# Guide to Enigma executable building

What you need:

- Visual Studio 2010 (Visual C++ Express 2010 might be sufficient, but is not tested)
- Newest DirectX SDK (e.g. June 2010)
- Java Runtime (grab the JRE from http://www.oracle.com/technetwork/java/javase/downloads/index.html)
- QT 4.8 (only if you want to build eStudio itself, which most people should not need)

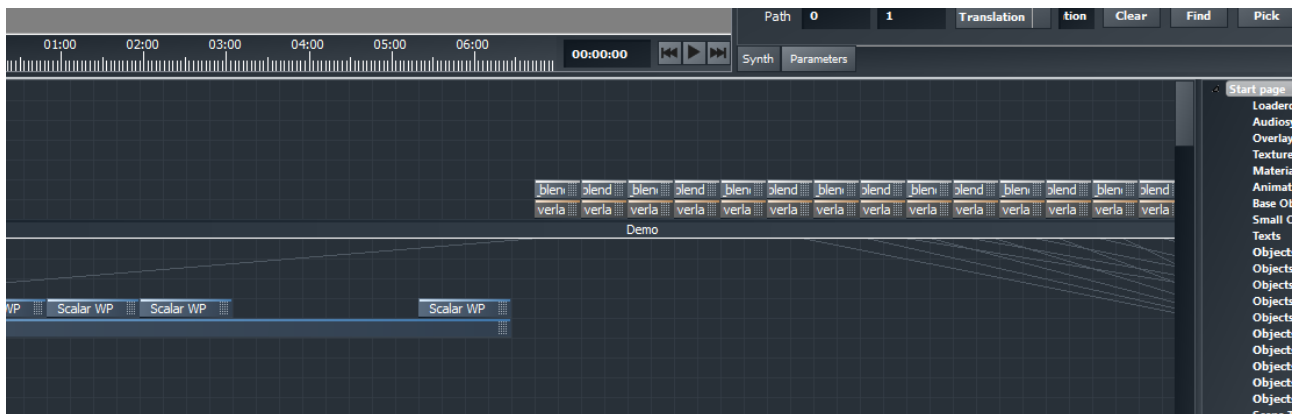## Step 1: Exporting your project from eStudio

Step 1.1: Delete production.cpp

you will find this file in /code/eplayer3 directory. This is exactly what we want to export and generate by the tool so delete the old first so we can see if the export worked fine afterwards.
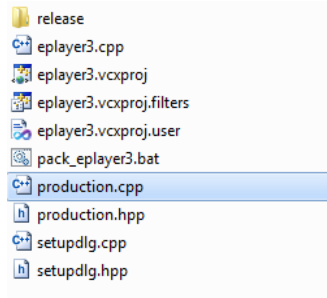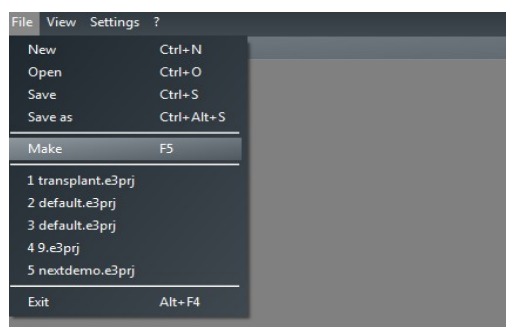

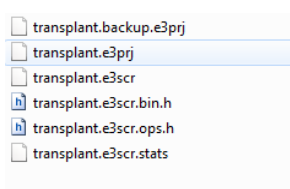### Step 1.2: Run eStudio and export demo

Run the tool and find and click the „demo" operator of your project:

Now export it using the „make" option in the „file" menu:

After that you should have saved your project as a e3scr file. There should also be a couple more files which we will need. Namely the 2 files ending in bin.h and ops.h

Now comes the tricky part. The export will also create a file „productionsources.gen" that will sit in your binary directory. If you did not compile your eStudio yourself, this will probably contain some wrong paths. Open the file and replace „d:\dev\enigma3_trunk\" with the path, you unpacked the enigma code.
It might look something like this afterwards:

```
 1   eDemoOp;C:\dev\Enigma_Studio_3_-_Transplant_Edition\code\eshared\opstacking\miscops.cpp
 2   eSeqSceneOp;C:\dev\Enigma_Studio_3_-_Transplant_Edition\code\eshared\opstacking\sequencerops.cpp
 3   eFxMergeOp;C:\dev\Enigma_Studio_3_-_Transplant_Edition\code\eshared\opstacking\effectops.cpp
 4   eFxAdjustOp;C:\dev\Enigma_Studio_3_-_Transplant_Edition\code\eshared\opstacking\effectops.cpp
 5   eFxSsaoOp;C:\dev\Enigma_Studio_3_-_Transplant_Edition\code\eshared\opstacking\effectops.cpp
 6   eFxCameraOp;C:\dev\Enigma_Studio_3_-_Transplant_Edition\code\eshared\opstacking\effectops.cpp
 7   eSceneOp;C:\dev\Enigma_Studio_3_-_Transplant_Edition\code\eshared\opstacking\miscops.cpp
 8   eLightOp;C:\dev\Enigma_Studio_3_-_Transplant_Edition\code\eshared\opstacking\modelops.cpp
 9   eModelMergeOp;C:\dev\Enigma_Studio_3_-_Transplant_Edition\code\eshared\opstacking\modelops.cpp
10   eModelTransformOp;C:\dev\Enigma_Studio_3_-_Transplant_Edition\code\eshared\opstacking\modelops.cpp
11   eModelModelOp;C:\dev\Enigma_Studio_3_-_Transplant_Edition\code\eshared\opstacking\modelops.cpp
12   eMeshTransformOp;C:\dev\Enigma_Studio_3_-_Transplant_Edition\code\eshared\opstacking\meshops.cpp
13   eMeshMultiplyOp;C:\dev\Enigma_Studio_3_-_Transplant_Edition\code\eshared\opstacking\meshops.cpp
14   eMeshMergeOp;C:\dev\Enigma_Studio_3_-_Transplant_Edition\code\eshared\opstacking\meshops.cpp
```

