

# Forcing IE into a Fit

## How to force Internet Explorer in a specific width (and height)

by Bob Swart

In this short tip, I will share with you a problem (or rather a time consuming issue) that I've been faced with as a webmaster for the past year. I will also present the final solution that I've build for myself (not surprisingly - I trust - using Borland's Delphi), which turned out to be a WYNIWYG (What You Need Is What You Get), but not after some tweaking here and there...

### Website Resolutions

As a webmaster (for The Developers' Group website, among others), I feel it's my duty to ensure that the website looks good in different resolutions. From 1024 by 768 to 800 by 600 pixels as most important benchmarks. Unfortunately, with the release of C#Builder and Delphi for .NET, I've felt "forced" somewhat to purchase and use monitors with a higher screen resolution<sup>1</sup>. In fact, I'm now running everything in 1280x1024 at a minimum. Which is fine if you have the monitor to support it, but statistics report most people on the internet still use 1024x768 as screen resolution, followed by both 800x600 (decreasing) and 1280x1024 (increasing in number). Which poses me with a little problem: when running in 1280x1024 resolution, how do I get the Internet Explorer window to resize to 1024x768 or 800x600 in order to get a preview of what the site would look like in that specific resolution?

### The Desktop Solutions

The first solution I used actually dates back about a decade, when I first started to work on my own website (even then it was called Dr.Bob's Delphi Clinic, although the first URL was <http://home.pi.net/~drbob> which isn't around anymore today, but that's another story<sup>2</sup>). The trick was to use a desktop background image, shown at 1024x768 resolution, which was all white, but had a rectangle painted on it of 800x600 and another one of 640x480. The initial version had the rectangles centred on screen, but that meant you had to position the window at the upper-left corner of an inner rectangle, and then resize it. When I refined this by starting each rectangle in the upper-left corner of the screen, it became a lot easier to do this, since it's easier to drag a window to the upper-left corner of your screen and then resize it.

---

#### (Footnotes)

- 1 The fact that both C#Builder and Delphi 8 for .NET use an embedded form designer, reduces the design area of my screen (with the space taken by the Object Inspector, Project Manager and Tool Palette). I have a really hard time to design an 800x600 form using a 1024x768 resolution, not to mention a bigger form. As a result, a width of 1280 is the minimum width at which I consider using C#Builder or Delphi 8 for .NET. 1440 or 1600 is even better.
- 2 Dr.Bob's Delphi Clinic was one of the first homepages devoted to Delphi, and was mentioned in several places. As a result, I received a lot of visitors, which translates in traffic. My Dutch ISP at the time noticed the traffic, and sent me an e-mail early one month to say that I had exceeded the bandwidth limit of 300 MB for that month (probably in the first week). According to their e-mail, I should know that "it's a well known fact that erotic oriented pages get a lot of traffic", and as a result my homepage at <http://home.pi.net/~drbob> would be unreachable until the beginning of the next month. In the meantime, I could still use FTP to access the site and make the "necessary changes". Planet Internet not only "forgot" to verify that my website had all but erotic content (although one could admit Delphi 1 was very sexy), they were also not interested in my repeated requests to actually take a look at the homepage. As a result the content moved and can now be found at the <http://www.drbob42.com> domain, hosted by First-Web courtesy of the Developer's Group. Bandwidth requirements are still high, but I get no more complaints about "erotic oriented pages" - I'm even wearing a hat, what more can I do?

However, with the advancement of digital cameras (and the growing number of pictures I've made of my kids), I started to replace the rectangle background more often with nice pictures as background. And it just didn't feel good to draw rectangles onto these pictures. I could have used little markers (just a dot at the place of the lower-right corner that would make a 800x600 or 640x480 resolution), but that would mean a little bit of work for every new picture that I wanted to use as background. No, I needed an easier solution.

## The MoveWindow Solution

Wouldn't it be possible to enforce Internet Explorer into a specific width programmatically? The IE Com Object has no such methods, but more generically, if you have the Window handle, you can call the MoveWindow function which will not only move a window, but also help you to resize it. MoveWindow has six arguments:

```
function MoveWindow(hWnd: HWND; X, Y, nWidth, nHeight: Integer; bRepaint: BOOL): BOOL;
    stdcall;
```

Apart from the hWnd Window handle, we can pass the new X and Y co-ordinates (of the upper-left corner) as well as the new Width and Height values. The last argument specifies if the contents of the window needs to be redrawn (never hurts, if you ask me).

MoveWindow is exactly what I needed. That was the good news. The bad news is a new problem that I then faced: how do you get the handle of the Internet Explorer window?

