

MFC: Windows CE compatible

strcpy_s

WinCE: vyzaduje další parametr - velikost
strcpy_s(char * _Dst, rsize_t _DstSize, const char * _Src);

Solution:

1. přidat parametr velikost – kompatibilní i s win xp
2. náhradit pomocí **strcpy**, který velikost nevyžaduje

CreateWaitableTimer

WinCE: Not supported.

Solution:

1. Sleep()
2. Vlastní thread s event a sleep

_T()

WinCE: je čistě Unicode
makro _T() je definováno jako
#define __T(x) L ## x
(#define __TEXT(x) L ## x)
(#define _T(x) __TEXT(x))

Solution:

1. Makro __TEXT(x) ale již zřejmě definováno je.
Stačí tedy vložit “**#define _T(x) __TEXT(x)**”

2. Definovat _T přímo a překládat podle

```
#ifndef _UNICODE
#define __T(x) L ## x
#else
#define __T(x) x
#endif
```

CreateFileA

WinCE: CreateFileA – není unicode -> nelze použít
makro CreateFile se překládá jako CreateFileW a tím pádem jsou vstupní parametry TCHAR

Solution:

1. použít **CreateFile** jako unicode. Příklad fungující pod XP a CE
TCHAR deviceName[32];
swprintf_s(deviceName, sizeof(deviceName), _T("COM%d:"), port);
urt->ComFile = CreateFile(deviceName,

Overlapped

WinCE: Not supported.

WaitForMultipleObjects

WinCE: wait for all objects is not supported.

Příklad:

```
WaitForMultipleObjects( 3, p->hObject, FALSE /*This parameter must be set to FALSE*/, 1000 )
```

Queue set - Nastavení delky přijímací/vysílací fronty

WinCE: compiling OK, but run error

```
success = SetupComm( urt->ComFile,
                    0,          // InQueue
                    4096       // OutQueue
);

if (! success)
{
    UartFinalize( urt );
    return false;          // WinCE vrátí chybu
}
```

Solution:

Zatím neřešeno. Ponechána defaultní velikost fronty.

Cesty k souborům:

WinCE: Nejsou relativní cesty. Vše je vztaženo z kořenového adresáře

Solution:

1. Definovat cesty absolutně
2. Použít funkci pro získání aktuálního adresáře.

std::ostringstream, std::wostringstream - Now in new SDK is supported OK

WinCE: Tyto funkce využívají výjimky, jejichž podpora není funkční.

Při linkování se hlásí chyby typu:

```
error LNK2019: unresolved external symbol "public: __thiscall std::exception::exception(char const *)
```

Solution:

1. Vypnutí používání výjimek: `#define _HAS_EXCEPTIONS 0`
 - dojde sice ke zmenšení množství chyb, ale problém není zatím zcela vyřešen

Vyřešeno:

Nová kompilace Windows CE s podporou RTTI (Runtime Type Information) a AYGShell a následné vytvoření nové SDK.

pzn.:

Host: russellk (Microsoft)

Q: When I compile a eVC C++ project with the /GX option I receive the following error: Lnk2001 unresolved external symbol "void __stdcall `eh vector destructor iterator'(void *,unsigned int,int,void (__thiscall*)(void *))" When I remove the virtual des

A: Your OS Image must be built with SYSGEN_CPP_SEH and SYSGEN_RTTI to support /GX

Host: russellk (Microsoft)

Q: RE: the LNK err with the /GX option, "void __stdcall `eh vector destructor iterator' this error happens with the emulator. Why does this error occur in the emulator?

A: The OS image was not built with the SYSGEN_RTTI (Runtime Type Information) option. You will need to rebuild the image and include this feature for your application to link. The `eh vector destructor iterator' will be exported from COREDLL.

Grafika

CreateRectRgn

WinCE: Not supported.

Solution:

1. zkuste vytvořit region jinými funkcemi (CRgn je podporován)
- 2.

```
dc.SetBkMode(TRANSPARENT);  
dc->RoundRect( r_rnd, pt );
```

SetBkMode(TRANSPARENT)

WinCE: Nedochozí k překreslení pozadí rodičovským oknem. (pod prvkem se nekreslí podklad – tedy v podkladu se objevuje grafika i z jiných aplikací)

Solution:

1. Při zavolání OnPaint() u rodiče (majitele podkladu) na začátku před CPaintDC dc(this) skrtýt prvek s průhledným pozadím funkcí m_pruhlednyPrvek.ShowWindow(SW_HIDE) a po nakreslení pozadí znovu zavolat m_pruhlednyPrvek.ShowWindow(SW_SHOWNORMAL)
2. Vykreslování pozadí použitím např. BitBlt(...) před vlastním kreslením prvku

Pole tříd ovozených z MFC - Now in new SDK is supported OK

problém:

nejde deklarovat např.

```
CStatic m_Text[5];
```

Solution:

1. funguje CStatic *m_Text[5] či CStatic m_Text1, m_Text2,....

pzn.: Vytvoření pole vlastních tříd funguje.

Vyřešeno:

Viz. výše – vytvoření nové SDK pro platformu VIA Epia-M

Dialogs

Neklientské zprávy

Zprávy typu WM_NC.... Not supported., except WM_NCDESTROY.

Naopak např. zpráva ON_WM_MOUSEMOVE() je zasílána oknu i v případě, že se kurzor pohybuje po prvcích které okno obsahuje.

Solution:

1. try PossiblyEatMessage(...) ?

WINDOWPLACEMENT

WinCE: Not supported.

```
WINDOWPLACEMENT posDlg;  
GetWindowPlacement( &posDlg );  
LONG x = posDlg.rcNormalPosition.left;
```

Solution:

1. Ke zjištění polohy okna lze použít např
CRect r_wnd;
GetWindowRect(r_wnd);
int x = r_wnd.left;

2. Ke zjištění velikosti
CRect r_wnd;
GetClientRect(r_wnd);
int sizeX = r_wnd.Height();

Moving the our dialog by clicking on anywhere

I added one more feature in this sample i.e. Moving the our dialog by clicking on anywhere on the window. This can be do by posting WM_NCLBUTTONDOWN message to HTCAPTION Handle this statement in OnLButtonDownMessage(...)

```
PostMessage(WM_NCLBUTTONDOWN,HTCAPTION,MAKELPARAM(point.x,point.y))
```

WinCE: WM_NCLBUTTONDOWN - Not supported!

Solution:

1. Nelze použít ani v API - WM_NCHITTEST. Provizorní použití OnLButtonDown(), OnMouseMove(), OnLButtonUp(). Má však své nevýhody.