Save the file and then run „eplayercppcollector.bat" in the binary directory.
This should show you this nice result without any errors. If you see a problem here you might not have installed JRE correctly or maybe the paths above are still not correct.

```
C:\dev\Enigma_Studio_3_-_Transplant_Edition\binary>java -jar ./eplayercppcollect
or.jar

C:\dev\Enigma_Studio_3_-_Transplant_Edition\binary>pause
Drücken Sie eine beliebige Taste . . .
```

Now it is time to fire up Visual Studio by double clicking /code/enigma3.sln
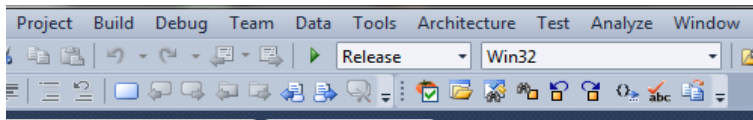
Step 2: Compiling the .exe

Step 2.1: Modifying production.hpp

Open up production.hpp and modify the lines 20/21 as seen in the screenshot below to point to the 2 .h files you exported ealier. For transplant this should already be correct.
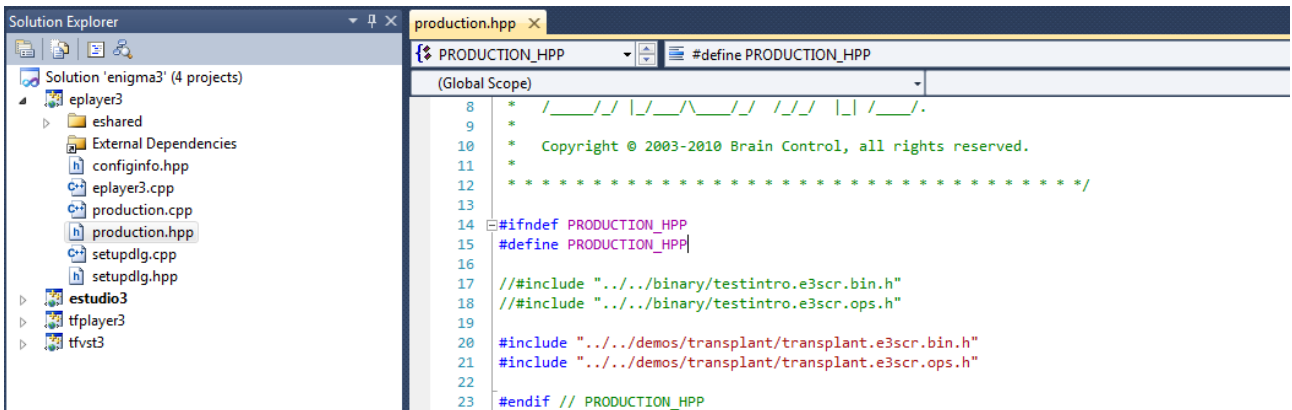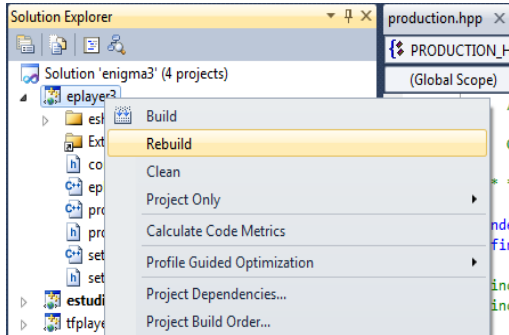
Step 2.2: Compiling

Check that the project is set to „Release" not „Debug"

Now you just compile the project by clicking „Rebuild" on the eplayer3 project:

After that ran through, you should have an „eplayer.exe" in your binary directory. Running „pack_eplayer3.bat" will compress that and leave you with the following (or similar) screen and a file „eplayer3_packed.exe" that will hopefully be smaller than 64k :)

Hope this all worked fine for you, if not you can message me at payne@braincontrol.org

Enjoy!