

Microsoft Visual C++/Microsoft Foundation Classes (MFC)

Frequently Asked Questions

[Michael Pickens \(mfc_faq@stingray.com\)](mailto:mfc_faq@stingray.com)

Version 5.6

Last updated: 7/01/98

ATTENTION!!! The MFC FAQ

5.5 is still available to download. It is formatted in fancy HTML Help so you can download it and search it locally on your machine. [Click here to download the file \(mfc_faq.exe 1.34 Mb - self installing with instructions\)](#).

The MFC FAQ is graciously sponsored by [Stingray Software](#). Stingray Software was founded to create and market MFC extensions. It is our belief that MFC/VC++ developers would rather use object-oriented class libraries, instead of binary, black box, OLE controls (a.k.a. ActiveX controls). We are now almost three years old and have several thousand customers who agree that MFC extensions are a great solution to many Windows development problems.

Here is a quick list of our products:

[Objective Grid](#) - The first and by far the most full-featured 100% MFC grid control on the market - click here to check out the hundreds of advanced features that this super-powerful table/spreadsheet control offers.

[Objective Toolkit](#) - Over 50 advanced GUI features for the MFC developer - bring your applications into the 90's and blow away your competition. Includes: Visual Studio style docking windows, incredible toolbar control, tree control,

outlook control, tabbed windows and dozens of other great features.

Objective Toolkit PRO - Advanced frameworks for the serious MFC developer such as: Model-View-Controller (design pattern that enhances document/view), layout manager, scripting engine and more!

Objective Chart - The only 100% MFC charting system! Tons of chart types are supported, and the product is also extremely extensible so you can add your own custom chart types in minutes!

Objective Diagram - Ever have to use GDI and MFC? Objective Diagram helps you create CViews in minutes instead of months!

Objective Edit - An embeddable source code highlighting control - has a complete Visual C++ look and feel.

Visual CASE - The only modeling tool designed from the ground up for Visual C++.

Objective Studio - Combines the Stingray MFC family of products in convenient packages for various developer levels including: standard (for the beginning MFC developer), PRO (advanced - 6 month+ - MFC developer) and Universal (multi-language [VB, VC, VJ] development).

[Click here to go to the Stingray WEB!](#)
Free demos, white papers, etc..

Instant - FAQ

To automatically be notified of new MFC FAQ releases, we invite you to join the Instant-FAQ program. To join, type your e-mail address into the online form below and hit submit. Or, send us an email with the subject 'Subscribe Instant-FAQ' to the MFC FAQ headquarters at mfc_faq@stingray.com and we'll add you for free. We at MFC FAQ headquarters aim to please!! (In other words, it's like a free MFC FAQ subscription, this is NOT a mailing list or listserver.)

<i>enter you email address to sign up for instant -FAQ today!</i>	
<input type="text"/>	<input type="button" value="Submit"/>

Shameless Plug...

Don't be one of the few MFC/VC developers on the planet without a copy of MFC Internals! This is the ONLY book that covers the "other" side of MFC - how it is implemented. Ever hit an ASSERT in MFC? A bug? Ever have to work around some limitation in MFC? MFC Internals is for you! The book covers MFC 4.0, but NOTHING has changed since then in the areas covered, so it is still a very valid reference for those of you that don't want to stop at the written documentation. <http://www.amazon.com> has the best price and we have some details on the book at http://www.stingray.com/mfc_internals or at <http://www.aw.com> .

Buy a copy (or 2 or three) today!!

Looking for a job?

Stingray now has a ton of [job openings](#) and if you are at the top of the MFC/ActiveX food chain - we need you!

Copyright

This document is compilation Copyright (c) 1997 by Scot Wingo. It may be freely copied and/or distributed in its entirety as long as this copyright notice is not removed. It may not be sold for profit or incorporated into commercial products without the author's written permission. [Compilation copyright means that you can freely use individual sections of this document, but any significant collection of sections is subject to the copyright. Hey, I may want to turn this into a book or a movie some day!]

This FAQ is in no way connected with Microsoft. It contains some answers to frequently asked questions about their products. The author in no way guarantees that any of these answers are correct. This is just a collection of information posted to public on-line forums to help the average MFC programmer. [In other words, please don't sue me.]

Release Notes

The FAQ has been completely revamped to be more usable in today's HTML-based world. The MFC/Visual C++ FAQ has tons of answers to questions you may encounter, book reviews, Visual C++/MFC undocumented secrets, an unauthorized history of MFC and much more!

We'll be updating the FAQ bi-monthly and future releases will include:

- Samples that demonstrate the FAQs!
- HTMLHelp edition!
- Complete Visual C++ 6.0 coverage (when it is released) - including new MFC/ATL libraries.
- Many new FAQs.

Many of you may not realize it, but we receive almost a hundred emails a day at MFC FAQ headquarters. Most of the emails are Instant-FAQ™ submissions, but many of them are questions about MFC. Unfortunately, we are not able to answer your questions, this would take far too long and it would only benefit one person - you. Instead, we urge you to post your question to a public forum, namely the MFC news groups or the MFC mailing list and we have even created a MFC FAQ newsgroup that we invite you to post questions to. By sharing your question and answer with the MFC community, many more people benefit from the experience. The MFC FAQ newsgroup and a variety of others are discussed in [section 2.2](#) of the MFC FAQ.

FYI, George Shepherd and Scot are regular columnists for the Dr. Dobbs Journal (The Undocumented Corner Column)- <http://www.ddj.com> and the Microsoft Systems Journal (The Visual Developer Column) - <http://www.msaj.com> and also (phew!) Visual C++ Developer's Journal - <http://www.vcdj.com> so check out the new Shepherd/Wingo material monthly in these different mags. (The Stingray scoop and monthly Scot's Soapbox at <http://www.stingray.com> are also updated pretty frequently.)

Michael Pickens and Scot Wingo - MFC FAQ dudes

mailto:mfc_faq@stingray.com
<http://www.stingray.com>

What's New in the MFC FAQ?

Ok - enough rambling it's time for the moment you have all been waiting for - new MFC FAQ material! This version of MFC FAQ is only being distributed as an HTML file right here at this location. If you are not reading this at http://www.stingray.com/mfc_faq, then it is probably an old version so you should click on that link and go the site to see if it has been updated. If you would like to get a free subscription to the MFC FAQ, which is sent via e-mail as HTMLHelp, then [click here](#).

One of the cool things about the MFC FAQ is that many of the FAQs have been answered by members of the MFC team or someone here at Stingray so you can be pretty sure they are going to be helpful.

Disclaimer: One of the things we do at Stingray for product ideas, especially our Objective Toolkit product, is monitor the newsgroups and implement the most commonly asked for features that are not in MFC. For example, how to implement docking windows such as those found in Visual Studio is commonly asked. So the answer to this FAQ is that Stingray's Objective Toolkit does this, but it is not trivial to do with raw MFC. It is not the intention that the MFC FAQ be a total product plug, but to be honest if that's the answer, that's what is going to end up in here. If this bothers you, press "back" now.

To find out what's new in a current release of the MFC FAQ look for these keywords throughout our site:

NEW!! - An entirely new, fresh FAQ!

UPD!! - An updated FAQ!

Or, type these into our search engine to view only the newest or updated FAQs. [NOTE: The old keys (% and &) have been removed because they were reserved in our search engine.]

If you have ideas for new sections of the MFC FAQ (the ActiveX/OLE section is admittedly not that great right now), please email mfc_faq@stingray.com

Want to learn more about MFC/VC?

There are three resources I am compelled to plug that every MFC/VC++ developer should be aware of. These are covered in detail in section TODO, but I want to bring them to your attention here too.

1. **The MFC mailing list** - this was sponsored by the world famous David Elliott, but he burned out on maintaining and moderating the list and now it is sponsored by Microsoft and moderated by a team of moderators. This is an EXCELLENT way to learn about the MFC FAQ. Details on joining are in [section 2.2.2](#).
2. **Visual C++ Developer's Conference** - This is a conference 100% about Visual C++. If you are like me and are getting burned out on every conference suddenly covering the Internet, but you want to learn more about OLE/MFC/ATL/VC/COM/ActiveX as far as applications/Windows programming is concerned, then VCDC is for you. [Section 2.5](#) has details or surf to <http://www.vcdj.com> .
3. **Visual C++ Developer's Journal** - Both an online and a printed magazine VC++/MFC - <http://www.vcdj.com> - excellent stuff. The online magazine comes out monthly. The printed magazine is published bi-monthly in February, April, June, August, October and December. Both are FREE!!

See also: [section 13.5.5](#).

1. MFC FAQ Info and Credits

The MFC FAQ maintainers can be contacted at mfc_faq@stingray.com

We welcome any comments, criticisms or suggested additions for the FAQ.

- 1.1. [How do I get this FAQ?](#)
 - 1.2. [What's the goal of the FAQ?](#)
 - 1.3. [Why are you doing this?](#)
 - 1.4. [How can I contribute!?!](#)
 - 1.5. [How is this different from the MSVC MFC FAQ?](#)
 - 1.6. [What other languages is the FAQ translated to?](#)
 - 1.6.1 [Japanese version of MFC FAQ](#)
 - 1.7. [Credits](#)
-

1.1. How do I get this FAQ?

Hey, this is one of those chicken and egg questions, if you're reading this you already have the FAQ. But seriously, the MFC FAQ is put out just about monthly, so chances are you are reading an old and stale version, instead of the fresh and fancy latest version. If you want to make sure you have the latest version:

"[Instant-FAQ](#)" - If you'd like to have a copy of any new MFC FAQ releases emailed right to your front door, just [click here...](#) or send us an email via mfc_faq@stingray.com and we will add you to the 'subscription'. No cost or anything, just another FAQ service.

"Why wait for the announcements and then go get the FAQ yourself, use "Instant-FAQ"

"Instant-FAQ changed my life!!! I no longer have to wait for months for the latest FAQ. It's just waiting in my email box the second it's released, no fuss no muss! Thanks Instant-FAQ"

-Anonymous Instant-FAQ participant.

[All restrictions apply in states that do not allow Instant-FAQ service, please see your local MFC FAQ dealer for details.]

This FAQ is only available on the World Wide Web at http://www.stingray.com/mfc_faq - if you are reading it somewhere else (e.g. MSDN) it is probably stale.

Sadly, I can not email the FAQ to you because of my current workload.

If you send me an email asking for the FAQ, I'll just put you on Instant-FAQ and you'll get the next version announcement instead.

An announcement of a new FAQ release is posted to the Usenet newsgroups:

- comp.os.ms-windows.programmer.controls
- comp.os.ms-windows.programmer.graphics

- `comp.os.ms-windows.programmer.tools.mfc`
- `comp.os.ms-windows.programmer.misc`
- `comp.os.ms-windows.programmer.ole`
- `comp.os.ms-windows.programmer.tools`
- `comp.os.ms-windows.programmer.win32`
- `comp.lang.c++`

The Microsoft MFC newsgroups which I can never remember.

