Pardon me if this is not the right section to post this and please redirect me to te corret one.

Long time I didn't post anything... anyway.

I just got the new 2008.1 release of upp and bumped into some problems. (be prepared for a very long post)

I was using upp 2007.1 and had some programs that cross compiled with no problems between win32,linux,freebsd and wince. And when using the new version of upp the wince port can no longer be compiled as vc9.0 trows a bunch of errors.

I'm here hoping you can help me fix most(or all) of them, as I wouldn't want to go back to the previows upp version until there is no fix for this. So here are the errors:

First of all:
-TheIDE doesn't AutoSetup the MSC9ARM compiler option: No problem, I used my previous bm and updated.

-Upp doesn't recognize the new MSC9ARM flag: Every #ifdef macro directive (or at least most of them), that detects for ARM compilation in upp just recognices MSC8ARM flag, so when using MSC9ARM compilation goes on as if it wasn't. No problem, changed the compiler option to MSC8ARM.

-Then the first error appeared as process.h isn't in the include directories for wince in vs9, so I added the sdk path to it and then it found the process.h header but some other errors appeared that said there were duplicated methods. Had to put that #include inside a macro to not process it while using wince. I think this is the root of many problems, but not all.
C:\upp\uppsrc\Core\Core.h(343):

//CAMBIO WINCE
  #ifndef PLATFORM_WINCE
  #include <process.h>
  #endif

-I had to fix some files from the upp source as follows:
C:\upp\uppsrc\Core\App.cpp(17):
//CAMBIO WINCE
String GetHomeDirectory() {
 //CAMBIO WINCE
 #ifndef PLATFORM_WINCE
 return FromSystemCharset(getenv("HOMEDRIVE")) +
     FromSystemCharset(getenv("HOMEPATH"));
 #else

```
 return ".";
 #endif
}
```

There is no getenv function for wince, I think there should be a replacement somewhere but I couldn't find it (not that I looked so much as I didn't need this feature).

C:\upp\uppsrc\Core\Log.cpp(171):

```
void LogStream::Flush()
{
 int count = (int)(p - buffer);
 if(count == 0) return;
 //CAMBIO WINCE
 #ifndef PLATFORM_WINCE
 if(options & LOG_COUT)
  Cout().Put(buffer, count);
 if(options & LOG_CERR)
  Cerr().Put(buffer, count);
 #endif
#ifdef PLATFORM_WIN32
 if(options & LOG_FILE)
  if(hfile != INVALID_HANDLE_VALUE) {
   dword n;
   WriteFile(hfile, buffer, count, &n, NULL);
  }
 if(options & LOG_DBG) {
  *p = 0;
  //CAMBIO WINCE
  #ifndef PLATFORM_WINCE
  ::OutputDebugString((LPCSTR)buffer);
  #endif
 }
#else
...
```

There is no definition for Cout(), Cerr() nor OutputDebugString (sorry didn't confirm this last one), for WINCE.

C:\upp\uppsrc\Core\Mt.cpp(55):

```
#ifdef CPU_64
unsigned int
#else
//CAMBIO WINCE
#ifdef PLATFORM_WINCE
dword __stdcall
```

```
#else
uintptr_t __stdcall
#endif
```

...

This one was because the next error.

C:\upp\uppsrc\Core\Mt.cpp(92):

```
bool Thread::Run(Callback _cb)
{
 AtomicInc(sThreadCount);
 threadr = true;
 Detach();
 Callback *cb = new Callback(_cb);
#ifdef PLATFORM_WIN32
 //CAMBIO WINCE
 //unsigned thread_id;
 //handle = (HANDLE)_beginthreadex(0, 0, sThreadRoutine, cb, 0, &thread_id);
 dword thread_id;
 handle = (HANDLE)CreateThread(0, 0, sThreadRoutine, cb, 0, &thread_id);
#endif
#ifdef PLATFORM_POSIX
```

...

