**Not quite what you are looking for? You may want to try:**

- Printing and Print Preview OpenGL with MFC
- An MFC Chart Control with Enhanced User Interface

**highlights off**

12,650,187 members (28,078 online)

CODE PROJECT®
For those who code

STOP COUNTING USERS

Syncfusion®

| home | articles | quick answers | discussions | features | community | help | | |
|------|----------|---------------|-------------|----------|-----------|------|--|--|

MFC DPI 🔍

# Fight the dialog units, **DPI** and Large Fonts

**George Yohng**, 17 Nov 2003     `Zlib`

Rate:

★★★★★     4.95 (24 votes)

Guarantees pixel-to-pixel matching appearance of resource-based dialogs for different font **DPI**s

**Download demo project (VC++7.1) - 54 Kb**

**Download source - 5 Kb**
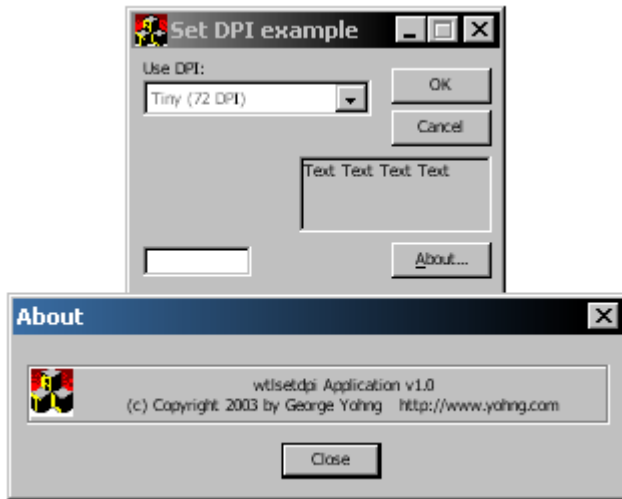
# Introduction

Resource-based dialogs, made with a dialog editor do have a potential problem of being scaled improperly for larger Windows font resolutions, no matter whether **MFC**, ATL, WTL or bare Win32 is used.

The given class allows to pin down and fix a particular **DPI** mode for a particular dialog resource. Furthermore, **DPI** resolution/scaling of a resource-based dialog can be dynamically changed in run-time.

# Background

For example, static bitmaps on the dialogs normally do not get resized, thus if one would want to make some fancy bitmap background or matched illustration, one will obviously run into problems with "Large Fonts" Windows mode. This is especially useful for wizard-style and login dialogs.

Some of the users tend to have "Large size (120 **DPI**)" font mode set, which is problematic for the developers, as triple checks are to be made to find out whether dialogs designed look properly in 120 **DPI** mode.

If the program interface is mostly bitmap based, the best way is to lock the resolution down to 96 **DPI** and disallow any further dialog scaling. Well, unfortunately, Windows does not seem to have an easy way of turning off **DPI**-dependent dialog scaling and "dialog units". I've been looking through the network and so far found no easy solutions for the given problem.

Therefore, I have written a class. Once `Attach` method is invoked in the `WM_INITDIALOG` handler before the dialog is shown, the dialog will be resized and adjusted in run-time to match the specified resolution.

The supplied code contains a subroutine to re-parse of the dialog resource and to re-calculate **DPI**-related values such as control positions and sizes. Resolution in **DPI** is specified as the parameter to the `Attach` method, and the standard Windows resolution is 96 **DPI**.

# Using the code

The code is tested and functional within **MFC**, ATL/WTL and Win32 frameworks.

**MFC** Example:

```
...

BOOL CMyDlg::OnInitDialog()
{
    CDialog::OnInitDialog();

    dpi.Attach(AfxFindResourceHandle(IMAKEINTRESOURCE(IDD), RT_DIALOG),
            m_hWnd,IDD,96.0); // 96 is the DPI

    // The rest of your initialization code goes here

    return TRUE;
}
```

ATL/WTL Example:

```
...

BOOL CMyDlg::OnInitDialog()
{
    CDialog::OnInitDialog();

    dpi.Attach(_AtlBaseModule.GetResourceInstance(),m_hWnd,IDD,96.0);
    //                                              ^^^^ DPI

    // The rest of your initialization code goes here

    return TRUE;
}
```

# Caveats

I have looked for an easier way, but so far I have found none. The parser will only work with **DIALOGEX** structures and will not work with obsolete **DIALOG** structures.