(<news://news.microsoft.com/microsoft.public.visualc.mfc> or something like that)

- There's also an announcement on the MFC mailing list. (See Section 2.2.2.)
-

1.2. What's the goal of this FAQ?

The goal of this FAQ is to be a clearing house for answers to MFC questions commonly asked on primarily the Usenet newsgroups and the MFC List. We try not to duplicate the great documentation out there, the tech notes, MSDN, etc.. Hopefully you will look there first and then use this FAQ as your last resort. The FAQ will be much more interesting if you check those resources first, then we won't have to cover 'the basics'.

I'm trying to load the FAQ up with good phone numbers, CODE-CODE-CODE!, URLs and email addresses of places to call. I'll be putting 800 numbers first, and then the US number second. International folks hate it when there are only 800 numbers, so if you want a phone number in the FAQ, please send both.

1.3. Why are you doing this?

I keep seeing the same posts and they're driving me insane!!! But seriously, I started the MFC FAQ when I had gobs of free time and was trying to get really totally submerged in MFC. I figured, what better way to get submerged, than to manage the FAQ.

1.4. How can I contribute!?!

Good question! You can send me an email with your frequently asked question and answer and I'll stick it right in the FAQ. It's that easy, We'll even make sure you get credit for that entry. Please remember to tell me what section of the FAQ you would like your FAQ added to.

Also, if you search for the string '%%TODO' that indicates there is some information I need to fill in and would really like your help with.

1.5. How is this different from the MSVC MFC FAQ?

There is a MFC FAQ that comes with MSVC, this FAQ is different because:

1. It's not from Microsoft (so it has more dirt and less evangelism!)
 2. The information here is gleaned from Usenet and CompuServe, where their information is from technical support. Hopefully the Usenet/CompuServe folks ask better questions (they are definitely cheaper ;->)
-

1.6. What other languages is this FAQ translated to?

1.6.1. Japanese version of MFC FAQ

I am coordinating efforts with Masahiko Funaki to have a Japanese version of the FAQ. He gets all changes so that he can maintain the Japanese version. He can be reached via email at: 'funa@hic.co.jp'.

The MFC FAQ-J is now available:

http://www.st.rim.or.jp/~funa/mfc_faq.html

1.7. Credits

I borrowed the format from Tom Haapanen's Windows Programmers FAQ, so I'm eternally indebted to him for that. Tom also knew the exact dates of all MFC releases practically down to the second.

I'd also like to use this space to thank some of the folks that always try and help out people on the Usenet groups:

Chris Marriott - He has the highest bandwidth of darn good answers I've ever seen.

Mike Blaszcak - A Microsoftie that reads Netnews and gives great answers! (See section 2.4.2 for info about his book!) If anybody knows how to pronounce his name, let me know.. Mike is now the lead of the MFC Team. If you see a guy post and he signs:

.B ekiM - that is Mike. He's a little backwards, but gives good answers. Ask him about his RX7.

And of course thanks to everyone in the Microsoft MFC/MSVC group that made MFC the phenomenon it is today: Scott Randall, Jim McCarthy, Dean McCrory, John Elsbree, etc., etc..

These folks have given me lots of great feedback on the FAQ and get a big round of applause!

- Jim B - jimb@turningpoint.com
- Niels Ull Jacobsen - null@diku.dk
- Vinny Mayfield converted the FAQ over to Word for me.

Individual FAQ contributions are usually near the FAQ.

2. MFC/VC++ Resource Guide

Hopefully this FAQ is where you will look when all else fails. This section documents what else is out there and how to get your paws on it. FYI - This section used to be called "Juicy MFC Information".

2.1. [Microsoft](#)

- 2.1.1. [Visual C++ and Visual C++ subscription](#)
- 2.1.2. [Microsoft Developers Network \(MSDN\)](#)
- 2.1.3. [Knowledge Databases](#)
- 2.1.4. [Microsoft Software Library](#)

2.2. [Internet](#)

UPD!!

2.2.1. [World Wide Web Pages](#)

- 2.2.2. [Mailing lists](#)
- 2.2.3. [Usenet Newsgroups](#)
- 2.2.4. [Other FAQs of interest](#)
- 2.2.5. [FTP sites](#)
- 2.2.6. [The MFC FAQ FTP Archive!](#)

2.3. [CompuServe](#)

2.4. [Printed Matter](#)

- 2.4.1. [Magazines and journals](#)
- 2.4.2. [Books on MFC](#)

UPD!!

2.4.3. [Cool MFC Articles](#)

2.5. [Trade Shows](#)

2.6. [Users Groups](#)

2.7. [Technical Support](#)

NEW!! 2.8. [MFC/VC++ Related Companies](#)

2.1. Microsoft

MFC comes with Microsoft's Visual C++, Watcom's C++, and Symantec's C++ compilers.

2.1.1. Visual C++ and Visual C++ subscription.

Microsoft no longer sells subscriptions to Visual C++. However, you can get updates to it through the MSDN Universal subscription. See Section 3 for the details of what MFC release is the latest and greatest. Only major versions are sold through outlets, so if you need a point release, you need the Universal Subscription that is now done with MSDN.

Prices fluctuate, so contact one of the compiler vendors for information:

- Microsoft(800) 719-5577(206) 936-8661<http://www.microsoft.com>
- Watcom and Symantec carry MFC compatible VC++'s but I think you would be nearly insane to choose one of these over VC++. They tend to be behind VC++ in functionality and never seem to have the latest MFC.

There are also two programmer mail order houses that carry all of the MFC supported compilers including the Visual C++ subscription:

- Programmer's Paradise - My personal favorite.(800) 445-7899/(908) 445-7899
- The Programmer's Shop(800) 421-8006/(617) 740-2510

2.1.2. Microsoft Developers Network (MSDN)

Another great investment is the Microsoft Developers Network (MSDN), which is a quarterly subscription set of CD-ROMs that contains all of the information you can imagine about the Microsoft developer products, including MFC. There are two levels, level one gets you the CDs that have all the developer tool information. Hey - the MFC FAQ is even on there now. The MSDN options change almost monthly so instead of detailing them here, why don't you just surf on over to <http://www.microsoft.com/msdn> and check out what they have to say. The Universal subscription gives you the biggest bang for the buck because it

sends out everything you could ever want in one affordable package.

If you are Internet challenged, you can contact MSDN at:

(800) 759-5474
(303) 684-0914
msdn@microsoft.com

2.1.3. Knowledge Databases

Microsoft support creates these knowledge databases that answer frequently asked questions and sometimes have samples and stuff. You can get the knowledge databases from <ftp.microsoft.com>, CompuServe, and they come on the MSDN (see section 2.1.2 above) CD-ROMs.

There's a tool that lets you quickly search these. Whenever I've tried to find something here, I always end up with MFC 1.0 information that's not very useful. Your mileage may vary.

<http://www.microsoft.com/KB>

Is the URL that I use, change the product to Visual C++ and party on!

2.1.4. Microsoft Software Library

Microsoft maintains a giant library of samples, tools, and demos called the Microsoft Software Library. Here's how to find it on the various electronic media :

- CompuServe - GO MSL
- Microsoft Download Service (MSDL) Dial (206) 936-6735 to connect to MSDL
- Internet (anonymous FTP) <ftp://ftp.microsoft.com>. Change to the SOFTLIB\MSLFILES directory.

John Elsbree, elsbree@msn.com, programmer.misc, 6/24/95

2.2. Internet

Throw on some sunglasses, put the top down and take a cruise on that big ole' information highway. There's tons of great MFC information at your finger tips (sorry billg), if you know where to look.....

UPD!! 2.2.1. World Wide Web Pages

Here are some URLs with MFC, or MFC related information:

- <http://www.microsoft.com> - Microsoft's WWW.
- <http://www.microsoft.com/visualc> - Visual C++ MotherShip for information - check this weekly if you want to be really hip (or a BIG VC++ NERD depending on your perspective <g>).
- <http://www.stingray.com/dev/dev.htm> - For MFC Developer's ONLY!!
- <http://www.webcom.com/~sleslie/resources.html> - Great page by Scott Leslie of VC++/MFC resources on the web.
- <http://www.stingray.com> - MFC extensions company, Stingray Software
- <http://www.visionx.com/mfcpro> - The MFC Professional - a site that has some useful MFC information, not updated very frequently though.
- <http://www.vcdj.com> - Visual C++ Developer's Journal.
- <http://www.r2m.com/windev/MFC.html> - Microsoft Foundation Classes Internet Resources
- <http://www.richardhaleshaw.com/> - Richard Hale Shaw's page
- <http://www.program.com/index.html> - Program.Com
- http://www.kudonet.com/~ixfwin/winprog_faqs/winprog_faqs.html - Internet Wisdom: Windows Programming
- <http://www.aul.fiu.edu/tech/visualc.html> - Visual C++ Programming Page
- <http://walden.mo.net/~mikemac/progdir.shtml> - The Programmer's Link
- <http://www.grfn.org/~ypwong/ComputerProgramming.html> - Computer Programming Resources
- <http://www.webcom.com/~sleslie/resources.html> - MFC/Win32/Visual C++ Resources on the Web

Lots of magazines are on the web, see section 2.4.1 for those URLs

2.2.2. Mailing lists

The MFC mailing list was originally started by David Elliott and now it has been picked up by none other than Microsoft (Big Brother is listening!). But seriously, it is really great that Microsoft decided to keep this source of peer-level support alive. Oh MFC mailing list, why do I love you? Let me count the ways:

1. Very low noise.
2. Sometimes Mike yells at people and it makes me laugh uncontrollably.
3. It's moderated so someone else has to deal with all of the bounces, weirdos, flames, spams, etc..
4. It's an advanced group - there are no: "What is COject!?" kind of questions - or at least they are few and far between.
5. Want to learn MFC? Spend two weeks reading this and see if you can answer every question. Don't stop until you can. You now know MFC better than the top 2% of programmers.

Interested? Instructions on joining are at:

<http://www.microsoft.com/workshop/essentials/mail.asp> (you may have to have IE to view this, don't know).

There is a nice archive at <http://microsoft.ease.lsoft.com/archives/index.html>

If you want to have some fun, ask if you should call `TerminateThread()`!

2.2.3. Usenet Newsgroups

You can get to Usenet groups through the Internet, or through many of the on-line services these days. Here's a quick rundown of newsgroups that have MFC-specific content or similar things of interest:

- `comp.os.ms-windows.programmer.tools.mfc` - A MFC-only newsgroup - and it's a good one. This is relatively new in the Netnews Universe, so you may have to get an administrator to add it at your news site. It's available on MSN, AOL and CIS.
- `comp.os.ms-windows.programmer.tools` - This is where most of the MFC stuff USED to happen, nothing but noise now. You want to check out the `.mfc` group first.
- `comp.os.ms-windows.programmer.misc` - Give it a shot, lots of noise here, but usually good noise.