_beginthreadex is not a WINCE function and had to change to CreateThread.

C:\upp\uppsrc\Core\Mt.cpp(295):

```
bool Mutex::TryEnter()
{
 //CAMBIO WINCE
 #ifndef PLATFORM_WINCE
 if(!sTec) {
  if(HMODULE hDLL = LoadLibrary("Kernel32"))
   sTec = (TEC) GetProcAddress(hDLL, "TryEnterCriticalSection");
 }
 #else
 if(!sTec) {
  if(HMODULE hDLL = LoadLibrary(_T("Kernel32")))
   sTec = (TEC) GetProcAddress(hDLL, _T("TryEnterCriticalSection"));
 }
 #endif
```

...

Here, the LoadLibrary and GetProcAddress for WINCE need wchar strings, so I used the _T macro so the compiler fixed the issue.

C:\upp\uppsrc\Core\NetNode.cpp(40):

```cpp
void NetNode::SetPtrs()
{
 //CAMBIO WINCE
 #ifndef PLATFORM_WINCE
 SetPtr(local, net.lpLocalName);
 SetPtr(remote, net.lpRemoteName);
 SetPtr(comment, net.lpComment);
 SetPtr(provider, net.lpProvider);
 #else
 SetPtr(local, (char *&)net.lpLocalName);
 SetPtr(remote, (char *&)net.lpRemoteName);
 SetPtr(comment, (char *&)net.lpComment);
 SetPtr(provider, (char *&)net.lpProvider);
 #endif
}
```

This ain't the correct fix, but I post it to show that NETRESOURCE for WINCE has wchar strings, and local, remote, comment and provider are regular Strings.

C:\upp\uppsrc\Core\NetNode.cpp(85):

```cpp
 //CAMBIO WINCE
 #ifdef PLATFORM_WINCE
 const wchar *s=sn.lpRemoteName;
 #else
 const char *s = sn.lpRemoteName;
 #endif
```

Same issue as before, this had it's consecuences:

C:\upp\uppsrc\Core\NetNode.cpp(94):

```cpp
 //CAMBIO WINCE
 #ifndef PLATFORM_WINCE
 nn.local = n.lpLocalName;
 nn.remote = n.lpRemoteName;
 nn.comment = n.lpComment;
 nn.provider = n.lpProvider;
 #else
 nn.local = (WString().Cat()<<n.lpLocalName).ToString();
 nn.remote = (WString().Cat()<<n.lpRemoteName).ToString();
 nn.comment = (WString().Cat()<<n.lpComment).ToString();
 nn.provider = (WString().Cat()<<n.lpProvider).ToString();
 #endif
```

Don't know if (WString().Cat()<<n.lpLocalName).ToString() is the best way to do it, but It works... i

guess.

C:\upp\uppsrc\Core\NetNode.cpp(109):

```
 if(s[0] == '\\' && s[1] == '\\')
 //CAMBIO WINCE
 #ifdef PLATFORM_WINCE
 nn.name =
FromSystemCharset(DosInitCaps(GetFileName(~((WString().Cat()<<s).ToString())).ToWString())
;
 #else
 nn.name = FromSystemCharset(DosInitCaps(GetFileName(s)));
 #endif
```

Same as before GetFileName uses wchar strings on WINCE.

C:\upp\uppsrc\CtrlCore\Win32Clip.cpp(31):

```
 //CAMBIO WINCE
 #ifndef PLATFORM_WINCE
 if(fmt == "files")
  return CF_HDROP;
 #endif
```

CF_HDROP isn't defined for WINCE, didn't want to bother with it so I commented it out.