Also you have to explicitly specify the dialog font. For proper sizing, you need to use `Microsoft Sans Serif` or `Tahoma` (and **NOT** `MS Sans Serif` or `MS Shell Dlg`). `Tahoma` has exactly the same metrics as `Microsoft Sans Serif`. You can use any other TrueType/OpenType font, avoid using bitmap fonts, as they will not scale well.

Due to obvious reasons, the size of a checkbox square is not affected, though it still will get proper placement and align.

## License

This article, along with any associated source code and files, is licensed under The zlib/libpng License

## Share

EMAIL          TWITTER

## About the Author

**George Yohng**

Web Developer

Austria 🇦🇹
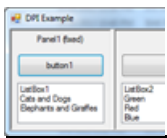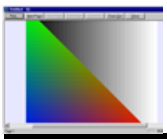
No Biography provided

## You may also be interested in...

Printing and Print Preview OpenGL with MFC

10 Ways to Boost COBOL Application Development

Proper Resizing of SplitterContainer Controls at any DPI


Using the Intel® Edison Module to Control Robots


Mobile Messaging with Twilio


SAPrefs - Netscape-like Preferences Dialog

# Comments and Discussions

Add a Comment or Question

Search Comments               Go

Spacing | Relaxed ▼ | Layout | Open All ▼ | Per page | 25 ▼ | Update

First  Prev  Next

## My vote of 5                                omindream        7-Feb-13 17:05

works seamlessly.... thanks a lot....

Reply · Email · View Thread · Permalink · Bookmark

## Why don't effect on multiline edit and checkbox?        nhchmg        22-Jul-11 18:10

If edit have scroll bar,and dialog have checkbox,then edit and checkbox will show large.

Reply · Email · View Thread · Permalink · Bookmark

2.00/5 (1 vote)

## Help for use in Visual C

symons65    28-Jun-10 1:10

how do I use it in C?
thanks

Reply · Email · View Thread · Permalink · Bookmark

## How can use it in SDI or MDI type application?

"_$h@nky_"    27-May-09 18:53

Hi ,

this is a fantastic job,its really helps alot.

please tell me how can i use it in SDI and MDI type application.

thanks in advance.

To accomplish great things, we must not only act, but also dream;
not only plan, but also believe.

Reply · View Thread · Permalink · Bookmark

## scaling factor for dialog resource [modified]

athish1    2-Apr-07 21:23

ur code snippet for DIALOGEX resource is useful

can u suggest me incase of DIALOG resource??

:(
-- modified at 3:07 Tuesday 3rd April, 2007

Reply · Email · View Thread · Permalink · Bookmark    2.00/5 (2 votes)

## PropertySheet Tab title size is still too big

Luckymmkay    2-Dec-06 1:02

What a huge help this class has been. Recently the software I had been working on was released over seas in Japan and Korea. Both Windows XP versions experience the text sizes inside
our property pages enlarged and pushed out side of the dialogs visible area. This has all been fixed now with your class.

But still the property pages tab titles are still enlarged. I'm not sure how I can fix this. Do you have any suggestions?

Reply · Email · View Thread · Permalink · Bookmark

## Re: PropertySheet Tab title size is still too big

George Yohng    2-Dec-06 1:37

Thanks for using the class 🙂

I guess, this will have to do with the parent form, that includes the tabs. I think, the most simple solution for this would be to force a font for the parent form, but if the parent form is a different application (like if developing ActiveX control), it will have to stay large.

http://www.yohng.com/

Reply · Email · View Thread · Permalink · Bookmark

## Re: PropertySheet Tab title size is still too big

Luckymmkay    2-Dec-06 3:40

I tried setting the parent of the property sheets font size to Arial and Microsoft Sans Serif of size 10 with no luck. The text size will still be enlarged in the Tab titles still. It of course displays correctly on English Windows XP, but will be enlarged on a Korean Windows XP system. What a pain!

I'm rather new to this font size issue, and I will need to bring myself up to speed. Thanks very much for your help.

Reply · Email · View Thread · Permalink · Bookmark       5.00/5 (1 vote)

## an issue when using it in my dialogbar

aisonbert    5-Apr-06 11:49

Hi All,

Thanks for you wonderfull class, it works well in common dialog resources.
But when I used it in dialogbar, I encountered a problem.
I used a CDialogBar object in my Frame/View program.Of course the dialogbar uses a dialog resource, but it displays different size in different DPI mode.Then I found your class, I used it in my dialogbar. The controls, including buttons and static images in my dialogbar keep the same sizes however the DPI changes, but the dialogbar's size still changes.
Does anyone know why? Any tips?