- `comp.os.ms-windows.programmer.win32` - I like this group, doesn't seem too crowded, people know what's going on, I guess because they're really into win32.
- `comp.os.ms-windows.programmer.ole` - An occasional thread about some of the OLE classes pops up here.
- `comp.lang.c++` - This is suppose to be for C++ language issues, but people are always posting about MSVC and MFC here. This group is really frustrating to read, I recommend staying away. There's about a 100000:1 noise ratio in here.

Microsoft has ditched their Compu\$erve forums (Yeah!) and now they sponsor "peer support" via a news server at <news://msnews.microsoft.com> . Point your news reader there and then navigate to the `microsoft.public.vc.mfc`. There is a whole hierarchy under VC you may want to explore, but this one is the most active and interesting in my opinion. If you are having problems getting onto this, try this URL:

<news://msnews.microsoft.com/microsoft.public.vc.mfc> if that doesn't work, try a different email reader, but please don't email us, we are not in the business of helping people read newsgroups -sorry.

2.2.4. Other FAQs of interest

- **Windows FAQ** - Maintained by Tom Haapanen.

<ftp.metrics.com:~/faq>

Explains lots of great stuff about Windows that you won't find here.

- **Windows Programming FAQ** - Maintained by Tom Haapanen.

<ftp.metrics.com:~/faq>

Awesome Windows Programming FAQ. I think it's been resurrected by a new fellow under Tom's watchful eye. Keep an ear to the newsgroups for the details.

- **C++ FAQ** - Maintained by Marshall Cline.

[ftp to rtfm.mit.edu:/pub/usenet/comp.lang.c++](ftp://rtfm.mit.edu/pub/usenet/comp.lang.c++), also on comp.lang.c++

This is an awesome FAQ, answers every C++ question you'd ever conceive of. It's not specific to MSVC so be careful, some behavior could be different. These guys have a book out on it.

- **WinHelp FAQ** - Maintained by Pete Davis

Posted regularly to comp.os.ms-windows.programmer.winhelp If you have a question about the Windows help system, this is for you.

- **Windows NT FAQ** - Maintained by Adam Hamilton

Posted to comp.os.ms-windows.nt.misc? Pretty good, lots of networking stuff in there.

2.2.5. FTP sites

There's not much MFC code out there. You can check out these sites:

- [ftp.microsoft.com](ftp://microsoft.com) - Has all the knowledge databases and a couple of MFC specific things. Check out the software library too (softlib).
- [ftp.cica.indiana.edu](ftp://cica.indiana.edu) - pub/pc/win3/progrmr
- wuarchive.wustl.edu - Good site, don't know about MFC stuff. pub/win3/progrmmr
- [ftp.uu.net](ftp://uu.net) - /vendors/microsoft has some developer support things that are on both ftp.microsoft.com and CompuServe. Microsoft seems to be moving away from here.
- [oak.oakland.edu](ftp://oak.oakland.edu) - (SimTel Mirror) SimTel/win3/pgmtools

If you don't have Internet access, you can order CD-ROMs with these entire archives from Walnut Creek. Call (800) 786-9907 or (510)947-5996 for more information.

2.2.6. The MFC FAQ FTP Archive!

We've started a MFC FAQ FTP archive. This archive contains MFC samples that illustrate FAQs and previous releases of the MFC FAQ.

ftp://ftp.stingray.com/pub/stingray/mfc_faq

To submit a sample, upload it into the mfc_faq/incoming directory and send an email to mfc_faq@stingray.com explaining the submission.

Currently the MFC FAQ FTP Archive contains:

- multdocs.zip - Shows how to have multiple CDocuments.
- multview.zip - Shows how to have multiple views.
- onetime.zip - How to restrict your application to one instance.
- dlgcbr.zip - How to put controlbars in your dialogs.

%%TODO - Update these samples for 5.0 -someone? anyone? Bueller? Bueller? Bueller? These were lost when the landru.unx.com account disappeared, anyone have them?

2.3. CompuServe

CompuServe, well how can I say this politely- sucks! Don't waste your time.

2.4. Printed Matter

2.4.1. Magazines and journals

Ok, ok, ok, I admit it! I'm a MAGAZINE JUNKIE! There's just no better place to pick up a good little nugget of MFC info than a well written rag. I think I get every magazine on this list and more. Here's a list of all mags I know of with my review. Please feel free to submit any reviews you have of magazines - it's always a subjective kind of thing.

I've tried to list these in order of MFC content. The ones first have the most MFC content with the ones last having the least. Even though they don't directly cover MFC, most of the Windows programming magazines are still pretty valuable. Any of these that I've read, I put comments with. If you'd like to comment, email me and I'll throw it in here. Please don't tell me about more magazines - I'll go broke <g>.

Visual C++ Developer

Pinnacle Publishing (\$149/yr)
(800) 788-1900 voice
(206) 251-1900 voice
(206) 251-5057 fax
visualcdev@pinpub.com
<http://www.pinpub.com/vcd/home.htm>

My favorite, very meaty, lots of MFC coverage. They have had a rocky year, but appear to be back on track now.

Visual C++ Professional

Oakley Publishing (\$149/yr)
(800) 234-0386
(503) 747-0800

Similar to above, but just a smidge less MFC.

Microsoft Systems Journal (MSJ)

Miller Freeman Inc.

(800) 666-1084
(303) 678-0439
<http://www.microsoft.com./msj/>

Visual Programmer column is EXCELLENT!!!!

Dr. Dobbs

Miller Freeman Inc.
(800) 456-1215
(303) 678-0439
<http://www.ddj.com>

Check out the "Undocumented Corner" for the continuing adventures of George and Scot's wild MFC Internals explorations. Ok, it's not that exciting, but it is a living.

Windows/DOS Developers Journal

Miller Freeman
(913) 841-1631

Continuing to add strong MFC material - getting better!

C/C++ Users Journal

Miller Freeman
(913) 841-1631

Not much MFC specific stuff at all, language-y articles.

C++ Report

SIGS Publications
(800) 361-1279
(212) 242-7447

Great if you want to learn about the dynamics of instantiated, real-time templates as applied to generic programming and patterns in modern-day CASE/UNIX based GNU compilers. <yawn> Don't these guys have day jobs?

2.4.2. Books on MFC

There's tons of books on Visual C++, but not many of them go into the details about MFC. Here's a list of books that are over 50% oriented towards MFC (e.g. don't have pages of screenshots of compiler options dialogs and the MSVC IDE)

Microsoft Visual C++ Documentation Set

Microsoft Press

MSPRESS

(416) 293-8464

Volume 2: Programming with MFC and Win32 (\$27)ISBN 1-55615-802-5

Volume 3: MFC Library Reference (\$45)ISBN 1-55615-801-7

These come in the on-line docs, but nice to have printed versions of them. If you are going to do MFC programming, volume 2 is a 'must have'

Inside Visual C++, Fourth edition

Microsoft Press (\$45)

David J. Kruglinski (Ex-Microsoftie)

ISBN: 1-55615-661-8

<http://mspress.microsoft.com/>

ISBN is 1-55615-891-2

Sounds like it would be Visual C++ specific, but this is the 'Petzold of MFC programming. In other words, it's a 'must have' (along with volume 2 of the printed docs. If you can't afford both, stick with volume 2 on-line and get this one)

*****Note, the MFC/VC++ 5.0 version (fourth edition) just hit the streets - it looks pretty good. -see last comment for the scoop on the latest version. Do not buy the third or second edition, it is bird-cage liner at this point. Rumor has it that David passed away recently in a freak hangliding accident. Does anyone know if this is true or not?!***

The best MFC book that I know of is "Inside Visual C++" by David Kruglinski. Despite the name, this is primarily an MFC book.

Chris Marriott, chris@chrism.demon.co.uk, programmer.graphics, 6/15/95

I owned the previous version of Inside VC++ and learned a lot about MFC [ed note - he now has the third NEW version]. This was the book that got me started in MFC programming and I have recommended it to everyone who asked. I recently bought the latest edition and my impressions are as follows:

- It still is THE book for SDK programmers who want to learn MFC.
- Its strong suit is the coverage of the Document/View architecture, printing and print preview and on-line help.
- It is weak in its coverage of Windows controls (CListCtrl, CTreeCtrl, etc, etc)

I did not feel that the new material it contained justified the almost \$50.00 price tag, however. I recommend that anyone who owns an earlier edition of the book spend a little time in the bookstore with this 3rd edition before taking it home.

- *Blaine Anderson* Blaine.Anderson@msfc.nasa.gov, *mfc-l*, 1/14/96

The Revolutionary Guide to MFC Programming

WROX Press (\$45), June 1996

Mike Blaszczyk

ISBN: 1-874416-47-8

<http://www.wrox.com>

NOTE: A second edition of this book is out, it has MFC in the title, the old one has Win32 in the title.

This is my personal favorite MFC book (well next to MFC Internals of course...).

- scot@stingray.com

If you understand C++ and some windows programming this is a great book to learn MFC.

Shelley Lambert, reach@ee.mcgill.ca, *programmer.tools*, 6/18/95

It is the best book on Win95 programming I have found.

Gary Coombs, N40JW@News.Radio.Org, *programmer.win32*, 6/19/95

MFC Internals - AW George Shephard and Scot Wingo

Addison Wesley \$39.95
George Shepherd and Scot Wingo
ISBN: 0201407213
<http://www.aw.com>

This book is best described as the 'other side' of the Mike B book. Where Mike shows you how to use MFC at an advanced pace, we show you how MFC works under the hood (also at an advanced pace). If you've ever gotten lost in the MFC source, you'll know why this is **extremely** helpful information. Plus, it's MFC 4.0 so it's pretty darn current information. Check out the March "Undocumented MFC" column in Dr. Dobbs for a sneak peek at the book.

-scot@stingray.com - co-author.

"Quite simply, this book is a must-have for any serious MFC developer."

""This book is definitely not a rehash of existing documents. It is not a "how-to" book-it is a "how does it work" book."

-Dean McCrory - technical editor, lead of the MFC team.

Writing Windows Applications with MFC

M+T Books (\$40)
Bryan Waters
ISBN: 1-55851-379-5

At first I thought it was too beginner, but the last 5 chapters kick butt. Lots of info on MFC internals, exception handling, and topics not covered elsewhere. Good for someone new to Windows and MFC programming. - not updated to cover 4.0.

Visual C++ How-To: The Definitive MFC Problem Solver

Waite Group Press (\$40) Apr-95, 570 pages
Info: <http://www.dnai.com/waite/>
(800) 788-3123
(510) 658-3453
Scott Stanfield, Ralph Arveson, Alan Light & Mickey Williams
Includes a CD-ROM
ISBN 1-878739-82-4

Note: Not updated for VC++ 4.0.