C:\upp\uppsrc\CtrlCore\Win32Wnd.cpp(25):

```
static bool PeekMsg(MSG& msg)
{
 if(!PeekMessage(&msg, NULL, 0, 0, PM_NOREMOVE)) return false;
 //CAMBIO WINCE
 #ifdef PLATFORM_WINCE
 return PeekMessageW(&msg, NULL, 0, 0, PM_REMOVE);
 #else
 return IsWindowUnicode(msg.hwnd) ? PeekMessageW(&msg, NULL, 0, 0, PM_REMOVE)
                     : PeekMessage(&msg, NULL, 0, 0, PM_REMOVE);
 #endif
}
```

There is no IsWindowUnicode for WINCE.

C:\upp\uppsrc\CtrlCore\Win32Wnd.cpp(635):

```
 //CAMBIO WINCE
 #ifdef PLATFORM_WINCE
 DispatchMessageW(&msg);
 #else
```

```
 if(IsWindowUnicode(msg.hwnd))
  DispatchMessageW(&msg);
 else
  DispatchMessage(&msg);
 #endif
```

Same as above.

C:\upp\uppsrc\CtrlLib\FileSel.cpp(25):

```
//CAMBIO WINCE
 #ifndef PLATFORM_WINCE
 if(!icon || !GetIconInfo(icon, &iconinfo))
  return Image();
 ...
 #else
 return Image();
 #endif
```

GetIconInfo ain't present on WINCE

C:\upp\uppsrc\CtrlLib\FileSel.cpp(1026):

```
...
//CAMBIO WINCE
#ifdef PLATFORM_WINCE
dir.Add(FromSystemCharset(GetWinRegString("Desktop", fs,
HKEY_CURRENT_USER).ToWString())));
#else
dir.Add(FromSystemCharset(GetWinRegString("Desktop", fs, HKEY_CURRENT_USER)));
#endif
...
```

FromSystemCharset is defined in Core/Util.h as requesting a WString on WINCE, while the same function for other OSs requires a simple String. Which leads me to the last parts I modified.

These where no errors but had to make some changes:
C:\upp\uppsrc\Core\Util.h (32):

```
#ifdef PLATFORM_WINCE
WString ToSystemCharset(const String& src);
String  FromSystemCharset(const WString& src);
WString ToSystemCharsetW(const char *src){return ToSystemCharset(String().Cat()<<src);};
String  FromSystemCharsetW(const wchar *src){return
FromSystemCharset(WString().Cat()<<src);};
#else
String  ToSystemCharset(const String& src);
String  FromSystemCharset(const String& src);
```

WString ToSystemCharsetW(const char *src);
String FromSystemCharsetW(const wchar *src);
#endif

I had to include ToSystemCharsetW and FromSystemCharsetW for WINCE, because there are
parts of the upp source that call them without first detecting the current compilation target.

And these are the last errors I can't fix, and I hope you can help me:

Quote:
----- Core (GUI MT MSC8ARM FORCE_SPEED WIN32 MSC) (2 / 12)
heaputil.cpp
c:\upp\uppsrc\core\heaputil.cpp(223) : error C2485: '$I' : atributo extendido no reconocido
c:\upp\uppsrc\core\heaputil.cpp(48) : error C2485: '$I' : atributo extendido no reconocido
c:\upp\uppsrc\core\heaputil.cpp(71) : error C2485: '$I' : atributo extendido no reconocido
c:\upp\uppsrc\core\heaputil.cpp(103) : error C2485: '$I' : atributo extendido no reconocido
c:\upp\uppsrc\core\heaputil.cpp(122) : error C2485: '$I' : atributo extendido no reconocido
sheap.cpp
c:\upp\uppsrc\core\sheap.cpp(327) : error C2485: '$I' : atributo extendido no reconocido
c:\upp\uppsrc\core\sheap.cpp(332) : error C2485: '$I' : atributo extendido no reconocido
c:\upp\uppsrc\core\sheap.cpp(337) : error C2485: '$I' : atributo extendido no reconocido
c:\upp\uppsrc\core\sheap.cpp(342) : error C2485: '$I' : atributo extendido no reconocido
c:\upp\uppsrc\core\sheap.cpp(347) : error C2485: '$I' : atributo extendido no reconocido
c:\upp\uppsrc\core\sheap.cpp(352) : error C2485: '$I' : atributo extendido no reconocido
c:\upp\uppsrc\core\sheap.cpp(357) : error C2485: '$I' : atributo extendido no reconocido
c:\upp\uppsrc\core\core.h(387) : error C2485: '$I' : atributo extendido no reconocido
heap.cpp
c:\upp\uppsrc\core\heap.cpp(240) : error C2485: '$I' : atributo extendido no reconocido
c:\upp\uppsrc\core\heap.cpp(245) : error C2485: '$I' : atributo extendido no reconocido

