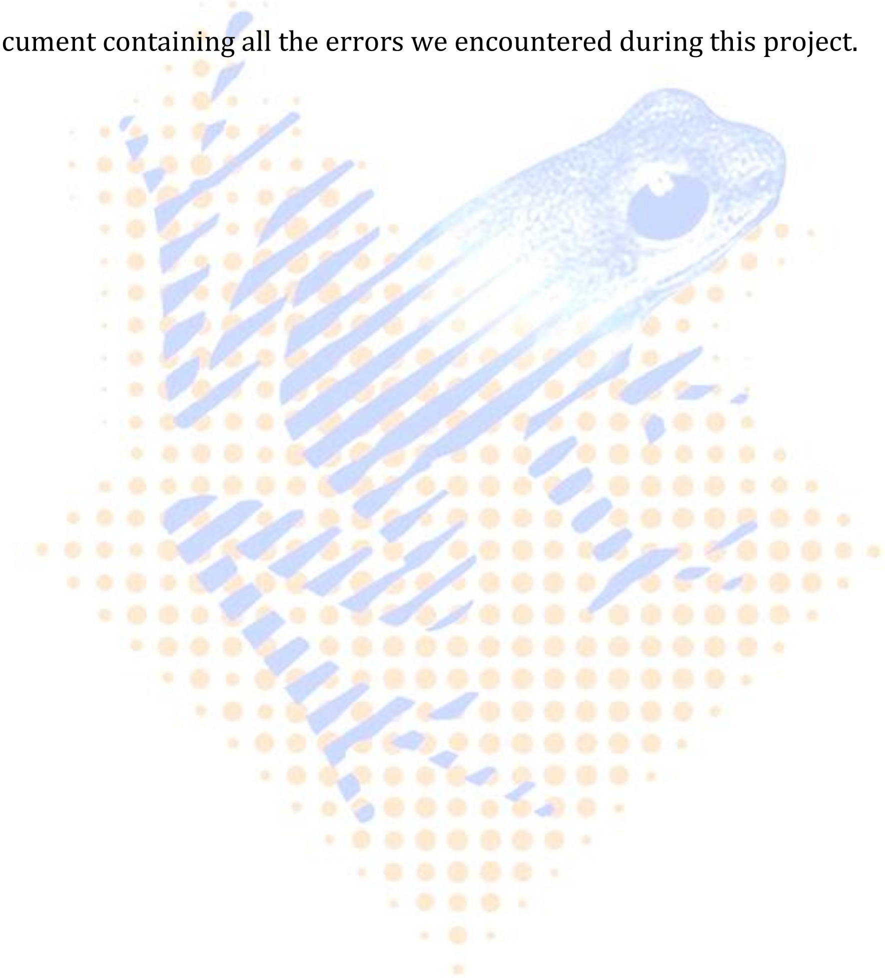
Encountered Errors

A document containing all the errors we encountered during this project.



Frank Blaauw & Spencer Shaw

10-6-2009

Encountered Errors

By Spencer Shaw and Frank Blaauw.

This guide is written for development on the Playstation 3 and gives a How-to on some of the errors we encountered during our project. This document was written for the UMCG department Radiology. Answers for these errors were obtained using the IBM SDK Forums and some by trial and error.

In no event will the UMCG nor the writers of this document be liable for any damage arising directly or indirectly from any use of the information obtained from this document.

This guide and many other guides can be downloaded from: http://code.google.com/p/fedora-cell-project/

If you encounter any errors on using this document, please read the inform us via the google-code page or google-group.

Table of Contents

1. The Errors 4

1.1 ALF Template bug. 4

1.2 Alf accelerator side partitioning 5

1.3 Eclipse crashing on startup. 6

1.4 System Simulator. 7

1. The Errors

During this project we encountered a lot of problems, therefore we decided to create a document containing the most important, time consuming and moreover the most annoying problems. Note, most of these problems were (submitted) bugs, and might be fixed in a new version of the SDK.

1.1 ALF Template bug.

When you use Eclipse to create a Cell B/E Project using ALF, and you use the wizard to create a ALF project, you might experience some trouble running the program. The program will compile and run, and everything will look fine, except that it doesn't do anything. This is because the wizard generates a library file with the wrong name.

For example, if your project is called: myAlfTestProject, the wizard will create 3 directories:

- ppu_myAlfTestProject
- spu_myAlfTestProject
- libmyAlfTestProject

the last directory contains the problem. When you compile your application, and then open the directory: libmyAlfTestProject/Binaries, or libmyAlfTestProject/ppu-gnu32-debug, you will see a shared library called liblibmyAlfTestProject. Your code will look for a library called libmyAlfTestProject only the directory contains a file called **lib**libmyAlfTestProject. To fix this, go to your ppu_myAlfTestProject.c file and change the

```
sprintf(library_name, "libmyAlfTestProject.so");
```

to

```
sprintf(library_name, "liblibmyAlfTestProject.so");
```

and everything will work like a charm. On the google code page of this project is a fix (Alftemplate_fix.zip) which contains an edited template. This template doesn't contain this error.

1.2 Alf accelerator side partitioning

When you use the Eclipse wizard to create a project using Accelerator side partitioning, you might experience an error that says it can't find the definition of 'PTR_TO_ADDR64'. This is because the method is yet to be defined in the PPU-Header file. To do this add these lines:

```
  #define PTR_TO_ADDR64( ptr , addr) {                        \
         (addr) = ( (unsigned long)(ptr) & 0xFFFFFFFFUL);      \
 }
```

The program should run now. Note that the slashes on the end of the code are required. Whether this method fits your needs or not, is up to you.

1.3 Eclipse crashing on startup.

If your eclipse crashes at about 90% on startup, try the following steps.

1. Download a backlevel xulrunner version from

   http://releases.mozilla.org/pub/mozilla.org/xulrunner/releases/1.8.1.3/contrib/linux-i686/xulrunner-1.8.1.3.en-US.linux-i686-20080128.tar.gz

2. Extract its contents into an accessible directory such as /opt/xulrunner.

   ```
   tar -x xulrunner-1.8.1.3.en-US.linux-i686-20080128.tar.gz
   ```

3. Edit the file /opt/cell/ide/eclipse/eclipse.ini to read:

   ```
   ...
   -vmargs
   -Dorg.eclipse.swt.browser.XULRunnerPath=/opt/xulrunner/
   ...
   ```

Note: Make sure you have the -vmargs argument prior to the -Dorg.eclipse... argument, also if you have executed the xulrunnerfix.sh, this file's last line should be:

```
-Dorg.eclipse.swt.browser.XULRunnerPath=/opt/xulrunner/
```

Delete this line.

4. You can now start eclipse

   ```
   /opt/cell/ide/eclipse/eclipse
   ```

Your Eclipse IDE is now ready to use.

If Eclipse still crashes, or doesn't work like it should work, try running eclipse in the IBM Java VM:

```
/opt/cell/ide/eclipse/eclipse –vm /opt/ibm/java2-i386-50/jre/bin
```

Your Eclipse IDE should now work properly.

1.4 System Simulator.

If you are planning to use the system simulator, you might encounter a problem that the system simulator stops loading at about 21%, this is because the kernel is waiting for a trigger which isn't send correctly. So what you have to do is simulate a trigger. Type this in the console of the system simulator:

```
echo “Welcome to Fedora Core”

echo “INIT: Entering runlevel: 2”
```

Your system simulator will now continue booting, and will be ready for use in a couple of minutes.