I can tell you that its a real good book for what its intended to be. Which is, sort of like your FAQ. It comes with a CD and has little tidbits like:

1. How to animate an icon. I adapted this a bit to figure out how to change icons (in an MDI on the fly).
2. How to detect multiple instances of a Windows app and pull the other one to the top of the Z-order. This is a perennial Usenet question and I pulled the code and had it working first time.
3. How to put a bitmap on the main client window of a MDI app. Again, a nasty little thing to figure out on your own, but given good directions, quite easily done from the book.

The CD is also broken out between those that are using VC 1.5 versus 2.0. Like other Waite Group "How-To" programming books, this one is in "Cookbook" or FAQ format. You read the table of contents for the functionality you are looking for, and the associated chapter contains the code to implement it.

Topics include:

Document and View, Status Bars and Toolbars, Controls, Multimedia, Bitmaps and Icons, Dialogs OLE and DDE, System and a FAQ of short tips. Many new classes are given here which all supplement the MFC in a fairly logical manner. Some of the chapters which were of interest to me were:

- 2.1 - Put the current time in the status bar
 - 2.3 - Add a status bar to my Views
 - 2.4 - Display Progress information in the status bar
 - 4.1 - Preview an AVI file using the common file dialog
 - 4.3 - Play AVI files in a CView
 - 4.4 - Play large WAV files
 - 4.6 - Create 3D animation using OpenGL and MFC
 - 5.5 - Smoothly animate bitmaps
 - 6.6 - Write customized DDX/DDV routines
 - 8.3 - Localize MFC applications using resource DLLs
 - 8.5 - Make multiple inheritance work in MFC
- And many tidbits from the FAQ...

I like the cookbook format which allows me to glue in specific functionality on demand. I found this book to be fairly helpful in that regard. I look forward to Volume II

cntrline@winnet.com via email, 7/15/95

Visual C++ 2: Developing Professional Applications in Windows 95 and Windows NT using MFC

Marshall Brain and Lance Lovette
Prentice Hall
ISBN: 0-13-305145-5

Note: Not updated for VC++ 4.0.

This was the third MFC book I picked up, and by far the best one. Like Brain's other books (Heart of NT...) this one sets out to answer some frequently asked questions, as enumerated in the introduction. It's definitely a comprehensive book, divided into five sections covering the basics of Visual C++ and MFC, Controls, Using the App Wizard with MFC, Professional Features, and Advanced MFC classes. It has an interesting appendix that is designed to get C programmers up to speed with the C++ concepts that they will need to use MFC right away. It does cover things like OLE, ODBC and Threads as well as some 'example chapters' covering things like splash screens, and multiple document templates per application. It's pretty cool because it goes through MFC without the Class/App Wizards before introducing them to you. It's very clear, and most everything is done by example, which I really like ;). What it doesn't do is cover any of the new common controls despite its title implying Windows 95 development. This seems to be ok, because Nancy Cluts' book looks like it should handle that part well enough ;) I highly recommend this one, especially if you have previous windows programming experience and are looking to migrate to MFC 3.x for 95/NT.

Topics of interest:

Understanding Message Maps, Debugging and Robustness, Dynamic Data Exchange and Validation, How MFC Works with Windows, Creating a Self Drawing Control, Property Sheets, OLE, MFC Threads, Combining Two

Documents and Views in a Single Application, Subclassing and Windows Procedures, as well as the standard treatment of every control from edit to combo boxes.

jgordon@hyperimage.com

Animation Techniques in Win32

Nigel Thompson
Microsoft Press (Early 1995)
ISBN: 1-55615-669-3

At first glance, I ignored this book on the shelves, however, when I finally picked it up, it used MFC completely throughout. A very good book on doing graphics with MFC.

jdc@biosym.com, John Clark, via email 6/6/95

Teach Yourself Microsoft Foundation Class Library Programming in 21 Days

Robert Shaw
SAMS (\$30)
ISBN: 0-672-30462-7

Note: Not updated for VC++ 4.0.

I find "Teach Yourself Visual C++ in 21 Days" a good reference, since it is set up to cover a small group of features each "day", so you can quickly go to a subject area, and get a from-scratch description of it and how to use it. However, the author is not strong on using the VC IDE, claiming that "you have to write this code yourself the first few times, so you better appreciate what AppStudio and ClassWizard are doing behind the scenes." I'd prefer the approach where we use the tools at hand, and then map out what those magic MESSAGE_MAP macros are really doing.

Paul.McGuire@SEMATECH.Org, email, 7/11/95

Robert Shaws` book is terrific for the beginner learning MSVC and MFC. Before trying out the examples found within, download the updated files via ftp from the address specified in the book. This will save you alot of frustration. The examples are well thought out. This book is better than Steve Holzner`s series of books on OLE or MFC (what isn't?). Once a

beginner has finished with this book he is ready for more serious books such as "Inside Visual C++" by David Kruglinski.

mario@mksinfo.qc.ca

MS Foundation Class Primer: Prog Windows 3 & Windows NT w/MFC

Jim Conger

The Waite Group Press (\$32) Mar-93, 830 pages

ISBN: 1-878739-31X

Note: Not updated for VC++ 4.0.

The best MFC book that I've seen that DOESN'T use VC++ Wizards is Microsoft Foundation Class Primer.

stasic@news.dorsai.org, ED, programmer.misc, 6/24/95

This book is a good intro for people new to MFC, but it's getting pretty dated. This book was written during the time of MS C++ 7.0, and some of the newest functionality just isn't covered (for obvious reasons ;), which can confuse a newbie. At any rate, it's still makes pretty good book for someone looking for the basics.

cntrline@winnet.com, email, 7/15/95

Master Visual C++ 2,2/E w/CD-ROM (covers MFC 3.0 library)

Howard W. Sams & Company (\$ 50) Jan-95, 1183 pages

Gurewich & Gurewich

ISBN: 0-672305-321

Note: Not updated for VC++ 4.0.

I picked up a copy of "Master Visual C++" specifically for its coverage of creating custom VBX's. This is not covered in Kuglinski's "Inside Visual C++" nor in "Teach Yourself Visual-C++ in 21 Days". "Master Visual C++" is just my speed, since I am a dabbler who does not read instructions or tutorials, just jumps in over my head, then starts looking for the routine calls I overlooked. Similar to the Heavy Metal book, this book could have been squeezed down to about 1/3 the size, but here that is not the point. Each example starts with a demo of the finished product, then goes through the steps descriptively, then goes through the steps again with the code. My little VBX was done in a day an a half, and I have a much better feel for the

ins and outs of this process.

Paul.McGuire@SEMATECH.Org, email, 7/11/95

Heavy Metal Visual C++ Programming

Steve Holzner

Brady Publishing

ISBN 1-56884-196-5

Note: Not updated for VC++ 4.0.

Hate to disagree, but I find this book disappointing. The reason is, I felt like being treated as a semi-idiot when reading the book. Don't get fooled by its number of pages, it could have been easily reduced to half of its size, if the author knew how to list the code efficiently. Furthermore, there are too many misprints in the version I have.

Huayong (yang@math.umass.edu), programmer.tools, 6/18/95

The rule of thumb is never buy Holzner's book. His book only leads you the the door. You can't find practically useful examples. Almost every updated vesion is the same.

Runhong (chen@nice3.ne.rpi.edu), programmer.tools, 6/18/95

Programming the Windows 95 User Interface

Microsoft Press (\$34.95)

ISBN 1-55615-884-X

Nancy Cluts

NOTE: Microsoft Developer Network Developer Library contains the complete text of this book. Might want to check it out there first before buying it?

%%TODO - get a review!

Visual C++ Power Toolkit

Ventana Press (\$50), 1995, 785 pages

Info: <http://www.vmedia.com/vc++.html>

(919) 942-0220

Richard Leinecker, Jamie Nye

Includes a CD-ROM

ISBN 1-56604-191-0

Note: Not updated for VC++ 4.0.

YACTB = Yet Another Cookbook Type Book. Although this book tries to be a sourcebook for programmers by giving many new classes and clever code, it just isn't laid out as logically as other books I've seen recently (see: "Visual C++ How-To" listing). This book covers a lot of topics, such as: Customizing the Interface, Animation, Screen Effects, Music & Sound, Advanced Animation, Data Compression, OLE 2.0, Communications and Scanner Input (Using TWAIN). I haven't found this book to be as useful as other books for my purposes yet, but it does include a lot of new Classes which may prove useful for other programmers.

cntrline@winnet.com, email, 7/15/95

I just grabbed this one and at first glance it looked really useful, then I started reading it and BLECH! The author's writing is, well, it's not so great. Also, the classes are sloppy and show no OO design at all. The author's comment is "I'm not hungarian, so why should I name variables like one". Can't say I recommend this one.

scot@stingray.com

The following are rumored to be coming out RSN (Real soon now)

MFC 4.0 Bible - The Waite Group (\$45)

Extending MFC - AW David Shmitt

Programming Windows 95 with MFC, Microsoft Press, Jeff Prorise
%%TODO - get review, I think this is out.

UPD!!2.4.3. Cool MFC Articles

There's probably a million 'good' MFC articles out there, but this section is for the ultra-hip, totally-swinging, must-have, gotta-read MFC articles. If you submit an article for this section, please include all the 'volume' information and a short paragraph about why it's so damn cool! [Note if you have questions about the magazines or MSDN mentioned below, see sections: 2.4.1 and 2.1.2]

- "Fun with MFC: 33 Tips to Help You Get the Most Out of ..." Paul

DiLascia, MSJ, November 1993, Volume 8, Number 11

This is my A#1 MFC article of all time. Even though this article is 2 years old, it has principals and ideas that are extremely valuable to today's MFC programmer. Good news is that it's on MSDN in their MSJ archives! You can also find this gem under books online: Extensions: articles/Technical Articles/Fun with MFC: 33 Tips.

scot@stingray.com

- "Meandering Through the Maze of MFC Message and Command Routing" Paul DiLascia, MSJ, July 1995, Volume 10, Number 7

Best explanation of messages ever, a MFC FAQ "must-read".

- "Rewriting the MFC Scribble Program Using an OOD Approach" Allen Holub, MSJ, August 1995, Volume 10, Number 8.

Not my favorite article, but 'DID' stir up lots of talk on most MFC forums. Any article that generates this much noise has to be good.

- "Self Decorating Pages" David A. Schmitt, Windows Tech Journal, July 1995
- "Dad, Can I Drive the Printer Tonight?" -David A. Schmitt, Windows Tech Journal, August 1995

This series of articles is pretty good, especially if you are into printing.

- "Plunge into MFC's User Interface" Scot Wingo, Windows Tech Journal, August 1994

Stuff by MFC FAQ maintainer:

- C++ Report, SIGS Publications, "A Brief History of MFC" - recaps section 3 of this FAQ and throws in some new information.
- Dr. Dobbs, January 1996, "Extending MFC" - describes an MFC grid extension.
- Windows Tech Journal, November 1994, "[All aboard the MFC Express](#)"
- Windows Tech Journal, August 1994, "[Plunge Into MFC's User Interface](#)"

%%TODO - Come on MFC FAQ readers - let's get some new submissions in

this section, getting boring!

