Subject: Win_CE
Posted by arturbac on Fri, 06 Jul 2007 21:06:41 GMT
View Forum Message <> Reply to Message

Code needs correction
add in Link in MSCBuilder
-NODEFAULTLIB:\"oldnames.lib\"

like :
```
if(HasFlag("MSC8ARM"))
link <<  " -subsystem:windowsce,4.20 /ARMPADCODE -NODEFAULTLIB:\"oldnames.lib\" ";
```

---

correct the line with GUI flag
adding || HasFlag("MSC8ARM")

```
if(HasFlag("GUI") || HasFlag("MSC8ARM"))
link << (HasFlag("WIN32") ? " -subsystem:windows" : " -subsystem:windowsce");
  else
   link << " -subsystem:console";
```

And I added protetic fix for Unicode only on WCE in Core/log.cpp

```
#if defined(PLATFORM_WIN32)
#if defined(PLATFORM_WINCE)
 wchar_t pwcs[512];
 mbstowcs(pwcs, backup, strlen(backup));
 DeleteFile(pwcs);
#else
 DeleteFile(backup);
#endif

#elif defined(PLATFORM_POSIX)
 unlink(backup);
#else
 #error
#endif

#if defined(PLATFORM_WIN32)
#if defined(PLATFORM_WINCE)
 wchar_t wfilename[512];
 mbstowcs(wfilename, filename, strlen(filename));
 MoveFile(wfilename, pwcs);
#else
 MoveFile(filename, backup);
#endif
#elif defined(PLATFORM_POSIX)
```

```
 !rename(filename, backup);
#else
 #error
#endif
```

And this one helps digging problems

Add after PutConsole("Linking...");
line PutConsole(link);

Subject: Re: Win_CE
Posted by arturbac on Fri, 06 Jul 2007 21:36:58 GMT
View Forum Message <> Reply to Message

Next part

Win32DnD.cpp

exclude from PPC CF_HDROP
like

```
#ifndef PLATFORM_WINCE
 if(cf == CF_HDROP)
  return "files";
#endif
```

correct

```
static int CF_PERFORMEDDROPEFFECT = RegisterClipboardFormat(_T("Performed
DropEffect"));
```

Threads also dosn't compile on PoketPC

Subject: Re: Win_CE
Posted by mirek on Fri, 27 Jul 2007 11:25:11 GMT
View Forum Message <> Reply to Message

Quote:
Add after PutConsole("Linking...");
line PutConsole(link);

You can achieve that (and more.. by activating Verbose in Setup menu.

Mirek

Subject: Re: Win_CE
Posted by mirek on Fri, 27 Jul 2007 11:29:36 GMT
View Forum Message <> Reply to Message

Patches applied.

BTW, does it mean you have achieved to compile&run somthing on PDA?

Mirek

Subject: Re: Win_CE
Posted by arturbac on Sat, 28 Jul 2007 19:45:32 GMT
View Forum Message <> Reply to Message

Some days ago i compiled one app but
Startup of this app was so long so i forget about testing it more.
It was a simple window app nothing more.
Propably i will use in the future Core(NTL, types) on PPC.

Subject: Re: Win_CE
Posted by shalom on Sat, 04 Aug 2007 21:58:19 GMT
View Forum Message <> Reply to Message

I don't know what compilers you people were using, and exactly what you managed to do.
I am using the Mingw32 version of cegcc (i.e. arm-wince-mingw32ce-gcc)
I tried to compile an example (using the IDE) - and I started with compilation errors 'cAlternateFileName not defined'. I added -DPLATFORM_WINCE to the compile options, and that went away.

This definition needs to be added automatically when WINCE is chosen as the platform.

Now I have
CtrlLib\RichTextView.cpp:458: error: 'PrinterJob' was not declared in this scope
I suggest that this function be #ifdef'ed out (and replaced with a stub) on WINCE.

Those are the only problems in the CtrlLib.
There are a few problems with the CtrlCore too.

In general - do these problems appear with other WinCE compilers?

Is anyone else using cegcc?
Any advice?

Subject: Re: Win_CE
Posted by shalom on Sun, 05 Aug 2007 01:33:39 GMT
View Forum Message <> Reply to Message

Using the same compiler as above, I had to change the sDumpWindow function in Win32Wnd.cpp:

```
static BOOL CALLBACK sDumpWindow(HWND hwnd, LPARAM lParam) {
 String dump;
 dump << (IsWindowEnabled(hwnd) ? "ena" : "dis") << ' '
      << (IsWindowVisible(hwnd) ? "vis" : "hid") << ' '
      << Sprintf("owner=0x%x ", GetWindow(hwnd, GW_OWNER));
 Ctrl *ctrl = Ctrl::CtrlFromHWND(hwnd);
 if(ctrl) {
#ifdef _DEBUG
  dump << "Ctrl: " << UPP::Name(ctrl);
#endif
 }
 else if(!lParam)
  return TRUE;
 else
 {
#ifdef PLATFORM_WINCE
  wchar clsname[256], title[1024];
#else
  char clsname[256], title[1024];
#endif
  ::GetClassName(hwnd, clsname, __countof(clsname));
  ::GetWindowText(hwnd, title, __countof(title));

#ifdef PLATFORM_WINCE
 WString wClsName = clsname;
 WString wTitle = title;
  dump << "HWND: " << Sprintf("0x%x", hwnd) << ", class = "
       << wClsName << ", title = " << wTitle;
#else
  dump << "HWND: " << Sprintf("0x%x", hwnd) << ", class = "
       << clsname << ", title = " << title;
#endif
 }
 LLOG(dump);
 return TRUE;
}
```