Bert

### Re: an issue when using it in my dialogbar

aisonbert     7-Apr-06 11:01

I got it!
I need to send a message to the dialogbar, tell it the actual size it should have, then overwrite its CalcDynamicLayout() method and CalcFixedLayout(), let them return the actual dialogbar size, then the CFrameWnd object will recalculate the size of the dialogbar.
Here actual size is the size which was display at a fixed DPI, for example 96.0

    2.67/5 (5 votes)

### the issue using MSLU in windows 98/me

ssliao     22-Feb-06 15:50

after I added the MSLU in my project, compiling with unicode option, then running under windows98, in an button click handle, I call the mmioOpen function, but it return NULL always. when I compile my project whithout unicode option, then it run good.

Who can tell me why? Thanks!

Believe myself, I can do better!

### Re: the issue using MSLU in windows 98/me

ssliao     24-Feb-06 17:06

the issue was solved. because some unicode version function can't be called in windows98. for example mmioOpen, it must be replaced with mmioOpenA.

Believe myself, I can do better!

### Oh no!

Alexander Gräf     26-Sep-05 19:46

Do you know why Microsoft decided to include this feature? Most people using a different DPI setting or large fonts have a reason they do so. Now you come along and pin down the size of the dialog instead of designing clean dialogs which can accomodate the different size settings of the users... I dont know what to say, it's simply stupid.

### Re: Oh no!

George Yohng        26-Sep-05 19:55

Now, if your company's project is in beta stage, five days left until release and this problem is discovered - your suggestion isn't helpful at all, and mine provides a solution.

Apparently, the case that I described is quite common. It's however better to have fixed DPI resolution than a screwed one. And this is a quick fix.

http://www.yohng.com/

Reply · Email · View Thread · Permalink · Bookmark

### Re: Oh no!

Nic Wilson        4-Apr-06 13:17

What is stupid, is users like you criticizing and not providing a better solution. What you ask is not simple and can be very messy, having to resize control, dialogs, set different fonts, resize images, etc. and then watch the glitches as the screen refreshes it.

This solution is simple, fast, and in the majority of cases fixes the issue with little pain. The dialog can be designed with both type of users (96 & 120) in mind so the impact on them is reduced.

If you dont like this then provide us with a better mechanism, instead of just saying its stupid. There are some examples on the internet all of which so far have some issues which need to be addresses to be a complete solution. Meanwhile this solution is great imho.

cheers
nw 🙂

Reply · Email · View Thread · Permalink · Bookmark

### Re: Oh no!

CodeSafe Hawk        12-Mar-09 23:22

hehe, just go the other way around, dont use the class, instead, place an invisible static control on your images, and in oninitdialog, find out the size and location of the static, and realign/scale your images to this size.

btw: i faced and solved this problem back at 1999 when i developed a skinned application (and then assigning region to that window to make it non-rectangular (win9x doest support layered windows) ).
our support team stated, that people might have selected to use large fonts for a reason (and they are absolutely right), and it is not for you to judge wether their reasons are justified for your application or not (consider people with sight disabilities).
since i cared more about the visual aspect of the application (ie, a scaled image was out of the question) i decided to reject this support request.

Reply · Email · View Thread · Permalink · Bookmark

## Excellent job

Probably people don't realise how valueable this can be. Considering that a lot of very expensive applications I use do not scale properly on different DPI displays this seems to be the case. However it saved me a lot of time that I would have spent coding the scaling of my dialog bitmaps. It's simple to use and it works. Excellent job.

You got my 5 points...

Regards,
Ingo

Reply · Email · View Thread · Permalink · Bookmark

## MFC and IMAKEINTRESOURCE

Hello,

If IMAKEINTRESOURCE symbol gets you an error (MFC example), then use simple MAKEINTRESOURCE (without leading I).

Thanks,
George.


http://www.yohng.com/


Reply · Email · View Thread · Permalink · Bookmark

### Re: MFC and IMAKEINTRESOURCE

Hi
This is not the problem as the compilation would fail.
In the case I have I get a compilation and an executable. The problem is that in one of
the calls performed within dpi.Attach() there is a failure.

jac


Reply · Email · View Thread · Permalink · Bookmark

2.00/5 (1 vote)

## Re: MFC and IMAKEINTRESOURCE

George Yohng     15-May-05 22:41

This message is not directed to you, but to the users that are unable to compile. This is an easy fix.

Check resolution for your MFC problem in the next thread called 'MFC example'.