2.5. Trade shows

- Visual C++ Developer's Conference

There have been two US conferences put on by these guys and they keep getting better. These are kind of modeled after VBITs, where you have very focused talks on VC++/MFC/ATL. Check out <http://www.vcdj.com>. We really highly recommend this show since it is slowly becoming the only place for raw, uncensored MFC/VC information!

- Software Development (East and West)

Miller Freeman
(415) 905-2222
<http://www.mfi.com/sdconfs>
tloftus@mfi.com

This show has gone down hill somewhat as far as MFC/VC is concerned.
Internet internet internet - sigh!

2.6. Users Groups

I don't know of any MFC specific user's groups, but there are some really good C++ user's groups that have some MFC content in their programs. The NW C++ User's group has a really nice email newsletter that anybody can subscribe to by emailing cpp@netcom.com. Here's the contact information for the known C++ User's groups

%%TODO - add information on DC VC++ group.

<i>Area</i>	<i>Contact Name</i>	<i>Contact Email</i>
New York, NY	Saul Rosenberg	Rosenberg@decus.org
Redmond, WA	Steve Shimeall	steves@amc.com
San Jose, CA	Gene Paymar	epaymar@hooked.net
Santa Clara, CA	Marian Corcoran	mcorcora@ix.netcom.com
Tucson, AZ	Martin Lapidus	lascaux@primenet.com
North West	Robin Rowe	cpp@netcom.com
Boston, MA	John Barrie	70373.2241@compuserve.com

2.7. Technical Support

Microsoft MSVC Technical Support
(206) 635-7007

NEW!! 2.8. MFC/VC++ Related Companies

- [DevelopMentor](#) - Simply the *BEST* developer training company. Excellent MFC/OLE offerings too.
 - [Microsoft home page](#) - Yo BILL! What's up?!
 - [NU-MEGA](#) - The penguin guys.
 - [Xtras for Visual C++](#) - Company that sells Visual C++ add-ons.
 - [The Development Exchange](#) - The guys that do VBPI and the Microsoft catalogs.
-

3. A BRIEF HISTORY OF MFC

I constantly see postings from folks confused about what version to use where, etc.. So this history kind of explains where MFC has been and where it's going. [This is the history as I remember it. If you have any details that you can fill in, release dates, inside info, etc.. Let me know!]

To understand the MFC releases, you need to realize that there are 16 and 32-bit releases of MFC. 16-bit releases call Win16 and run on Windows 3.1 and 3.11 (Windows for Workgroups)

The 32-bit releases call Win32 and run on Windows NT and Windows 95 (and can run on Windows 3.1 with Win32s, but it still calls Win32)

Skip to [section 3.13](#) for a quick table of releases if you want to skip the long descriptions of each release.

In each section, the new classes added to MFC are highlighted. It's nearly impossible to do cool class diagrams in ASCII, so their hierarchy is just shown by indentation. I've also left CObject out of most of the hierarchies for brevity. You can pretty much assume that CObject is a parent of most of the classes except things like CString, etc.

3.1. [How do I know what version of MFC I'm using?](#)

3.2. [Pre-MFC 1.0](#)

3.3. [MFC 1.0](#)

3.4. [MFC 2.0](#)

3.5. [MFC 2.1](#)

3.6. [MFC 2.5](#)

3.6.1. [MFC 2.51](#)

3.6.2. [MFC 2.52](#)

3.6.3. [MFC 2.52b](#)

3.6.4. [MFC 2.52c](#)

3.7. [MFC 3.0](#)

3.8. [MFC 3.1](#)

3.9. [MFC 3.2](#)

3.10. [MFC 4.0](#)

3.11. [MFC 4.1](#)

3.12. [MFC 4.2](#)

3.12.1. [MFC 4.2b](#)

3.12.2. [MFC 4.21](#)

3.13. [Table of MFC Releases](#)

3.1. How do I know what version of MFC I'm using?

Check the top of the MFC header file `afxver_.h` (lives in `mfc\include\afxver_.h`). Older versions use `afxver.h`. In there you will see a `#define` for `_MFC_VER`.

This is the hex version of the MFC release with this key:

0x0250

_____	Point release (very minor)
_____	Minor number
_____	Major number

So in this example, the MFC release is 2.5.0.

3.2. Pre-MFC 1.0

In the beginning, Microsoft created a group called the AFX group (stands for (A)pplication (F)ramework(X)). This group probably was created to come out with an OWL competitor, since Borland C++ was doing pretty well. They came up with a class library that presented a very high abstraction from the Windows API, kind of like OWL does.

[Note: The group was actually formed before OWL 1.0 according to elsbree@msn.com]

Legend has it that they then went and tried writing applications with it for a couple of months and found that it was just too far removed from the Windows API. They couldn't leverage any knowledge of Windows and most of these guys were die hard Windows API hackers.

Legend also has it that at this point they scrapped the ENTIRE AFX class library, and then worked on what would become MFC 1.0... You can still see remnants of the AFX days, many of the source files have afx prefixes and lots of macros in MFC today still have AFX in them.

The AFX group was actually responsible for two things: the MFC library and the IDE's support for MFC (namely, the resource editor and the wizards). The AFX name was dropped in April 1994, and the group's members simply became part of smaller teams within the Visual C++ group. One of those smaller teams is today's MFC team.

I've also heard rumors that MFC was once called the MS Fulcrum Classes.

3.3. MFC 1.0

This release was simultaneously released with Microsoft C/C++ 7.0 in early 1992. It provided a very thin abstraction over the Windows API. This initial release did not have any of the document view architecture we know and love today, however it did lay the foundation by introducing CObject, MFC persistence via CArchive and many other features still heavily used in MFC today. This was a 16-bit release released in 03/92.

A 32-bit version of MFC 1.0 was released in July 1993, together with Windows NT 3.1 and the final Win32 SDK. The files there are dated 93-07-24. The Win32 SDK did not include Visual C++ or the MFC 2.x.

MFC 1.0 Introduced These Classes:

General:

- CObject
- CWinApp
- CMenu
- CArchive
- CDumpContext
- CRuntimeClass
- CMemoryState
- CFileStatus
- CString
- CTime
- CTimeSpan
- CPoint
- CRect
- CSize

Exceptions:

- CException
- CMemoryException
- CNotSupportedException

- CArchiveException
- CFileException
- CResourceException

File Services:

- CFile
- CStdioFile
- CMemFile

Collections:

- CByteArray
- CWordArray
- CDWordArray
- CPtrArray
- CObArray
- CStringArray
- CPtrList
- CObList
- CStringList
- CMapWordToPtr
- CMapWordToOb
- CMapPtrToWord
- CMapPtrToPtr
- CMapStringToPtr
- CMapStringToOb
- CMapStringToString

Graphics:

- CDC
- CClientDC
- CWindowDC
- CPaintDC
- CMetaFileDC
- CGdiObject
- CPen
- CBrush

- CFont
- CBitmap
- CPalette
- CRgn

Windows Support:

- CWnd
 - CFrameWnd
 - CMDIChildWnd
 - CMDIFrameWnd
 - CDialog
 - CModalDialog
 - CStatic
 - CButton
 - CEdit
 - CListBox
 - CComboBox
 - CScrollBar
-

3.4. MFC 2.0

MFC 2.0 was a 16-bit release that shipped with Visual C++ 1.0. It added the Document/View framework on top of MFC 1.0 and also added OLE 1.0 classes, message maps and common dialog classes. This was a 16-bit release released on 02/93.

MFC 2.0 Introduced These Classes:

General:

- CCreateContext
- CPrintInfo
- CDataExchange
- CCmdUI

Exceptions:

- CFileException
- COleException

Dialogs: [Note CModalDialog was nuked and functionality moved to CDialog, you can still see a #define CModalDialog CDialog in the header files]

- CDialog [not introduced here, but revamped to be a base for common dlgs]
- CFileDialog
- CColorDialog
- CFontDialog
- CPrintDialog
- CFindReplaceDialog

Windows Support: [Just new derivatives of CButton/CEdit and VBX class.]

- CButton
- CBitmapButton
- CEdit

- CEdit
- CEdit
- CVBControl

Document Architecture:

- CCmdTarget
- CWinApp [*Moved in hierarchy, use to be derived from CObject*]
- CDocTemplate
- CSingleDocTemplate
- CMultiDocTemplate
- CDocument
- COleDocument
- COleClientDoc
- COleServerDoc

Views:

- CView
- CScrollView
- CFormView
- CEditView

Control Bars:

- CControlBar
- CToolBar
- CStatusBar
- CDialogBar
- CSplitterWnd

OLE 1.0 Classes:

- COleServer
 - COleTemplateServer
 - CDocItem
 - COleClientItem
 - COleServerItem
-

3.5. MFC 2.1

MFC 2.1 shipped with Visual C++ 1.1 for NT, it was basically a Win32 port of MFC 2.0. It was a 32-bit release that was released on 08/93.

MFC 2.1 Did NOT Introduce Any New Classes.

3.6. MFC 2.5

MFC 2.5 shipped with Visual C++ 1.5. It introduced the OLE 2 and ODBC classes. It was the last 'official' 16-bit release. Released in 12/93

MFC 2.5 introduced these classes:

General: [OLE and DB related]

- CFieldExchange
- COleDataObject
- COleDispatchDriver
- CRectTracker

Exceptions: [OLE and DB exceptions]

- COleException
- COleDispatchException
- CDBException

Files: [New OLE File support]

- CFile
- COleStreamfile

Dialogs: [New OLE Dialogs]

- CDialog
- COleDialog
- COleInsertdialog
- COleChangeIconDialog
- COlePasteSpecialDialog
- COleConvertDialog
- COleBusyDialog
- COleLinksDialog
- COleUpdateDialog

Windows Support: [Some OLE additions]

- CFrameWnd
- COleIPFrameWnd
- CControlBar
- COleResizeBar

Document Architecture: [Significant classes added for OLE here.]

- CCmdTarget
- COleObjectFactory
- COleTemplateServer
- COleDataSource
- COleDropSource
- COleDropTarget
- COleMessageFilter
- CDocument
- COleDocument
- COleLinkingDoc
- COleServerDoc
- CDocItem
- COleClientItem
- COleServerItem

Views: [New view for DB support]

- CView
- CScrollView
- CFormView
- CRecordView

ODBC/Database Classes:

- CDatabase
- CRecordSet
- CLongBinary

3.6.1. MFC 2.51

A point release to 2.5 (16-bit) that was a bug fix release. Shipped with MSVC 2.0 in 9/94 with MFC 3.0 (32-bit).

