Subject: Middle-ground between CONSOLE_APP_MAIN and GUI_APP_MAIN
Posted by cbpporter on Fri, 05 Nov 2010 10:00:17 GMT
View Forum Message <> Reply to Message

I was wondering how could I achieve a middle ground between these two modes, basically a CONSOLE_APP_MAIN that expands to WinMain or alternatively a GUI application with a GUII_APP_MAIN that does not create any windows and does not use CtrlCore. I do not want to have a console window.

How does TheIDE control the option for creating a console app or not based on the absence or presence of the GUI flag.

A week ago I hacked this together and it works fine except for the acquisition of arguments, but I would like to apply a proper solution:

```
#ifdef PLATFORM_WIN32

void AppInit__(int argc, const char **argv);
void AppInitEnvironment__();

#define CONSOLE_APP_MAIN2 \
void ConsoleMainFn_(); \
 \
int APIENTRY WinMain(HINSTANCE hInstance, HINSTANCE, LPTSTR lpCmdLine, int nCmdShow) \
{ \
 UPP::AppInitEnvironment__(); \
 ConsoleMainFn_(); \
 UPP::DeleteUsrLog(); \
 UPP::AppExit__(); \
 return UPP::GetExitCode(); \
} \
 \
void ConsoleMainFn_()

#endif

#ifdef PLATFORM_POSIX

void AppInit__(int argc, const char **argv, const char **envptr);

#define CONSOLE_APP_MAIN2 \
void ConsoleMainFn_(); \
 \
int main(int argc, const char **argv, const char **envptr) { \
 UPP::AppInit__(argc, argv, envptr); \
 ConsoleMainFn_(); \
 UPP::DeleteUsrLog(); \
 UPP::AppExit__(); \
 return UPP::GetExitCode(); \
} \
 \
void ConsoleMainFn_()

#endif
```

Ideally, is there a way to have you application not open o console, but when it is opened from a console it still writes to it. If I compile it with GUI, i get no output, even if opened from cmd. If I compile it without, I get the output, but when opening it from the shell I get a new console window.