

## Re: wm\_message

---

*Source:* <http://www.tech-archive.net/Archive/VC/microsoft.public.vc.mfc/2007-04/msg00124.html>

---

- *From:* AI <AI@xxxxxxxxxxxxxxxxxxxxxxxxxxxxxx>
  - *Date:* Mon, 2 Apr 2007 03:02:01 -0700
- 

Well I have decided to initialize the pointers from the view without using messaging. But when is it proper to use uwm\_messaging?

—  
Just AI

"Joseph M. Newcomer" wrote:

What I would do is obtain the values I want from the document, set the in the dialog, invoke the dialog with DoModal(), and if it exits with IDOK, I will obtain the values from the dialog object and set the back in the document.

On Sat, 31 Mar 2007 15:00:01 -0700, AI <AI@xxxxxxxxxxxxxxxxxxxxxxxxxxxxxx> wrote:

I have been trying to rewrite some code in a attempt to write vc++ correctly. Anyway I have a SDI application. The class info headers are in the doc object, the dialog headers are in the view object. Hopefully correct so far. Now I want to initiate a dialog object for teams and display the teams already saved in the doc. I have been reading Joseph's essays on "dialog box and control design",but still alittle confused. I am thinking that I should write:

```
CTeams * info =
CLTeams *)GetParent()->SendMessage(UWM_QUERY_Teams);
in the OnInitDialog(), in the dialog.
```

\*\*\*\*\*

You tend to not need to do this sort of query unless you are in a modeless dialog where you need to keep querying the information. In a modal dialog you can just create the information you need. For example, one approach is to put an object in the dialog, and query the value from the document.

Consider that I have a collection of data of the form, in a file called parameters.h,

```
class Parameters {
public:
int thing;
int value;
CString stuff;
```

Re: wm\_message

```
};
```

Then, in the dialog, I might declare

```
class CMyDialog : public CDialog {
public:
Parameters parms;
....
};

class CMyDocument : public CDocument {
public:
void GetParameters(Parameters & parms);
void SetParameters(Parameters & parms);
....
}

void CMyView::OnSetParameters()
{
CMyDialog dlg;
GetDocument()->GetParameters(dlg.parms);
if(dlg.DoModal() != IDOK)
return;
GetDocument()->SeParameters(dlg.parms);
}
```

This is one of many themes and variations you can do.

\*\*\*\*\*

Then register the message and declare a variable. I am not familiar with wm\_ so do I add these to the dialog class and exactly how do I get a pointer from doc to view to dialog? I am not seeing it yet.

\*\*\*\*\*

I try to avoid ever having a pointer to a document in a dialog. I believe it contributes to complexity.

\*\*\*\*

And if I have a CTypedPointerArray of teams in my doc class and I get the pointer of this object to the dialog object can I do deleting and adding from the dialog object? And lastly when I close the dialog object will it affect the array in some way like deleting it? I will continue reading but wouldn't mind a little help also.

\*\*\*\*

I would be disinclined to allow the dialog to directly manipulate anything in the document. If you click "cancel" you have to undo the actions. This can be very complex.

If you want to be able to manipulate the object in both the dialog and the document, my inclination is to not make the information part of the document, but make the information

Re: wm\_message

Re: wm\_message

part of a different class. There is a reference to an instance of that class in the document, and a reference to an instance of that class in the dialog. If you define an assignment operator for the class, then you can make a copy for the dialog, manipulate it in the dialog, and copy it back upon getting IDOK. If you manipulate it directly, you have to remove the Cancel button because you don't have a way to back out of the changes, not usually a good idea.

As far as writing the contents of the document, I would define a serialization method of some sort (memory-to-file and file-to-memory, in whatever way you want) as part of the object. When the object is to be written out by the document, it merely invokes the serialization operation on the object, and the object takes responsibility for writing its contents out (I tend to use XML format, so it is nicely symmetric and objects can embed in other objects).

I would be largely disinclined in this case to use messages to do queries. As I said, I tend to do that only for modeless dialogs.

joe

\*\*\*\*\*

Thanks Al

Joseph M. Newcomer [MVP]

email: newcomer@xxxxxxxxxxxxx

Web: <http://www.flounder.com>

MVP Tips: [http://www.flounder.com/mvp\\_tips.htm](http://www.flounder.com/mvp_tips.htm)