3.6.2. MFC 2.52

A point release to 2.5 (16-bit) that added some of the MFC 3.0 features such as property sheets, Winsock and MAPI support. Shipped with MSVC 2.1 in 1/95 with MFC 3.1 (32-bit).

NOTE: This is only available via the MSVC Subscription.

Classes added in 2.52:

- CSocket
- CAsyncSocket
- CSocketFile
- CPropertyPage
- CPropertySheet

MAPI: [Note that MAPI support was added to CDocument, no new classes]

- CDocument::OnFileSendMail
- CDocument::OnUpdateFileSendMail
- COleDocument::OnFileSendMail

3.6.3. MFC 2.52b

[Excerpt from the 2.52b release notes]

Visual C++ 1.52b includes the industry-standard Microsoft Foundation Class Library (MFC) version 2.52b. Besides classes for OLE and database, this version includes classes for OLE control development.

Bugs Fixed in Visual C++ 1.52b

Visual C++ 1.52b fixes a number of bugs from Visual C++ 1.5. Of particular interest to most developers are the following bug fixes:

MFC 2.52b

Error L2025 occurred on CWnd::DoDataExchange. This bug has been corrected. See Microsoft Knowledge Base article Q120152 for more information.

CArchive::ReadObject sometimes caused an assertion in a CPtrArray object because CPtrArray can hold a maximum object size of 16K in a large memory model application. CArchive did not take the memory model into account; since CArchive enforced a 32K limit, objects of 32K could be written but an assertion was generated if CArchive read in more than 16K. This bug has been corrected.

There was a memory leak in 16-bit AUX_DATA.

The three pens used in the class CPropertySheet were leaking. See Microsoft Knowledge Base article Q128604 for more information.

The CArchive buffer pointer could wrap around the end of a segment unintentionally. This bug has been corrected.

If AfxSockInit failed, applications sometimes produced a General Protection Fault. This bug has been corrected. See Microsoft Knowledge Base article Q130653 for more information.

The MAPISendMail dialog did not stay modal. This bug has been corrected.

Source file DLGPROP.CPP had non-near data. This could prevent applications from being able to run multi-instance. This bug has been corrected.

3.6.4. MFC 2.52c

This bug-fix release shipped with Visual C++ 4.0 on a separate CD-ROM.

%%TODO check release notes for details - is anyone still using this stuff?

3.7. MFC 3.0

MFC 3.0 shipped with Visual C++ 2.0 in 9/94. It introduced docking toolbars, property sheets and template based collection classes. This was the first release in the MSVC subscription, it was a 32-bit release.

Classes added in MFC 3.0:

Tabbed Dialog Support:

- CWnd
 - CPropertySheet
 - CDialog
 - CPropertyPage
-

3.8. MFC 3.1

MFC 3.1 shipped with Visual C++ 2.1 in 1/95. It introduced MAPI, WinSock and Windows Common Controls. The MFC toolbar, status bar, etc.. still live in MFC. This is the latest release out. It is a 32-bit release.

NOTE: Only available via MSVC subscription.

Classes Added in MFC 3.1:

MAPI: [Note that MAPI support was added to CDocument, no new classes]

- CDocument::OnFileSendMail
- CDocument::OnUpdateFileSendMail
- COleDocument::OnFileSendMail

Windows Common Controls Classes: [All of these are CWnd derived]

- CAnimateCtrl
- CHeaderCtrl
- CHotkeyCtrl
- CImageList
- CListCtrl
- CProgressCtrl
- CSliderCtrl
- CSpinButtonCtrl
- CStatusBarCtrl
- CTabCtrl
- CToolBarCtrl
- CToolTipCtrl
- CTreeCtrl

WinSock Support:

- CAsyncSocket
 - CSocket
-

3.9. MFC 3.2

MFC 3.2 shipped with Visual C++ 2.2 via the subscription in mid/late July. It is a 32-bit release. Includes a 2.52b 16-bit release update.

[here's an excerpt from the release notes]

Support for Windows Common Controls in USRDLLs. In order to use the MFC Windows Common Control classes you must link in the appropriate static libraries, as shown in the following table:

Library	Target
DAFXCC.LIB	Release Version
DAFXCCD.LIB	Debug Version
DAFXCCU.LIB	Release Unicode Version
DAFXCCUD.LIB	Debug Unicode Version

Updated MFC Common Control classes that work with those controls that have been recently released in Windows NT 3.51 and those that will be available in Windows 95 and a future version of Win32s.

Improved common control documentation that is more complete and reflects changes since the first implementation.

New MFC sample applications

- **NOTEPAD +** Notepad+ is an enhanced version of Notepad. The enhancements are mainly from questions our customers have been asking.
- **SAVER** Saver is a screen saver written using MFC. It "morphs" the text "MFC" to the text "C++".
- **OLEVIEW** The OLEVIEW sample is very similar to the OLE2VIEW.EXE applet provided in \MSVC20\BIN. The sample illustrates how to implement OLE Object viewers through custom OLE interfaces. These custom interfaces are implemented in IVIEWERS.DLL
- **BATCH** BATCH is an MDI application that lets you compress different AVI (Audio Video Interleaved) files in different ways.

- **WORDPAD** WORDPAD illustrates use of the CRichEditCtrl and CRichEditView classes, which encapsulate the new Rich Edit common control.

These samples are located in the SAMPLES\MFC directory of your Visual C++ installation.

Updated Win32 SDK Components

With the release of Windows NT 3.51 and the upcoming release of Windows 95, the Win32 SDK has been updated to support both of these operating systems. Changes include new header files, new import libraries, and additional documentation. Visual C++ 2.2 includes the latest header files, import libraries and Win32 API documentation to support writing Windows NT- and Windows 95-compatible applications.

3.10. MFC 4.0

MFC 4.0 was released with Visual C++ 4.0 in November of 1995. Microsoft skipped from Visual C++ 2.2 to 4.0 to synchronize numbers, which they later broke in VC++ 5.0, but that's life I guess.

Classes added in MFC 4.0:

- CSynchronize
- CMutex
- CEvent
- CMultiLock
- CShellNew - *Windows 95*

Visual C++ 4.0 also includes the component gallery, STL support and tons of new features. Check out the Microsoft WWW for the details.

See FAQ section 11.24 and 13.7 for MFC/VC++ 4.0-specific questions.

3.11. MFC 4.1

The most important thing about this release is that it is the last release that supports Win32s. There are tons of MFC developers stuck on this release, so if you see someone posting that they are using MFC 4.1, that is probably why. This fixed many of the 4.0 bugs and added some new Internet features.

%%TODO - get the date and a more comprehensive list of what was new. Can't remember if ISAPI was in here yet, but I'm pretty sure it had the "sweeper" stuff - HTTP, gopher and ftp classes.

3.12. MFC 4.2

ISAPI classes added and actually usable OCX container support (still up for debate IMO).

%%TODO - get the date and more features.

3.12.1. MFC 4.2b

MFC 4.2 had numerous bugs in the areas of the Internet classes, OLE controls, etc.

%%TODO - again, more specifics.

3.12.2. MFC 4.21

Released on March 19, 1997 as part of Visual C++ 5.0, this is the latest and greatest MFC release. The only new feature added to this release is IntelliMouse™ support which evidently took the better part of an afternoon for .B ekiM to bang out. This release also includes another round of bug fixes and I would say is by far the most stable MFC since 3.2 (this is due to all of the new features added, which is understandable).

Unfortunately, we now have a situation where the MFC release doesn't match VC++ - everybody on the newsgroups thinks they are using MFC 5.0, which doesn't exist. Help me set them straight!!

3.13. Table of MFC Releases

Hint, MFC releases are always MSVC release - 1. This is because MFC 1.0 came out with MS C 7. This was all changed with 4.0 of course, now the formula is much easier: MSVC release = MFC release, well until VC++ 5.0 came out < sigh >.

MFC Release	MSVC Release	16 or 32 Bit	Notes
1.0		16	Just thin Windows coverage
2.0	1.0	16	Document/Views added
2.1	1.1 for NT	32	First 32-bit release for NT
2.5	1.5	16	OLE/ODBC, last 16-bit release
2.51	2.0	16	Bug Fixes
2.52	2.1	16	Adds Property Sheets to 2.5
2.52b	2.2	16	Shipped in July '95, bug fixes
2.5c	4.0	16	Shipped in Nov '95, bug fixes
3.0	2.0	32	Property sheets, dock toolbars
3.1	2.1	32	Winsock/MAPI, Win comctrls
3.2	2.2	32	Shipped in July '95, more comctrls
4.0	4.0	32	Win 95, thread classes, OCX containers
4.1	4.1	32	Tons of bug fixes, sweeper (WinInet) classes. Was last release to support Win32s.
4.2	4.2	32	More bug fixes, ISAPI classes
4.2b	internet dl	32	Lots-o-bug-fixes!
4.21	5.0	32	IntelliMouse™ support.

4. Exception Handling

4.1. [CException - Exceptions and Exception Handling](#)

4.1.1. [How do I throw a CUserException derived exception?](#)

4.1.2. [Do my exceptions have to be derived from CUserException?](#)

4.1. CException - Exceptions and Exception-handling.

4.1.1. How do I throw a CUserException derived exception?

When I try to catch a derived exception I get the following error: error C2039: 'classCMyException' : is not a member of 'CMyException' 'classCMyException' : undeclared identifier 'IsKindOf' : cannot convert parameter 1 from 'int*' to 'const struct CRuntimeClass*'

You need to make your CMyException class dynamically creatable using the DECLARE_DYNAMIC() and IMPLEMENT_DYNAMIC() macros. The CATCH macro expects to be able to access run-time information about the thrown class.

- *blaszczak@bix.com, Mike B, 6/5/95, comp.os.ms-windows.programmer.misc*

4.1.2. Do my exceptions have to be derived from CUserException?

No: The "User" in CUserException simply means that the exception is caused by user actions. It is a common misperception that this is the only exception you can derive from.

-*nuj@kruger.dk via email 11/23/95*

5. GDI Class Questions

5.1. [CDC](#)

5.1.1. [How do I create a CDC from a HDC?](#)

5.2. [CBitmap](#)

5.2.1. [How do I read a 256 color bitmap file from disk?](#)

5.1. CDC

5.1.1. How do I create a CDC from a HDC?

Sometimes the Windows API will just give you a handle to a DC and you might want to create a CDC from that. One example is owner-drawn lists, combos, and buttons. You will receive a draw item message with a hDC. Here's some code to turn that hdc into the more familiar CDC.

