home   **articles**   **quick answers**   **discussions**   **features**   **community**   **help**

Search for articles, questions, tips 🔍

Articles » Desktop Development » Dialogs and Windows » Dialogs   Next ➡

Article
Browse Code
Stats
Revisions
Alternatives

Comments & Discussions (165)

Add your own alternative version

EasySize - Dialog resizing in no time!

By **Marc Richarme**, 24 Mar 2007
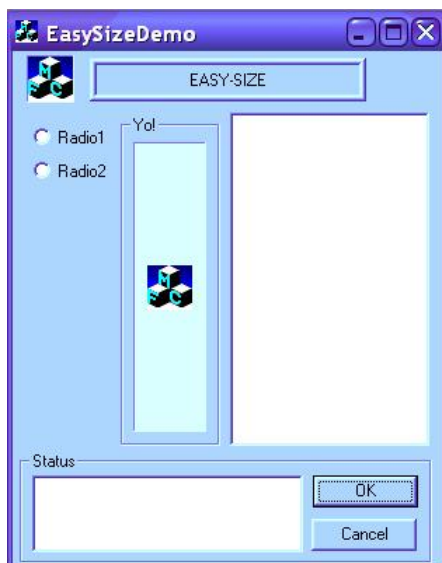
★★★★★ 4.95 (191 votes)   Like   0

Rate:

Tweet 0   g+1 0

Download source files - 1.71 Kb

Download demo project - 13.9 Kb

About Article

An easy way to position controls in resizable dialogs or property pages using just a few macros

Type	**Article**
Licence	
First Posted	**11 Dec 2001**
Views	**433,915**
Downloads	**17,023**
Bookmarked	**254 times**

VC6 VC7 Win2K WinXP
Dev Intermediate

Bookmark Print Email

Top News

Man throws away trove of Bitcoin worth $7.5 million

Get the Insider News free each morning.

Introduction

Related Videos

Have you ever thought of how annoying it actually was to spend a lot of time doing a basic GUI for your simple applications instead of focusing on the actual 'content'? Take for example a resizing dialog or property page. You need to write code for each control that will tell it where to go when the thing is resized, and this can take up a lot of time. Now I know that I'm not the first one to give a solution to this (CResizableDialog), but this article is on my approach.

Description

Basically, all you need to do is design your dialog the way you want it to look in the resource editor (don't forget to make it resizable), and then define how the controls will behave when the dialog is resized using one single macro for each control.

Usage

Note that all this works exactly the same way with both CDialog and CPropertyPage

1. #include *EasySize.h* to your *stdafx.h* (or put it in your include directory and #include *<EasySize.h>* , which I recommend)
2. Add DECLARE_EASYSIZE anywhere in your class declaration:

⊟ Collapse | Copy Code

```
class CEasySizeDemoDlg : public CDialog
{
DECLARE_EASYSIZE
...
```

3. Create an OnInitDialog handler if it doesn't already exist, and put this in the end of it: "INIT_EASYSIZE;" :

⊟ Collapse | Copy Code

```
BOOL CEasySizeDemoDlg::OnInitDialog()
{
    CDialog::OnInitDialog();
...
    INIT_EASYSIZE;
    return TRUE; // return TRUE  unless you set the focus to a control
}
```

4. Create an OnSize handler and add the UPDATE_EASYSIZE; macro to it:

⊟ Collapse | Copy Code

```
void CEasySizeDemoDlg::OnSize(UINT nType, int cx, int cy)
{
    CDialog::OnSize(nType, cx, cy);
    UPDATE_EASYSIZE;
}
```

5. Optional - If you want your dialog to have a minimum size, then create an OnSizing handler and add the EASYSIZE_MINSIZE macro as below:

⊟ Collapse | Copy Code

```
void CEasySizeDemoDlg::OnSizing(UINT fwSide, LPRECT pRect)
{
    CDialog::OnSizing(fwSide, pRect);
    EASYSIZE_MINSIZE(280,250,fwSide,pRect);
}
//(in this example, 280 is the minimum width and 250 the
//minimum height we want our dialog to have)
```