I added the second #ifdef. The
I am surprised you didn't have problems with this - as apparently there is no AsString for wchar * since the template version uses ToString, and wchar is a primitive type. There is also a non-template version for PLATFORM_MSC that calls FormatPtr - for some reason I couldn't seem to use this (since the compiler decides that the template version is closer for wchar *).

Now I have a small problem with the compiler itself - after that I think it will all compile correctly.

Subject: Re: Win_CE
Posted by arturbac on Sun, 05 Aug 2007 08:47:22 GMT
View Forum Message <> Reply to Message

shalom wrote on Sat, 04 August 2007 23:58I don't know what compilers you people were using, and exactly what you managed to do.
I am using the Mingw32 version of cegcc (i.e. arm-wince-mingw32ce-gcc)

1. Microsoft (R) C/C++ Optimizing Compiler Version 14.00.60131 for ARM
2. Code generated by mingw for win32 can be 50 times slower than by other avaialbe compilers for win32(just try for example ntbench). So using mingwce for WinCe propably is mistake...

Subject: Re: Win_CE
Posted by shalom on Sun, 05 Aug 2007 13:27:29 GMT
View Forum Message <> Reply to Message

The reason I am using this compiler is that it is just about the only free compiler around. I have EVC++ 3 and 4 - but I understand that UPP doesn't support these (quite rightly - the template support is at the level of the old MSVC 6).

The compiler is actually hosted (and compiled) under Cygwin.
Unfortunately I don't have anything to compare it with, unless I try to compile some of the EVC++ applications with it.

Anyway I can now compile all the 'clock' example, but I am stuck with linking. The IDE insists on adding the following libraries to the link command line, and I am working on that right now.

-ladvapi32 -lshell32 -lwinmm

By the way, the linker also complains that the libraries have no indexes, so I ran ranlib 'by hand' - which I can live with, though I expect there's a way of adding in it somewhere.

Subject: Re: Win_CE
Posted by arturbac on Sun, 05 Aug 2007 14:13:12 GMT
View Forum Message <> Reply to Message

shalom wrote on Sun, 05 August 2007 15:27I have EVC++ 3 and 4

Propably evc4 sp4 is better choice than mingwce , hoever evc4 doesn't support full c++ standard and there could be problems for example with templates.

Subject: Re: Win_CE
Posted by mirek on Sun, 05 Aug 2007 15:18:45 GMT
View Forum Message <> Reply to Message

shalom wrote on Sat, 04 August 2007 17:58I don't know what compilers you people were using, and exactly what you managed to do.
I am using the Mingw32 version of cegcc (i.e. arm-wince-mingw32ce-gcc)
I tried to compile an example (using the IDE) - and I started with compilation errors 'cAlternateFileName not defined'. I added -DPLATFORM_WINCE to the compile options, and that went away.

This definition needs to be added automatically when WINCE is chosen as the platform.

Now I have
CtrlLib\RichTextView.cpp:458: error: 'PrinterJob' was not declared in this scope
I suggest that this function be #ifdef'ed out (and replaced with a stub) on WINCE.

Those are the only problems in the CtrlLib.
There are a few problems with the CtrlCore too.

In general - do these problems appear with other WinCE compilers?
Is anyone else using cegcc?
Any advice?

These are very good news. In fact, we planned full support for cegcc, even distributing U++ with it, but so far this has not gone anywhere (too much else to do).

If you would succeed here, the path would be open.

The only WinCE tests we did was with VS C++ 2005 compiler. I have succeeded to compile and run most of "examples" - it has looked a bit weird (we will need to make special chameleon branch for WinCE), but worked (with some crashing

Mirek

Subject: Re: Win_CE
Posted by shalom on Mon, 06 Aug 2007 06:52:28 GMT
View Forum Message <> Reply to Message

Since there is some interest, I will try to document what I have done in this thread.

Use of advapi32.lib etc: I added !GNU_ARM to the 'when' clause of that line in the various packages.

Running ranlib: This can be added to the custom build steps as 'post-link':
arm-wince-mingw32ce-ranlib.exe $(PATH)

The current problem is that I have undefined externals LocalLock and LocalUnlock (and a couple of others). These come mainly from the clipboard code in Win32Wnd.cpp.

There is no LocalLock/Unlock (at least up to CE 4.2) however the Microsoft documentation indicates that GetClipboardData returns a handle. I have never tried the clipboard functions in WinCE, so I don't know if the handle is really a pointer or not.

- Shalom Crown