Re: Windows CE BinaryCompression

Source:
http://www.tech−archive.net/Archive/WindowsCE/microsoft.public.windowsce.app.development/2007−01/msg00078.

- *From*: "Wagner Bertolini Junior" <wagnerbertolini@xxxxxxxxxxxxxxxxxx>
- *Date*: Mon, 15 Jan 2007 11:49:01 −0300

Ok, now i understand it!

I do not think about it because at my corp the default system of the mobile devices is "Windows CE 5.0"

Thanks a lot.

"<ctacke/>" <ctacke[@]opennetcf[dot]com> escreveu na mensagem news:Of0sQnKOHHA.4260@xxxxxxxxxxxxxxxxxxxxxx

> If it must be portable, then it's certainly not the way to go. Use
> something like ZIP, GZIP or TAR. SharpZipLib is available and works fine.
> And it's a .NET interface.
>
> −−
> Chris Tacke
> OpenNETCF Consulting
> Managed Code in the Embedded World
> www.opennetcf.com
> −−
>
> "Wagner Bertolini Junior" <wagnerbertolini@xxxxxxxxxxxxxxxxxx> wrote in
> message news:OORizyJOHHA.1872@xxxxxxxxxxxxxxxxxxxxxx
>
> > I will really need a specific format, because i need to decompress this
> > file in another location sometimes.
> >
> > but i´m really interested on this API.
> >
> > See this article:
> >
> > Save Space Using Windows CE Built−in Compression API's
> > Jeff Zamora
> >
> > Buried inside the (undocumented) Windows CE API are BinaryCompress
> > and

BinaryDecompress. These functions offer a built−in way to quickly and easily compress data, and should meet all your data compression needs.

You can use BinaryCompress and BinaryDecompress to perform buffer compression. BinaryCompress will take a buffer and return a compressed buffer. BinaryDecompress will take a compressed buffer and turn it back into the original. BinaryCompress and BinaryDecompress are defined in coredll.

I assume the compression API uses the same compression as the object store. That compression is comparative − Windows CE compresses the buffer using several methods and takes the winner. And remember, all files kept in the object store are compressed, so this is redundant if you are just storing data in a file on your device. On the other hand, these APIs might be useful if you are saving data to a compact flash card, or if you are generating backup files that may end up back on the device.

Here's an example:

```
DWORD BinaryCompress(LPBYTE bufin, DWORD lenin, LPBYTE bufout,
DWORD
lenout);
DWORD BinaryDecompress(LPBYTE bufin, DWORD lenin, LPBYTE
bufout, DWORD
lenout, DWORD skip);

TCHAR string[] = _T("The quick brown fox jumps over the lazy man who
did
not come to the aid of his country in time.");
BYTE buffer[1024];
BYTE inpBuffer[1024];

int length = BinaryCompress((BYTE*)string, wcslen(string) *
sizeof(TCHAR), buffer, 1024);
BinaryDecompress(buffer, length, inpBuffer, 1024, 0);

AfxMessageBox((TCHAR*) inpBuffer);
CString message;
message.Format(_T("%i, %i"), wcslen(string), length);
AfxMessageBox(message);
```

In this example, the uncompressed buffer is 188 bytes and the compressed buffer is 148 bytes. Not the best, but it is free.

Coredll also contains StringCompress and StringDecompress. I imagine these are the same as their binary counterparts, except they stop compressing when they encouter a terminating \0.

"<ctacke/>" <ctacke[@]opennetcf[dot]com> escreveu na mensagem news:u1K5tm2NHHA.1252@xxxxxxxxxxxxxxxxxxxxxx

> You are going about this the completely wrong way. That's not what the
> function is for, and explaining how to get your data into some format
> that would be compatible in all cases would be long and counterproductive. Get a valid compression library or write something
> simple like RLE. You should not be P/Invoking for something like this.
>
> --
> Chris Tacke
> OpenNETCF Consulting
> Managed Code in the Embedded World
> www.opennetcf.com
> --
>
> "Wagner Bertolini Junior" <wagnerbertolini@xxxxxxxxxxxxxxxxx> wrote in message news:B5B8025E-5128-4C25-A038-DECE2F86E711@xxxxxxxxxxxxxxx
>
> > I am just trying to compress a file.
> >
> > Why, it doesn't work?
> >
> > I saw someone using it.. but he was using
> > C++ to do it.
> > But i think that is the same usage.
> >
> > About the Word Aligned, means that the
> > address of my pointer must be
> > aligned?
> > Now i get in trouble... I don´t know if i can
> > do that over .NetCF but i
> > need
> > to do a research....
> >
> > "<ctacke/>" wrote:
> >
> > > WORD aligned means it
> > > falls on a WORD divisible
> > > address (address % 2
> > > == 0).