You can use this technique for any of the other MFC class/ Windows handle pairs too.

```
void MyODList::DrawItem(LPDRAWITEMSTRUCT lpDrawItem)
{
    CDC myDC;
    myDC.Attach(lpDrawItem->hDC);
    //Do more stuff here
    //If you don't detach, it will get deleted and windows will
    //not be happy if you delete it's dc..
    myDC.Detach();
}
```

Another approach is to call the CDC FromHandle method:

```
CDC* pDC = CDC::FromHandle(lpDrawItem->hDC);
```

It's not clear which is 'better', FromHandle() is less error prone because you do not have to remember to 'detach'.

jmccabe@portage1.portup.com (Jim McCabe) 6/5/95

5.2. CBitmap

5.2.1. How do I read a 256 color bitmap file from disk?

Currently, MFC does not contain direct support for reading and displaying DIBs or BMPs. However, there are a number of sample applications that illustrate just how to do this.

The first sample is the MFC sample application DIBLOOK. The MULTDOCS sample uses the same source code provided by DIBLOOK to read and display DIBs and BMPs.

Two other examples provided with Visual C++ are the SDK samples DIBVIEW and SHOWDIB.

Question posted on comp.lang.c++, 6/5/95, found this in MS FAQ, 6/25/95

6. Windows, Controls, and Dialog Class Questions

6.1. [Windows](#)

- 6.1.1. [How can I use a custom icon for a window?](#)
- 6.1.2. [How do I change the styles for a window that's created by MFC?](#)
- 6.1.3. [How do I get the minimal size of a window using MFC?](#)
- 6.1.4. [How do I change a Window's title?](#)
- 6.1.5. [How do I get rid of 'Untitled' in my main window caption?](#)
- 6.1.6. [How do I maximize my MDI child?](#)
- 6.1.7. [Why does focus go nutso with a CSplitterWnd?](#)
- 6.1.8. [How do I make my first MDI child window start out maximized?](#)
- 6.1.9. [How do I make a window stay on top of the others?](#)
- 6.1.10. [How do I make a window move in front of another?](#)
- 6.1.11. [How do I implement docking windows like DevStudio has?](#)
- NEW!!** 6.1.12. [Why is the GetSafeHwnd\(\) function needed, when would a HWND may be unsafe?](#)

6.2. [CSplitterWnd](#)

UPD!!

- 6.2.1. [How do I create non-resizable \(static AND fixed\) splitter bars?](#)

6.3. [Controls](#)

- 6.3.1. [How do I get a CControl from a Dialog Template?](#)
- 6.3.2. [How do I subclass a control using MFC?](#)
- 6.3.3. [Why do I get an ASSERT when I subclass a control?](#)
- 6.3.4. [How do I validate the contents of a control when it loses focus?](#)
- 6.3.5. [How do I enable/disable a bank of checkboxes?](#)
- 6.3.6. [How do I change the background color of a control?](#)
- 6.3.7. [How do I trap the key for my control?](#)
- 6.3.8. [How can I DDX with a multiple selection listbox?](#)
- 6.3.9. [How do I change the background color of a BUTTON???](#)
- 6.3.10. [Why isn't CEdit putting things on separate lines?](#)

6.3.11. [How do I get to the CEdit in a combobox?](#)

UPD!! 6.3.12. [How do I load more than 64K into an edit control?](#)

6.3.13. [How do I subclass the listbox portion of a combobox?](#)

6.3.14. [How do I inherit a from a MFC standard control class?](#)

NEW!! 6.3.15. [How do you add controls to a CDialog dynamically, instead of using a dialog resource?](#)

NEW!! 6.3.16. [Why is my fixed-height owner drawn listbox's MeasureItem never called?](#)

6.4. [Windows Common Controls \(A.K.A. Windows 95 Controls\)](#)

6.4.1. [Can I use these controls under NT or Win32s?](#)

6.4.2. [Where's a demo of these wickedly cool controls?](#)

6.4.3. [How do you handle NM_DBLCLK for a CListCtrl?](#)

6.4.4. [Does CTreeCtrl support multiple selection?](#)

6.4.5. [When I expand a node in my CTreeCtrl, there's no visual feedback, what to do?](#)

6.4.6. [How do I get multiple selection, tooltips, editable nodes, multiple columns in my trees?](#)

NEW!! 6.4.7. [How do I tell a CListCtrl object to select the whole line and not just the left column, when you click on an item?](#)

NEW!! 6.4.8. [How do you turn off the sliding animation that happens when a TreeCtrl is expanded or collapsed?](#)

NEW!! 6.4.9. [When implementing drag/drop operations, how do you get the different cursors: copy, shortcut, circle with the slash, etc. ?](#)

6.5. [Dialogs](#)

6.5.1. [How do I center my dialog?](#)

6.5.2. [How do I get the 'old style' common dialogs on win95?](#)

UPD!! 6.5.3. [How do I subclass a win95 common dialog?](#)

6.5.4. [CDialog::Create\(\) fails, what could be wrong?](#)

6.5.5. [How do I create a toolbar/statusbar in a dialog?](#)

6.5.6. [Why isn't my CDialog::PreCreateWindow\(\) getting called?](#)

6.5.7. [How do I embed a common dialog in a property page?](#)

6.5.8. [Why can't I DDX/DDV to initialize my CDialog controls?](#)

6.5.9. [How do I change the captions of a CPropertyPage?](#)

6.5.10. [How do I trap F1 in my dialog?](#)

6.5.11. [How do I change the icon for a dialog-only MFC application?](#)

NEW!! 6.5.12. [How to I get rid of the Help and Apply buttons on my CPropertySheet-derivative?](#)

NEW!! 6.5.13. [How do you add a resize 'handle' to a dialog?](#)

6.1. Windows

6.1.1. How can I use a custom icon for a window?

The Microsoft Foundation Class Library stores icons for the main frame window and the MDI frame window as resources. The icon with resource ID `AFX_IDI_STD_MDIFRAME` is the icon for the MDI frame window, and the icon with resource ID `AFX_IDI_STD_FRAME` is the icon for the main frame window. To replace these icons in your application, add an icon to your resources file with the appropriate ID.

The application specifies the icon for a view in an MDI child window when it creates the template. The application uses the icon with the specified resource ID when the user minimizes the MDI child window that contains the corresponding view.

This technique allows you to specify one icon for the application to associate with these windows. Windows also supports dynamically painting a minimized window. To do this with MFC, use `AfxRegisterWndClass()` to register a window class with a `NULL` icon handle. Override the `PreCreateWindow()` function in the window class for the dynamically painted icon and copy the name returned by `AfxRegisterWndClass()` into the `lpszClassName` member of the `CREATESTRUCT`. This creates the window using the class that has a `NULL` icon. When the user minimizes this window, the icon receives `WM_PAINT` messages that it can process to display information appropriately. To do so, override the `OnPaint()` message handler and call the `IsIconic()` function to see if the window is minimized. If so, create a `CPaintDC` object and use it to draw on the icon. If the window is not minimized, call the base class version of `OnPaint()` to update the window normally.

MSVC Knowledge Base 6/4/94

6.1.2. How do I change the styles for a window that's created by MFC?

To change the default window attributes used by a framework application

created in AppWizard, override the window's PreCreateWindow() virtual member function. PreCreateWindow() allows an application to access the creation process normally processed internally by the CDocTemplate class. The framework calls PreCreateWindow() just prior to creating the window. By modifying the CREATESTRUCT structure parameter to PreCreateWindow(), your application can change the attributes used to create the window.

The CTRLBARS sample application, provided with the Microsoft Foundation Class Library version 2.0, demonstrates this technique to change window attributes. Note that depending on what your application changes in PreCreateWindow(), it may be necessary to call the base class implementation. For more information, see MSVC knowledge base article Q99847.

MSVC Knowledge Base 6/7/95

6.1.3. How do I get the minimal size of a window using MFC?

Write a handler for WM_GETMINMAXINFO.

blaszczak@BIX.com, Mike Blaszczak, 6/12/95 via programmer.misc

6.1.4. How do I change a Window's title?

```
AfxGetApp()->m_pMainWnd->SetWindowText("My Window Title");
```

-OR-

```
AfxGetMainWnd()->SetWindowText ("My Own Title" );
```

aj536@freenet.toronto.on.ca, mfc-l, 7/9/95

6.1.5. How do I get rid of 'Untitled' in my main window caption?

Override the PreCreateWindow() function in your MainFrame class and do the following in it..

```
cs.style &= ~FWS_ADDTOTITLE ;
```

You can also set the initial window position (cs.x, cs.y, cs.cx, cs.cy) this way and change your class (cs.lpszClass) this way! **Remember to call CFrameWnd::PreCreateWindow at the end...**

netninja@svpal.org, programmer.misc, 7/29/95

6.1.6. How do I maximize my MDI child?

```
void CMainFrame::ActivateFrame(int nCmdShow)
{
    if (!m_bActivated)
    {
        m_bActivated = TRUE;
        nCmdShow = SW_SHOWMAXIMIZED;
    }
    CFrameWnd::ActivateFrame(nCmdShow);
}
```

where m_bActivated is a member variable of your frame object.

duane@anasazi.com, programmer.win32, 8/3/95

6.1.7. Why does focus go nutso with a CSplitterWnd?

Whenever I move the splitter bar, the I-beam cursor in my edit control goes away. I have to click again in the edit control to get back the cursor.

The following Knowledge Base Article (GO MSKB on CIS) explains the focus problem associated with splitter windows and a couple of work-arounds to the problem. This may be of help to you.

ID: Q108434

FIX: CSplitterWnd Class Does Not Handle All Focus Cases

Ramesh, NetQuest., MSMFC, 8/3/95

6.1.8. How do I make my first MDI child window start out maximized?

Here is a solution which works for me:

```
class CChildFrame : public CMDIChildWnd
{
    // .. stuff deleted ...

    // This makes the MDI child maximized.
```



```

virtual void ActivateFrame(int nCmdShow)
{
    // if another window is open, use default
    if(GetMDIFrame()->MDIGetActive())
        CMDIChildWnd::ActivateFrame(nCmdShow);
    else // else open maximized.
        CMDIChildWnd::ActivateFrame(SW_SHOWMAXIMIZED);
}

// ... stuff deleted ...
};

```

- Stephen Bade, bade@convergent-design.com

6.1.9. How do I make a window stay on top of the others?

```
SetWindowPos(&wndTopMost, NULL, NULL, NULL, NULL, SWP_NOMOVE | SWP_NOSIZE)
```

(like DBWIN does)

lee@usa.nai.net, via mfc-l, 1/19/95

6.1.10. How do I make a window move in front of another?

Call either:

```
SetWindowPos(&wndTop, NULL, NULL, NULL, NULL, SWP_NOMOVE | SWP_NOSIZE)
```

Or

```
BringWindowToFront();
```

lee@usa.nai.net, via mfc-l, 1/19/95

6.1.11. How do I implement docking windows like DevStudio has?

MFC does not easily let you do this. The problem is that the dockbar/control bar architecture is built for basic toolbars, not windows. We have solved the problem in our Objective Toolkit product - <http://www.stingray.com> has demos and white papers - check it out!

NEW!! 6.1.12. Why is the GetSafeHwnd() function needed, when

would a HWND may be unsafe?