## WinSight to the rescue

In order to get a Window handle to an application, we can call the FindWindow function. This one takes two arguments:

```
function FindWindow(lpClassName, lpWindowName: PChar): HWND; stdcall;
```

I often use FindWindow to locate Delphi or Delphi applications, in which case the ClassName can be set to TApplication (for applications written in Delphi) or TAppBuilder in order to locate Delphi itself, as demonstrated with the code snippet below:

```
{$APPTYPE CONSOLE}
uses
  Windows;
var
  Wnd: HWND;
begin
  Wnd := FindWindow('TApplication', 'Delphi 7');
  if Wnd <> 0 then writeln('Delphi 7.x running');
  Wnd := FindWindow('TAppBuilder', nil);
  if Wnd <> 0 then writeln('TAppBuilder');
  readln
end.
```

The question is what to use as classname for Internet Explorer. In order to find out, I used good old WinSight, which can be found as ws32.exe in the Delphi7\bin directory. It appears that the Internet Explorer window that displays the webpages has a classname of IEFramе.

## The Delphi Solution

FindWindow returns only one window handle, if more instances of Internet Explorer are running, you will only get the top-most window handle back (but I can live with that, since it's usually the active Internet Explorer "IEFrame" window that I want to force to a specific size anyway).

So, based on this information, we can write a short test application to resize the top-most Internet Explorer window to 800x600, as follows:

```
begin
  Wnd := FindWindow('IEFrame', nil); // find the window
  if Wnd <> 0 then
    MoveWindow(Wnd, 0, 0, 800, 600, True); // then resize
end;
```

At first this seemed to work OK, but then I performed some more tests and noticed that the window isn't brought to the surface as the topmost window. So, in order to enforce that, I made sure to add a call to the BringWindowToTop function (passing the Wnd handle to it).

This worked fine, except for cases where the Internet Explorer window was minimised. In those situations, the window is not shown. Not even with the additional call to `BringWindowToTop`. I actually had to insert a call to `ShowWindow`, passing `Wnd` as first argument and `SW_RESTORE` as second argument.

This was also needed, since it appears that calling `MoveWindow` on a minimised window has no effect on the size (width and height) of this window. By this I mean that if you use `MoveWindow` to “resize” the window, and then call `ShowWindow` with `SW_RESTORE`, the window will restore to the original size it had (before I called `MoveWindow`). In order to compensate for that, I had to call `ShowWindow` with `SW_RESTORE` first, and follow-up with a `MoveWindow` to set the width and height to their right values. This causes a bit of a flicker when resizing a minimised window (which is first shown/restored in its original size, and then resizes), but at least it works, leaving the new window on top as well.

## The Final Solution

As final touch I’ve decided to place a `TRadioGroup` component on a small form, with pre-defined options for 640x480, 800x600, 1024x768 and 1280x1024 (for future use, when most people use that resolution - by which time I hopefully have access to 1600 or higher ;-)

The source code for the final application is now as follows:

```
unit ClientForm;
interface
uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, StdCtrls, ExtCtrls;

type
  TForm1 = class(TForm)
    RadioGroup1: TRadioGroup;
    Button1: TButton;
    procedure Button1Click(Sender: TObject);
  end;

var
  Form1: TForm1;

implementation
{$R *.dfm}

procedure TForm1.Button1Click(Sender: TObject);
const
  WH: Array[0..3,0..1] of Integer =
    ((640,480), (800,600), (1024,768), (1280,1024));
var
  Wnd: HWND;
  i: Integer;
begin
  Wnd := FindWindow('IEFrame',nil); // find the window
  if Wnd <> 0 then
  begin
    i := RadioGroup1.ItemIndex;
    ShowWindow(Wnd, SW_RESTORE); // first un-minimise
    MoveWindow(Wnd, 0,0, WH[i,0], WH[i,1], True); // then resize
    BringWindowToTop(Wnd) // and finally bring-to-top
  end
end;

end.
```

And the project itself can be downloaded from the downloads section of the DG website (which by the way looks just great in 800x600 as well as 1024x768 resolution).

What started out as a little nuisance, and began as a little project, has ended in some new insight into the workings of the good old Win32 API, and a useful tool that I now use on a daily basis when checking the look (and sometimes feel) of webpages in different resolutions. If you’re a webmaster, it may certainly have its use for you as well. If you have other tips, feel free to share them with us (by writing an article for this magazine, for example).

---

*Bob Swart (aka Dr Bob - [www.drbob42.net](http://www.drbob42.net)) is a software developer, author, trainer, consultant and webmaster for his own one-man company, Bob Swart Training & Consultancy in Helmond, The Netherlands. He writes for numerous computing magazines as well as his own training material, and is also webmaster to the group.*