DWORD aligned means it falls on a DWORD divisible address (address %4 == 0)

What exactly are you trying to do here anyway? I seriously doubt that you should be using this undocumented kernel API from an app, let alone a managed app.

––
Chris Tacke
OpenNETCF Consulting
Managed Code in the Embedded World
www.opennetcf.com
––

"Wagner Bertolini Junior" <wagnerbertolini@xxxxxxxxxxxxxxxxxx> wrote in message news:uYcWqroNHHA.5016@xxxxxxxxxxxxxxxxxxxxxx

Can anyone help me with this?

I found at winCe sources this function:

//––
//
Compressors − bufin and bufout must be WORD aligned, lenout must be at

least as
// large as
lenin,
CECOMPRESS_FAILED
is returned
if it doesn't
fit,
else
compressed
length
//
CECOMPRESS_ALL_ZEROS
returned if
buffer
entirely
zero. If
bufout
is
NULL, the
output
// length is
computed,
but the
results are
not stored.
Lenin must
be a
multiple of
2.
//−−

DWORD
NKBinaryCompress(
LPBYTE
bufin,
DWORD
lenin,
LPBYTE
bufout,
DWORD
lenout
)
{
DWORD
retval;

TRUSTED_API
(L"NKBinaryCompress",
CECOMPRESS_FAILED);

DEBUGMSG(ZONE_ENTRY,

(L"NKBinaryCompress
entry:
%8.8lx
%8.8lx
%8.8lx
%8.8lx\r\n",
bufin, lenin,
bufout,
lenout));

retval =
CECompress(bufin,
lenin,
bufout,
lenout, 1,
1024);

DEBUGMSG(ZONE_ENTRY,
(L"NKBinaryCompress
exit:
%8.8lx\r\n",
retval));

return
retval;
}

But i
CAN´T
understand
the
comment,
what is
WORD
aligned?

i'm trying to
use it and
i'm receving
an Error.. i
will post the
sample
code:

//DWORD
BinaryCompress(LPBYTE
bufin,
DWORD
lenin,
LPBYTE
bufout,

```
DWORD
lenout);
//DWORD
BinaryDecompress(LPBYTE
bufin,
DWORD
lenin,
LPBYTE
bufout,
DWORD
lenout,
DWORD
skip);
[DllImport("coredll.dll",
SetLastError=true)]
public static
extern
unsafe int
BinaryCompress(byte*
bufferIn, int
lenIn,
byte*
bufferOut,
ref int
lenOut);
[DllImport("coredll.dll",
SetLastError
= true)]
public static
extern
unsafe int
BinaryDecompress(byte*
bufferIn, int
lenIn, byte*
bufferOut,
ref int
lenOut, int
skip);
string
fileOpen =
"\\Temp\\test.bmp";
byte[] bFile;
bFile =
OpenFile(fileOpen);
byte[]
bufferNew
= new
byte[bFile.Length];
int iLen = 0;
int ret = 0;
fixed(byte*
```

```
bLP =
bFile)
fixed (byte*
bLPNew =
bufferNew)
{
ret =
BinaryCompress(bLP,
(int)bFile.Length,
bLPNew,
ref
iLen);
if (ret != 0)
{
ret =
Marshal.GetLastWin32Error();
MessageBox.Show(new
Win32Exception(ret).Message);
//
this always
return 87 /
Win32Exception
}
}

if anyone
wants to see
the
OpenFile
private
static byte[]
OpenFile(string
fileOpen)
{
byte[] bFile;
FileStream
fs =
File.OpenRead(fileOpen);
bFile = new
byte[fs.Length];
byte[] bAux
= new
byte[1024];
int
bytToRead
=
(int)fs.Length;
int count =
0;
int idx = 0;
while
```

```
(bytToRead
> 0)
{
count =
fs.Read(bAux,
0,
bAux.Length);
idx =
(int)fs.Length
–
bytToRead;
for (int i =
0; i < count;
i++)
bFile[idx+i]
= bAux[i];
bytToRead
−= count;
}
fs.Close();
return bFile;
}
```

The file length is 307254 bytes.
Any idea??
Thanks a lot.

.