
Subject: PLATFORM_MACOSX vs PLATFORM_OSX11
Posted by [daveremba](#) on Sat, 18 Jun 2011 21:59:29 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hello,

background:

I've been working on a port of U++ for the current MacOSX 10.6.7 on Intel 64 bit iMac hardware. I think an alpha version will be ready for testing soon (1-2 weeks maximum).

The current U++ code base was very close to working (it needed only minor changes).

The port I did uses X11 rather than the native MacOSX UI, and therefore looks a little different from native Mac apps, but seems to work fine.

a proposal:

I'd like to suggest changing the current OSX11 flag to MACOSX. Here is why:

1. I think OSX11 is a little confusing - it might mean any OS that has X11, or it might mean OSX ... etc.
2. The OS on Mac is formally called MACOSX, and X11 is one of its components.

Also, in current MacOSX 10.6.7 on Intel 64 with gcc there are some system flags that may be a little different that the current U++ expects:

```
__linux is not defined  
__APPLE__ is defined
```

implementation:

to implement the flag change that is proposed:

file: Core/config.h would now read:

...

```
#if __linux
#define PLATFORM_LINUX 1
// added:
#elif __APPLE__
#define PLATFORM_MACOSX 1
#define PLATFORM_BSD 1
#else
// removed APPLE from below
...

```

The other files that PLATFORM_MACOSX would effect for this port are:

```
Core/Core.h
Core/Cpu.cpp
CtrlCore/CtrlCore.upp
Draw/Draw.upp
ide/Builders/GccBuilder.cpp
ide/Builders/MakeFile.cpp
ide/Core/Host.cpp

```

only a few lines in each file;
so, not very much.

other changes:

1. a config file for Ndisasm on MacOSX/Intel64/gcc was added, and a Makefile flag need

2. To fix a UI bug, there was a change made to:
CtrlCore/CtrlKbd.cpp
Ctrl::SetFocus0(bool activate)

based on another thread:
[BUG?] X11 (at least OSX) Menus displayed under TopWindow, owner=RootWindow [message #30902]

I added the suggested code, wrapped inside PLATFORM_OSX11, so other platforms are not affected.

Thank you to Honza, and all the others who worked to get that problem under control.

poll & feedback from senior developers

After I get clarification on this, I'll get a version of the U++ MACOSX port available for anyone who would like to test it.

Thanks,

Dave
