Subject: Re: Building & using U++ without TheIDE
Posted by sergei on Thu, 13 Sep 2007 13:43:00 GMT
View Forum Message <> Reply to Message

I'm starting to see the big picture. There are 2 options:

1) Compiler decides what goes in and what doesn't.
2) User decides what goes in and what doesn't.

I wanted to go for option 1 - that way, if user uses PDF, PDF goes in. I understand that U++ works according to option 2 - what's in, is used. That is actually easier to implement.

My current idea goes like this:

1) Take the source, create a project (in any compiler/IDE) for a static library.
2) Add all files in all packages (according to .upp to exclude forgotten/unused files, and maybe exclude TheIDE-only packages).
3) Remove .icpps from the project (they will not be compiled).
4) Build debug/release/whatever libs.
5) Make a new project (that will use U++).
6) Include main headers of packages used, and icpps of packages used (only once, into one of the project's cpps).
7) Build the project - linker will throw out unused packages, and initialize used packages since the icpps were included.

I'd like to automate this process. I'll try to make a program to scan folder structure, parse .upps, create static lib project (e.g. include only used files), and create package headers.

By package headers, I mean it will be a header that handles icpps and dependencies inside.
Example (upppkg/CtrlLib.h):

```
#ifndef UPPPKG_CTRLLIB_H
#define UPPPKG_CTRLLIB_H

// Uses
#include <upppkg/CtrlCore.h>
#include <upppkg/RichText.h>

// Uses (platform)
#ifdef flagLINUX
#include <upppkg/PdfDraw.h>
#endif
#ifdef flagFREEBSD
#include <upppkg/PdfDraw.h>
#endif
#ifdef flagOSX11
#include <upppkg/PdfDraw.h>
#endif
```

```
// Header
#include <CtrlLib/CtrlLib.h>

// ICPP
#include <CtrlLib/CtrlLib.icpp>

#endif
```

Linking to libUpp and including this file should basically provide a similar environment as TheIDE provides, when you write #include <CtrlLib/CtrlLib.h> and add this package.

What do you think of this idea? It would take a second to generate headers from U++ source, another ~20 mins to build the library, and U++ is ready to use.

Questions:
1) Is the first header in file section of .upp always the most important one of that package? If not, how can the main header be determined?
2) I've found files with other extensions (not h/hpp/c/cpp/icpp) that maybe should be handled somehow - .dli, .iml, .in, .lay, .patch, .t, .upt, .usc, .vc. How should I take care of these?
3) Having a static lib + correct includes, there should be no problem using them in any project - exe/dll/lib, right? I've seen in another thread that there are problems with using U++ DLL in U++ EXE - wouldn't static linking each to U++ just work (OK, 1MB or so wasted, but still)?

P.S. I've tried to start making that parser, encountered 3 problems:
1) ToUnixName is implemented in Path.cpp but not defined in Path.h - can't use it.
2) I didn't find anything looking like this in the sources:

```
#ifdef PLATFORM_WIN32
const char cDirSep = '\\';
#else
const char cDirSep = '/';
#endif
```

Is there any reason for constantly using the chars '\\' and '/' and checking the OS?
3) I don't understand how unicode is implemented. There is String, AString, WString, but there is no TString, or whatever the name, like there is TCHAR that expands into char or wchar_t, depending on whether UNICODE/_UNICODE is defined. How do I define whether I'm in unicode or not? I mean, MessageBox will expand into MessageBoxA or MessageBoxW? And why path handling routines use char - can I handle unicode filenames with U++?