Reply · Email · View Thread · Permalink · Bookmark

## MFC example

**Anonymous**     **1-Feb-05 19:53**

Hi, your solution seems great, but it does'nt work for me in a simple MFC application with a simple modal dialog.
Can you add a MFC demo exmple with source so I could debug and compare.

Thanks a lot
sagi

Reply · View Thread · Permalink · Bookmark | Edit · Delete

## Re: MFC example

George Yohng     13-Feb-05 0:21

Hello,

I don't the environment at the moment, but I have checked it with MFC... Check that the following is taken in account:

1. OnInitDialog is properly mapped and called
2. if OnInitDialog is not called for some reason, ensure that dpi.Attach(...) is called from some other place.
3. The dialog should have Tahoma or any other scalable font set. See the article text for more information.

Thanks,
George.

http://www.yohng.com/

Reply · Email · View Thread · Permalink · Bookmark

## Re: MFC example

Hi I tried your suggestion.
But the code fails within the function dpi.Attach()
any chance yuou could check that ?
thank
jac

Reply · Email · View Thread · Permalink · Bookmark

## Re: MFC example

I have checked it, found few things that are not in compliance with my article.

1. First of all, you should have set explicit font for the dialog. Close your development studio and open test_setdpi.rc in notepad. Run notepad and open test_setdpi.rc

2. Press Ctrl-F (for find) and type in "DIALOG DISCARDABLE"

3. Here are the parts that match: (two of them in your case)

```
------------
IDD_ABOUTBOX DIALOG DISCARDABLE 0, 0, 235, 55
STYLE DS_MODALFRAME | WS_POPUP | WS_CAPTION | WS_SYSMENU
CAPTION "About test_setdpi"
FONT 8, "MS Sans Serif"
------------
```

And another part:
```
------------
IDD_DIALOG_TEST DIALOG DISCARDABLE 0, 0, 352, 156
STYLE DS_MODALFRAME | WS_POPUP | WS_CAPTION | WS_SYSMENU
CAPTION "Dialog"
FONT 8, "Microsoft Sans Serif"
------------
```

About box might be of little importance, but let's change it also.

4. Now we are doing hand editing to this. As I said in the article, DIALOG resource entries do not work. Only DIALOGEX do. So you change the lines to appear like:

```
---------------
IDD_ABOUTBOX DIALOGEX DISCARDABLE 0, 0, 235, 55
```

STYLE DS_SETFONT | DS_MODALFRAME | WS_POPUP | WS_CAPTION | WS_SYSMENU
CAPTION "About test_setdpi"
FONT 8, "Microsoft Sans Serif"
---------------
IDD_ABOUTBOX DIALOGEX DISCARDABLE 0, 0, 235, 55
STYLE DS_SETFONT | DS_MODALFRAME | WS_POPUP | WS_CAPTION | WS_SYSMENU
CAPTION "About test_setdpi"
FONT 8, "Microsoft Sans Serif"
---------------

Pay attention - I have changed DIALOG to DIALOGEX, added DS_SETFONT as additional flag right after STYLE, and changed font name to Microsoft Sans Serif. (for my own projects I normally use Tahoma font)

5. Save the file from Notepad

6. Recompile all, run. If something doesn't work - go back to this file and check that all options are intact.

Thanks,
George.

http://www.yohng.com/


Reply · Email · View Thread · Permalink · Bookmark

## Use it in Windows98


hansworscht          16-Jun-04 23:23

Hey folks,

the reason why the code doesn't work in Windows 98 is because it uses unicode versions of functions and datatypes (CreateFontW, wcslen e.g.) which are not available in Win 98 since it does not support unicode.

I got the code running in Win98 using the Microsoft Layer for Unicode(MSLU). That layer translates unicode api calls when the appliction is run under Win95/98/Me. For detailed information on MSLU and how to get and implement it, see:
http://msdn.microsoft.com/msdnmag/issues/01/10/MSLU/default.aspx

Attention: The unicows.lib file that comes with the latest Platform SDK was built with MS VC++ 7 and is not fully compatible with VC++ 6.

Workaround: Don't use the /DEGUG flag in VC++6, use VC++7 of course or use a alternative unicows.lib from http://libunicows.sourceforge.net. This .lib file can be used with a bunch of other compilers too and, what's nice also, you don't need to download the huge Platform SDK from Microsoft in order to get the lib file.

Hope that helps.

Regards,

Arno

1.00/5 (1 vote)

General     News     Suggestion     Question     Bug     Answer     Joke     Praise     Rant     Admin