6. Now you have to create the "EasySize Map" (or whatever you want to call it) in which you will specify the behavior of each dialog item. It can be placed anywhere inside your class implementation. The map looks like this:

⊟ Collapse | Copy Code

```
BEGIN_EASYSIZE_MAP(class_name)
    ...
    EASYSIZE(control,left,top,right,bottom,options)
    ...
END_EASYSIZE_MAP
```

The map from the demo application looks like this:

⊟ Collapse | Copy Code

```
...
    //}}AFX_MSG_MAP
END_MESSAGE_MAP()

BEGIN_EASYSIZE_MAP(CEasySizeDemoDlg)
    EASYSIZE(IDC_TITLE,ES_BORDER,ES_BORDER,
        ES_BORDER,ES_KEEPSIZE,ES_HCENTER)
    EASYSIZE(IDC_RADIO1,ES_BORDER,ES_BORDER,
        ES_KEEPSIZE,ES_KEEPSIZE,0)
    EASYSIZE(IDC_RADIO2,ES_BORDER,ES_BORDER,
        ES_KEEPSIZE,ES_KEEPSIZE,0)
    EASYSIZE(IDC_CONTENT,ES_BORDER,ES_BORDER,
        ES_BORDER,ES_BORDER,0)
    EASYSIZE(IDC_STATUSFRAME,ES_BORDER,ES_KEEPSIZE,
        ES_BORDER,ES_BORDER,0)
    EASYSIZE(IDC_STATUS,ES_BORDER,ES_KEEPSIZE,
        ES_BORDER,ES_BORDER,0)
    EASYSIZE(IDOK,ES_KEEPSIZE,ES_KEEPSIZE,
        ES_BORDER,ES_BORDER,0)
```

```
        EASYSIZE(IDCANCEL,ES_KEEPSIZE,ES_KEEPSIZE,
            ES_BORDER,ES_BORDER,0)
        EASYSIZE(IDC_MYICON1,ES_BORDER,IDC_RADIO2,IDC_CONTENT,
            IDC_STATUSFRAME,ES_HCENTER|ES_VCENTER)
        EASYSIZE(IDC_MYICON2,ES_BORDER,ES_BORDER,IDC_TITLE,
            ES_KEEPSIZE,ES_HCENTER)
END_EASYSIZE_MAP

/////////////////////////////////////////////////////
// CEasySizeDemoDlg message handlers
...
```

Looks confusing? It's not once you get the point (and I know I'm not good at explaining it) Read on.

EASYSIZE Macro

The EASYSIZE macro is used in the EasySize Map to specify what behavior your controls will have on dialog resize. It looks like this:

⊟ Collapse | Copy Code

```
EASYSIZE(control,left,top,right,bottom,options)
```

control is the ID of the dialog item you want re-positioned (which will be referred to as the 'current control' further on).

left, top, right and bottom can be either the ID of another control in the dialog (not the current control), or one of the special values, ES_BORDER and ES_KEEPSIZE.

Basically, if you specify an ID, the distance from the current control and the item designated by the ID will remain the same when the dialog is resized: The current control will 'stick' to the other item. ES_BORDER works the same way as if you had specified a control ID, except that it's the distance between the current control and the dialog border that will be kept constant. Specifying ES_KEEPSIZE in, let's say left, will keep the width of the current control the same, and will make the current control right-aligned to whatever you specified in right. The width (or height, if you specified ES_KEEPSIZE in top or bottom) of the current control will always remain what it is in the dialog resource. (I know this explanation sucks, but look at the demo application if you are confused or post you question in the board below). Obviously ES_KEEPSIZE cannot be specified in both "left and right" or "top and bottom".

options can be a combination of ES_HCENTER, ES_VCENTER and 0 (use 0 if you don't want any of the other). ES_HCENTER horizontally centers the control between the two items specified in left and right (both of those can not be ES_KEEPSIZE!). The width of the current control will always remain the same as in the dialog resource. ES_VCENTER works the same way, but for vertical centering (using top and bottom, and where the height will remain constant).

