Subject: Re: Pre processor and macro error
Posted by Klugier on Mon, 30 Nov 2020 20:54:05 GMT
View Forum Message <> Reply to Message

Hello Xemuth,

Please noticed that GLFW uses CMake as a build system. One of the initial step of this build to is to generate platform specific build basing on CMakeList.txt files. So, I think the flags are generated in this steps, so in our case we need to workaround it if we would like to port it our build system.

So, in your case, all you need to do is to add dependency to upp/Core and in file when flags are defined add something like that:

```
#include <Core/config.h>

#if defined(PLATFORM_WIN32)
 #define _GLFW_WIN32 1
#elif defined(PLATFORM_COCOA)
 #define _GLFW_COCOA 1
#elif defined(PLATFORM_POSIX)
 #define _GLFW_X11 1
#elif defined(flagWAYLAND) // <- Not supported yet, however you could still pass it as package specific flag
 #define _GLFW_WAYLAND 1
#elif define(flagOMESA) // <- Not supported yet - I don't know what OMESA is? Is it some kind of software rendering?
 #define _GLFW_OMESA 1
#endif
```

Backing to Wayland here is the documentation how to distinguish it on GTK side. Would be good to have it defined somewhere.

Also, if everything will work as expected please consider make the glfw package available in UppHub :)

If you do not want to use Core/config.h. You could port it like this:

```
#if defined(flagWIN32)
 #define _GLFW_WIN32 1
#elif defined(flagCOCOA)
 #define _GLFW_COCOA 1
#elif defined(flagX11)
 #define _GLFW_X11 1
#elif defined(flagWAYLAND) // <- Not supported yet, however you could still pass it as package specific flag
```

#define _GLFW_WAYLAND 1
#elif define(flagOMESA) // <- Not supported yet - I don't know what OMESA is? Is it some kind of software rendering?
#define _GLFW_OMESA 1
#endif

And then GLFW should accepts WIN32 COCOA X11 WAYLAND OMESA and in the package (application) that use that package all you need to do is manually specify flag in "Main package configuration(s)" dialog.

Klugier