Besides:

-TheIDE on linux (I'm using Ubuntu 64bits) asks for gcc 4.1, I find it weird as the previous version
worked fine with the gcc 4.2 compiler. But I think I found the problem, compilation with 4.2
sometimes makes TheIDE crash with a memory access error, and not only that, a program I'm
trying to compile with it can no longer start because it crashes with the invalid memory access
error, when compiled with the speed option, this didn't happen with the previous upp version
though. I have to use the Optimal or Debug setting, options that I don't like because I need the
best speed I can get.

-On a related issue, when installing upp on FreeBSD (64bits) from ports, I had set the CFLAGS
and CXXFLAGS to "-O3 -pipe -march=nocona -mfpmath=sse -msse -msse2 -ss3", and compiled
(gcc 4.2.1) and installed, the initial setup couldn't finish as it crashed with a invalid memory access
error, thing that didn't happen with the previous upp version. I had to coment out the CFLAGS and
CXXFLAGS, recompile and reinstall, and then TheIDE would run fine.

I must mention that the target platform I refer with WINCE is PocketPC 2003 (Maybe that's the source of all problems), and these are the bms I'm using:
MSC8ARM.bm
Quote:
BUILDER = "MSC8ARM";
COMPILER = "";
DEBUG_INFO = "2";
DEBUG_BLITZ = "0";
DEBUG_LINKMODE = "0";
DEBUG_OPTIONS = "-Od";
DEBUG_FLAGS = "";
RELEASE_BLITZ = "0";
RELEASE_LINKMODE = "0";
RELEASE_OPTIONS = "-O2";
RELEASE_SIZE_OPTIONS = "-O1";
RELEASE_FLAGS = "";
DEBUGGER = "msdev";
PATH = "C:\\Archivos de programa\\Microsoft Visual Studio
9.0\\VC\\ce\\bin\\x86_arm;C:\\Archivos de programa\\Microsoft Visual Studio
9.0\\SmartDevices\\SDK\\SDKTools;C:\\Archivos de programa\\Microsoft Visual Studio
9.0\\Common7\\IDE;C:\\Enrike\\Programacion\\SDL\\SDL_1_2_6_W
inCE_05_30_04\\lib\\PPC2003 ";
INCLUDE = "C:\\Archivos de programa\\Microsoft Visual Studio 9.0\\VC\\ce\\include;C:\\Archivos
de programa\\Microsoft Visual Studio
9.0\\SmartDevices\\SDK\\PocketPC2003\\Include;C:\\Enrike\\Pr
ogramacion\\SDL\\SDL-devel-1.2.12-mingw32\\SDL-1.2.12\\inclu de ";
LIB = "C:\\Archivos de programa\\Microsoft Visual Studio 9.0\\VC\\ce\\lib\\armv4;C:\\Archivos de
programa\\Microsoft Visual Studio
9.0\\SmartDevices\\SDK\\PocketPC2003\\Lib\\armv4;C:\\Enrike\
\Programacion\\SDL\\SDL_1_2_6_WinCE_05_30_04\\lib\\PPC2003 ";
REMOTE_HOST = "";
REMOTE_OS = "";
REMOTE_TRANSFER = "";
REMOTE_MAP = "";
LINKMODE_LOCK = "0";

