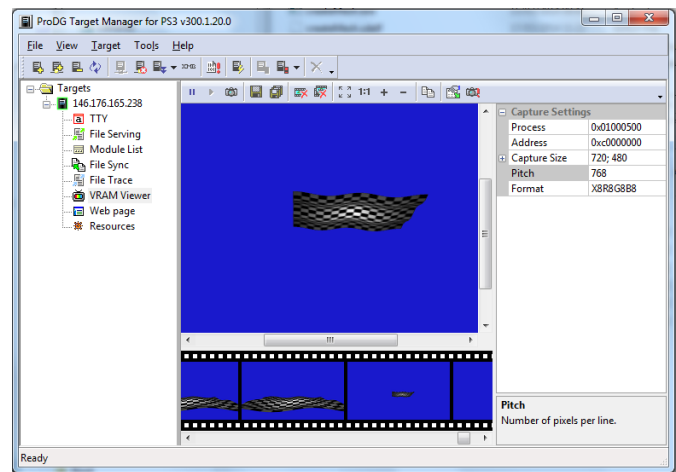
**Figure 8. PS3 Tools - You connect via the PS3 IP address (e.g., 146.176.165.238). You can see the PS3 IP address if you reset the PS3 in debug mode - it should display on the screen.**

- Any surprises?

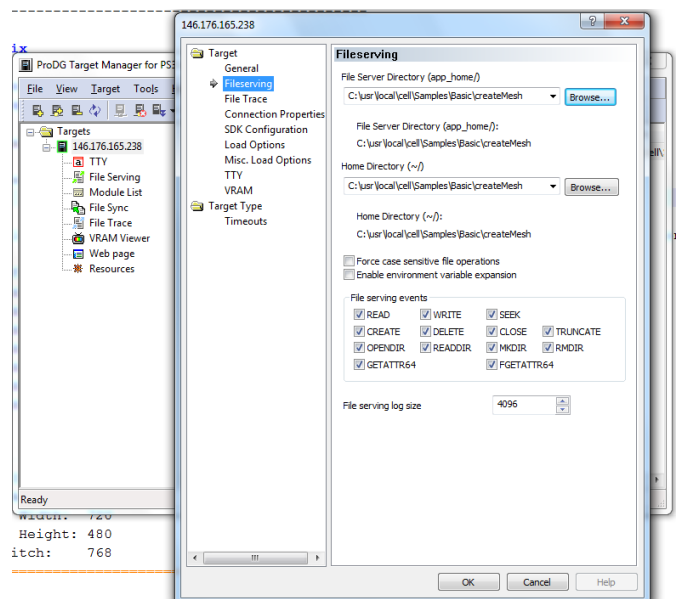
The lessons provide a basic introduction for getting started with Sony's Playstation 3 console development. So if you can provide any advice, tips, or hints during from your own exploration of PS3 development, that you think would be indispensable for a student's learning and understanding, please don't hesitate to contact us so that we can make amendments and incorporate them into future tutorials.

## Recommended Reading

Programming the Cell Processor: For Games, Graphics, and Computation, Matthew Scarpino, ISBN: 978-0136008866  
Vector Games Math Processors (Wordware Game Math Library), James Leiterman, ISBN: 978-1556229213  
Clean Code: A Handbook of Agile Software Craftsmanship, Robert C. Martin, ISBN: 978-0132350884



**Figure 9. Task Manager Capture -**



**Figure 10. Task Manager Folder -**

## References

- [1] James Leiterman. Vector games math processors (wordware game math library) (isbn:978-1556229213), 2011. [1](#)
- [2] Syd Logan. Cross-platform development in c++: Building mac os x, linux, and windows applications (isbn:978-0321246424), 2007. [1](#)
- [3] Matthew Scarpino. Programming the cell processor: For games, graphics, and computational processing (isbn: 978-0136008866), 2011. [1](#)