Re: CFont memory leak !?

Source: http://www.tech−archive.net/Archive/VC/microsoft.public.vc.mfc/2006−12/msg01845.html

- *From*: Joseph M. Newcomer <newcomer@xxxxxxxxxxxx>
- *Date*: Wed, 13 Dec 2006 22:24:18 −0500

I built an app that implements this code. It runs at 3.3MB and never changes size. So your leak is somewhere else.

By the way, instead of doing the Invalidate explicitly, anything that changes the behavior should call Invalidate. So ExStatic::SetNewFont should include at the end

if(::IsWindow(m_hWnd))
Invalidate();

This allows you to call the SetNewFont on the control before it is created, but guarantees that if you change the font, the redraw is forced. It should not be the responsibility of the caller to force an Invalidate
joe

On 13 Dec 2006 17:33:12 −0800, "Peter Lee" <gk64higo@xxxxxxxxx> wrote:

> sorry, I means 10+ MB, not 1MB...
> if I don't change nPointSize, just change LOGFONT lfItalic,
> lfUnderline, the memory will NOT increase...
> but if the nPointSize is changed in a big wide range continuously,
> memory increase until 10+MB (memory upperbound size depend on "int
> nPointSize = 1 + rand() % 60;" range)
>
> I also found some CFont related class at www.codeproject.com, the
> problem is still exist...
>
> I don't know if this behave is correct !? windows cache the font ?
>
> Joseph M. Newcomer ?g?D?G
>
>> (a) don't believe task manager
>> (b) 1MB is trivial, ignore it
>>
>> On 13 Dec 2006 01:09:40 −0800, "Peter Lee" <gk64higo@xxxxxxxxx>
>> wrote:

I look taskmgr.exe , memory takes about 1X MB, is there
something wrong
with my code?

```
OnInitDialog()
{
SetTimer(1, 20, NULL);
}

void CMy0dlgDlg::OnTimer(UINT nIDEvent)
{

int nPointSize = 1 + rand() % 60;
m_StaticValue.SetNewFont( nPointSize, _T("Arial"));

m_StaticValue.Invalidate();
}


bool ExStatic::SetNewFont(int nPointSize, LPCTSTR
lpszFaceName, CDC*
pDC /* = NULL*/ )
{

bool bRet = false;
CFont NewFont;

LOGFONT lf;

bRet = (0 != m_font.GetLogFont(&lf));
if (!bRet)
{
return false;
}

lstrcpyn(lf.lfFaceName, lpszFaceName,
sizeof(lf.lfFaceName)/sizeof(lf.lfFaceName[0]));
lf.lfHeight = nPointSize*10;

bRet = NewFont.CreatePointFontIndirect(&lf, pDC);
if (!bRet)
{
return false;
}

HFONT hNewHandle = (HFONT)NewFont.Detach();

m_font.DeleteObject();
m_font.Attach(hNewHandle);
return true;
```

}

****

Nothing jumps out here. I think you are being misled by what task manager is telling you.
Read my essay "How large is my program?" on my MVP Tips site. Frankly, I see absolutely
nothing worth discussing for a program as trivially small as 1MB. After you read the
article you will realize that program size as report by task manager has little to do with
how big your program is. In fact, only the first derivative matters, not the absolute
value. If the program keeps growing according to task manager, you quite possibly have a
leak. If it just 1MB, you have a rather small program, so don't worry about it.
joe
Joseph M. Newcomer [MVP]
email: newcomer@xxxxxxxxxxxx
Web: http://www.flounder.com
MVP Tips: http://www.flounder.com/mvp_tips.htm

Joseph M. Newcomer [MVP]
email: newcomer@xxxxxxxxxxxx
Web: http://www.flounder.com
MVP Tips: http://www.flounder.com/mvp_tips.htm
.