MSC9ARM.bm
Quote:
BUILDER = "MSC9ARM";
COMPILER = "";
DEBUG_INFO = "2";
DEBUG_BLITZ = "0";
DEBUG_LINKMODE = "0";
DEBUG_OPTIONS = "-Od";
DEBUG_FLAGS = "";
RELEASE_BLITZ = "0";

```
RELEASE_LINKMODE = "0";
RELEASE_OPTIONS = "-O2";
RELEASE_SIZE_OPTIONS = "-O1";
RELEASE_FLAGS = "";
DEBUGGER = "msdev";
PATH = "C:\\Archivos de programa\\Microsoft Visual Studio
9.0\\VC\\ce\\bin\\x86_arm;C:\\Archivos de programa\\Microsoft Visual Studio
9.0\\SmartDevices\\SDK\\SDKTools;C:\\Archivos de programa\\Microsoft Visual Studio
9.0\\Common7\\IDE;C:\\Enrike\\Programacion\\SDL\\SDL_1_2_6_W
inCE_05_30_04\\lib\\PPC2003 ";
INCLUDE = "C:\\Archivos de programa\\Microsoft Visual Studio 9.0\\VC\\ce\\include;C:\\Archivos
de programa\\Microsoft Visual Studio
9.0\\SmartDevices\\SDK\\PocketPC2003\\Include;C:\\Enrike\\Pr
ogramacion\\SDL\\SDL-devel-1.2.12-mingw32\\SDL-1.2.12\\inclu de ";
LIB = "C:\\Archivos de programa\\Microsoft Visual Studio 9.0\\VC\\ce\\lib\\armv4;C:\\Archivos de
programa\\Microsoft Visual Studio
9.0\\SmartDevices\\SDK\\PocketPC2003\\Lib\\armv4;C:\\Enrike\
\Programacion\\SDL\\SDL_1_2_6_WinCE_05_30_04\\lib\\PPC2003 ";
REMOTE_HOST = "";
REMOTE_OS = "";
REMOTE_TRANSFER = "";
REMOTE_MAP = "";
LINKMODE_LOCK = "0";
```

I didn't have any problem whatsoever with the last version of upp (2007.1), besides little details with the
network implementation (which I didn't use at the end).

I hope this long post didn't bore you, and also hope that it turns out to be a little helpful.

Thank you for your patience, and for your help.

niofis

Subject: Re: Problems Compiling for WinCE (PPC2003) ARM
Posted by niofis on Wed, 03 Sep 2008 19:51:30 GMT
View Forum Message <> Reply to Message

no help?

Subject: Re: Problems Compiling for WinCE (PPC2003) ARM
Posted by mirek on Wed, 03 Sep 2008 21:26:19 GMT
View Forum Message <> Reply to Message

---

niofis wrote on Fri, 22 August 2008 16:02-TheIDE on linux (I'm using Ubuntu 64bits) asks for gcc 4.1, I find it weird as the previous version worked fine with the gcc 4.2 compiler. But I think I found the problem, compilation with 4.2 sometimes makes TheIDE crash with a memory access error, and not only that, a program I'm trying to compile with it can no longer start because it crashes with the invalid memory access error, when compiled with the speed option, this didn't happen with the previous upp version though.

This is a confirmed error in GCC 4.2 (we have reported it in GCC's bugzilla and it was confirmed). That is why we are asking for 4.1, which works just fine.

Quote:
I have to use the Optimal or Debug setting, options that I don't like because I need the best speed I can get.

Optimal should be as fast as Fast...

Quote:
-On a related issue, when installing upp on FreeBSD (64bits) from ports, I had set the CFLAGS and CXXFLAGS to "-O3 -pipe -march=nocona -mfpmath=sse -msse -msse2 -ss3", and compiled (gcc 4.2.1) and installed, the initial setup couldn't finish as it crashed with a invalid memory access error, thing that didn't happen with the previous upp version. I had to coment out the CFLAGS and CXXFLAGS, recompile and reinstall, and then TheIDE would run fine.