Conclusion

Well I hope you figured out how this works, because it really can make your life easier. Note that using these macros will probably make your compiled code slightly bigger and slower than if you had coded the resizing routines manually, but in most cases the change is so small not even you will notice.

Last Update - Just corrected a few typos.

License

This article has no explicit license attached to it but may contain usage terms in the article text or the download files themselves. If in doubt please contact the author via the discussion board below.

A list of licenses authors might use can be found here

About the Author
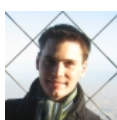
Marc Richarme
Engineer Nokia
Denmark 🇩🇰

My programming experience started a long time ago in QBasic (on a 25MHz 486).
I'm now mainly using Java, C++, C, MFC, Perl and PHP, but have used quite a number of other languages as well for various projects.

Article Top

Comments and Discussions

☐ Profile popups   Spacing [Relaxed ▾]   Noise [Medium ▾]   Layout [Normal ▾]   Per page [25 ▾]   [Update]

First Prev Next

📄 **My vote of 4** 📌	👤 **Miaoz**	**14-Oct-13 12:24**
ⓘ **good** 📌	👤 **liaoyungchun**	**20-Aug-13 15:54**
📄 **My vote of 5** 📌	👤 **kobe_guang**	**23-Jan-13 9:34**
ⓘ **about EASYSIAE micro** 📌	👤 **November 2013**	**16-Jan-13 15:32**
📄 **My vote of 5** 📌	👤 **tr0217**	**27-Nov-12 14:54**
📄 **My vote of 5** 📌	👤 **zhanglipeng**	**23-Nov-12 9:34**
ⓘ **Would need a more version where windows can be added and removed dynamically** 📌	👤 **llothar**	**20-Nov-12 0:08**
📄 **My vote of 5** 📌	👤 **qcth**	**15-Nov-12 15:17**
ⓘ **EASYSIZE controls do not show when dialog open** 📌	👤 **Member 9360202**	**7-Nov-12 3:53**
📄 **My vote of 5** 📌	👤 **ylh071032**	**4-Sep-12 14:44**
ⓘ **Great!!!** 📌	👤 **_oleg_**	**25-Aug-12 3:19**
ⓘ **re:** 📌	👤 **Proc_AJ**	**4-Aug-12 11:17**
ⓘ **Resize the dialog** 📌	👤 **ankitaGupta**	**14-Jul-12 17:54**
📄 **My vote of 5** 📌	👤 **Member 1643253**	**24-Apr-12 19:55**
ⓘ **dialog** 📌	👤 **Kingchj**	**22-Dec-11 10:54**
ⓘ **how to update control sizes on init dialog** 📌	👤 **rayjoslyn**	**21-Dec-11 4:10**
📄 **Question** 📌	👤 **haoyuntianyi**	**1-Dec-10 15:36**
📄 **Need help!** 📌	👤 **Arrin**	**27-Oct-10 18:31**
📄 **Two controls side by side** 📌	👤 **Jonnster**	**13-Oct-10 23:37**
📄 Re: Two controls side by side 📌	👤 walster	26-Jul-11 0:54
ⓘ **can it work at CFormView?** 📌	👤 **batsword**	**9-Sep-10 15:30**
☑ Re: can it work at CFormView? 📌	👤 jj00	22-Sep-10 5:29
📄 Re: can it work at CFormView? 📌	👤 batsword	12-Nov-10 9:11
📄 Re: can it work at CFormView? 📌	👤 magicpapacy	24-Dec-10 18:59
☑ Re: can it work at CFormView? 📌	👤 michaelsoros	13-Jul-13 19:44

Last Visit: 8-Dec-13 9:03   Last Update: 9-Dec-13 11:13   Refresh   **1** 2 3 4 5 6 7 Next »

📄 General 📰 News 💬 Suggestion ⓘ Question 🐞 Bug ☑ Answer 😄 Joke 🅶 Rant ⓘ Admin

Use Ctrl+Left/Right to switch messages, Ctrl+Up/Down to switch threads, Ctrl+Shift+Left/Right to switch pages.