IMO, the same thing. GCC 4.2 simply has serious bugs when compiling with -O3.

Quote:
I must mention that the target platform I refer with WINCE is PocketPC 2003 (Maybe that's the source of all problems), and these are the bms I'm using:

Well, I have to admit we have abandoned PocketPC for a while (nobody was using it, so no bug or problem reports etc...)

I guess it is time to restart efforts.

Have you tried that GCC version for PPC?

http://cegcc.sourceforge.net

At the moment, I do not like the idea of installing the whole visual studio just to get the commandline compiler. If I had working compiler, I could try to fix U++ at least to the 2007.1 level...

Mirek

Subject: Re: Problems Compiling for WinCE (PPC2003) ARM
Posted by niofis on Wed, 03 Sep 2008 23:05:59 GMT

View Forum Message <> Reply to Message

Thanks for your answer, sorry I had to pm you ,

I can help you with the WinCE development (as much as I can), If you want me to. By the way, you don't need the whole visual studio, embedded visual c++ 4 will just do the trick.

Anyway, I'll try the ppc version of gcc, and post here again.

Regards!

Subject: Re: Problems Compiling for WinCE (PPC2003) ARM
Posted by niofis on Fri, 05 Sep 2008 20:05:40 GMT

View Forum Message <> Reply to Message

No luck either, gcc_arm won't compile the whole sources .

Too many errors, basically because upp defines a lot of flags only when using MSC8ARM, and many places expect that.

niofis

Subject: Re: Problems Compiling for WinCE (PPC2003) ARM
Posted by MatthiasG on Fri, 20 Nov 2009 19:54:17 GMT

View Forum Message <> Reply to Message

Hello,

just found this old thread, and wanted to ask, if someone succeeded to use cegcc or any other free available compiler for win ce with upp till now?

greetings,
Matthias

Subject: Re: Problems Compiling for WinCE (PPC2003) ARM
Posted by koldo on Wed, 14 Apr 2010 08:26:29 GMT

View Forum Message <> Reply to Message

Hello Mirek

What is the status of U++ in Windows Mobile?

Subject: Re: Problems Compiling for WinCE (PPC2003) ARM
Posted by mirek on Wed, 14 Apr 2010 08:36:45 GMT
View Forum Message <> Reply to Message

koldo wrote on Wed, 14 April 2010 04:26Hello Mirek

What is the status of U++ in Windows Mobile?

Around 2004, I have suceeded to compile some examples and to run them on my WinCE handheld computer.

Then it became unmaintained. I would say it would require about week to get it working again.

I was basically stopped at lack of available free compiler.

If somebody could get CeGCC working in Win32 and suggest some decent setup (emulator etc..), I could try again.

One problem, though, is that I do not own any WinCE HW anymore...

Subject: Re: Problems Compiling for WinCE (PPC2003) ARM
Posted by koldo on Wed, 14 Apr 2010 13:59:07 GMT
View Forum Message <> Reply to Message

luzr wrote on Wed, 14 April 2010 10:36koldo wrote on Wed, 14 April 2010 04:26Hello Mirek

What is the status of U++ in Windows Mobile?

Around 2004, I have suceeded to compile some examples and to run them on my WinCE handheld computer.

Then it became unmaintained. I would say it would require about week to get it working again.

I was basically stopped at lack of available free compiler.

If somebody could get CeGCC working in Win32 and suggest some decent setup (emulator etc..), I could try again.

One problem, though, is that I do not own any WinCE HW anymore...

Hello Mirek

I have a Pocket PC 2003 PDA (still running).

Free eMbedded Visual C++ 4.0 SP4 supports until Windows CE 5.0 that I think it is similar to Windows Mobile 5. Actual version is 6.5.