If you see the MFC source code in `afxwin2.inl` in `msdev\mfc\include` directory, you will realize why you need `GetSafeHwnd()` function. Its implementation goes something like this :

```
_AFXWIN_INLINE HWND CWnd::GetSafeHwnd() const  
{ return this == NULL ? NULL : m_hWnd; }
```

which does nothing other than checking for a `CWnd` pointer and if it is not null returning the `m_hWnd` member variable of it. There is a possibility of `CWnd` is null and `m_hWnd` is junk so `GetSafeHwnd()` is a real useful function. It does the checking for you about window pointers, which are the part of Windows `PtrToHandle` maps. Just check about handlemaps which are permanent and temporary and then you will realize the importance of that function.

Chandra Sekhar Rentachinthala, chandra@india.aspectdv.com, mfc-1, 6/28/98

6.2. CSplitterWnd FAQs

%%TODO - add lots of these, email us your toughest ones.

FYI - [MFC Internals](#) has a whole chapter devoted to this - it's my personal favorite MFC class because it has lots of interesting internals, shortcomings, odd behaviors, etc..

We wrote a DDJ article where we show how to swap the splitter from horizontal to vertical. I think the code is at <http://www.ddj.com> if you are interested.

-scot@stingray.com - Scot Wingo

UPD!! 6.2.1. How do I create non-resizable (static AND fixed) splitter bars?

See KB article Q105150 "Preventing Static Splitters from Tracking". It describes how to do this. Basically, derive a class from the CSplitterWnd and override the left button and mouse move messages: call the CWnd versions rather than CSplitterWnd versions. Finally, remove menu items that may generate ID_WINDOW_SPLIT commands.

bills@anest4.anest.ufl.edu, via *mfc-l* 9/28/95

To add to the above answer, you should also handle the WM_SETCURSOR message and just call the CWnd version so that the horizontal and vertical resize pointers are not shown.

```
BOOL CMySplitterWnd::OnSetCursor(CWnd* pWnd, UINT nHitTest, UINT message)
{
    return CWnd::OnSetCursor(pWnd, nHitTest, message);
}
```

Michael Pickens, michaelp@stingray.com

6.3. Controls

6.3.1. How do I get a CControl from a Dialog Template?

You can get a pointer to a control from a already created dialog control by doing a simple typecast of the results from GetDlgItem. Here's an example that creates a CButton from a checkbox with ID : IDC_CHECK1.

```
void my_function(CDialog * pDialog)
{
    CButton* pButton = (CButton*)pDialog->GetDlgItem(IDC_CHECK1);
    ASSERT(pButton != NULL);
    pButton->SetCheck(m_bShowState);
}
```

Note that it's always safer to check for the validity of the results from GetDlgItem.

scot@stingray.com, 6/1/95

6.3.2. How do I subclass a control using MFC?

Read the documentation on SubClassDlgItem. Here's an example of how to call it:

```
BOOL CMyDialog::OnInitDialog()
{
    //Do your subclassing first.
    m_MyControl.SubClassDlgItem(ID_MYCONTROL, this);

    //Let the base class do its thing.
    CDialog::OnInitDialog();

    // Perhaps do some more stuff
    // Be sure to call Ctl3d last, or it will cause
    // assertions from multiple subclassing.
    Ctl3dSubclassDlg(m_hWnd, CTL3D_ALL);
}
```

Mike Williams, mikew@marlin.ssnet.com, mfc-1 6/1/95

6.3.3. Why do I get an ASSERT when I subclass a control?

Make sure that you subclass the control BEFORE you call Ctl3dSubclassDlg, if the 3-d control DLL is loaded first, it will already have subclassed your controls and you will get an assert.

Mike Williams, mikew@marlin.ssnet.com, mfc-1 6/1/95

6.3.4. How do I validate the contents of a control when it loses focus?

NOTE: This is in the Microsoft Software Library.

The FCSVAL sample application was created to show how an application can do control-by-control validation in a dialog box.

The application itself is just a modal dialog box displayed by the CWinApp::InitInstance(). After displaying the dialog box, InitInstance() simply quits the application.

The important part of the sample takes place in the dialog-box class implementation: There are two edit controls. The first takes input of an integer between 1 and 20. The second takes a character string as input with length less than or equal to 5. When you Tab or mouse-click from control to control within the displayed dialog box, the contents of the control that is losing focus are validated.

The CFocusDlg Class

The application's functionality centers around the CFocusDlg class and its implementation of four message handlers (discussed below). Normal data exchange (DDX) and validation (DDV) using the routines provided by MFC take place in OnInitialUpdate(), when the dialog box is first displayed, and when the user chooses the OK button to accept the input. This is default behavior provided by ClassWizard when member variables are connected to dialog-box controls and can be examined in the dialog class DoDataExchange() function.

Validating control contents when switching focus from one control to the next is done by handling the EN_KILLFOCUS notification sent by the edit control that

is losing focus. The idea here is to check the contents and, if they are not valid, to display the message box, inform the user, and then set the focus back to the control from which it came. Unfortunately, some difficulties arise when trying to set the focus (or display the message boxes) within a killfocus message handler. At this point, Windows is in an indeterminate state as it is moving focus from one control to the other. This is a bad place to do the validation and SetFocus() call.

The solution here is to post a user-defined message to the dialog box (parent) and do the validation and SetFocus() there, thus waiting for a safer time to do the work. (See "CFocusDlg::OnEditLostFocus()" in the file FOCUSDLG.CPP and "WM_EDITLOSTFOCUS user-defined message" in the file FOCUSDLG.H.) Another thing you will notice about this function is that it uses TRY/CATCH to do the validation. The provided DDX/DDV routines throw CUserExceptions when failing to validate or load a control's data. You should catch these and do the SetFocus() in the CATCH block.

Note: This sample has other cool stuff, but this is the major one I've seen asked about on the Net.

MS FAQ, 6/25/95

6.3.5. How do I enable/disable a bank of checkboxes?

I don't know about a magic way to do this using a single HWND, but there is a simple and self-documenting technique that I've been using for a long time. You can make a routine that accepts an array of UINTs (your control IDs) and a visibility flag. This function can be a stand-alone function, or you can put it inside a class. I have been collecting little utility functions like this and keep them in a CDialogBase class -- when I create a new dialog box in ClassWizard, I fix up the code to derive from CDialogBase instead of CDialog.

For example, the function might look like this:

```
void CDialogBase::ShowControls(UINT* pControls, UINT cControls, BOOL fVisible)
{
    for (UINT uIndex = 0; uIndex < cControls; uIndex++)
    {
        CWnd* pwnd = GetDlgItem(pControls[uIndex]);
        if (pwnd)
        {
```

```

        pwnd->ShowWindow(fVisible ? SW_SHOW : SW_HIDE);
        pwnd->EnableWindow(fVisible);
    }
}
}

```

Then later, often in your OnInitDialog handler, you can call this function with your control group:

```

#define SIZEOF_ARRAY(a) (sizeof(a) / sizeof(a[0]))
{
    static UINT aGroup1[] = { DLG_CHBOX1, DLG_CHBOX2, DLG_STATIC1 };
    static UINT aGroup2[] = { DLG_LABEL2, DLG_LABEL7 };
    ShowControls(aGroup1, SIZEOF_ARRAY(aGroup1), TRUE);
    ShowControls(aGroup2, SIZEOF_ARRAY(aGroup2), FALSE);
}

```

You can find many uses for these control arrays later too... (Changing fonts in a series of controls, etc...) Good luck,

jmccabe@portage1.portup.com, mfc-l, 7/18/95

6.3.6. How do I change the background color of a control?

Your dialog can trap the WM_CTLCOLOR message, look up the MFC help file notes for CWnd::OnCtlColor(). Before a control is about to paint itself, the parent window receives a chance to set its own default text color and background brush.

jmccabe@portage1.portup.com, mfc-l, 7/18/95

Also check out the MS KB article ID: Q117778 TITLE: Changing the Background Color of an MFC Edit Control.

Ramesh, MSMFC, 7/19/95

6.3.7. How do I trap the key for my control?

Handle WM_GETDLGCODE and return the appropriate value. Remember that the listbox (or any other control) can only handle keyboard input when it has the focus.

joej@golddisk.com, programmer.misc, 8/21/95, programmer.misc

6.3.8. How can I DDX with a multiple selection listbox?

Download MLBDDX.ZIP from the MSMFC library on CIS. You'll get all the necessary code. When the dialog closes, a provided CStringList will be filled with the selected items. Freeware.

-Patrick Philippot, CIS email, 8/3/95

6.3.9. How do I change the background color of a BUTTON???

NOTE: THE METHOD IN 6.3.6 WILL NOT WORK FOR BUTTONS!

If you want to change the color of a dialog button, you have to use owner-draw button. (you can use bitmap buttons) Changing the color through OnCtlColor() will not work for buttons. The following Knowledge Base articles (GO MSKB on CIS) may be of help to you.

ID: Q32685 TITLE: Using the WM_CTLCOLOR Message

ID: Q64328 SAMPLE: Owner-Draw: 3-D Push Button Made from Bitmaps with Text

This article explains sample code for a owner-draw button.

Ramesh, NetQuest., MSMFC, 8/3/95

6.3.10. Why isn't CEdit putting things on separate lines?

Make sure that the lines are separated with `\r\n`, not just `\n`.

sutor@watson.ibm.com, mfc-l, 8/7/95

6.3.11. How do I get to the CEdit in a combobox?

```
CComboCox combo;  
CEdit edit;  
// combobox creation ...  
// ...  
POINT tmpPoint = {1,1};  
edit.SubclassWindow( combo.ChildWindowFromPoint(tmpPoint)->GetSafeHwnd() );
```


jahans@slb.com, mfc-1, 8/25/95

Or:

Look into the mfc sample - npp - npview.cpp! Turns out all combo's create their edits with an ID of 1001 (decimal) so - if pComboBox is the pWnd object pointing to the combo - all you need is:

```
pComboBox->GetDlgItem(1001);
```

UPD!! 6.3.12. How do I load more than 64K into an edit control?

The Rich Edit Control available in VC++ 2.1+ supports much more than 64k. The Wordpad sample is a great way to learn more about this subject. If you're stuck with 16-bit programming, I think that magma systems has a 16-bit DLL that does this. Contact Marc Adler at: 75300.2062@compuserve.com for details.

scot@stingray.com

I have found under NT that unless you use SetLimitText(0) (or send a EM_SETLIMITTEXT with WPARAM = 0) you are limited to adding about 32K programatically. You dont get any warning or EN_ERRSPACE; you just loose the end of text you add.

Stephen Lee [stephenl@digita.com]

6.3.13. How do I subclass the listbox portion of a combobox?

The listbox portion of a combobox is of type COMBOLBOX (notice the 'L'). Because the ComboLBox window is not a child of the ComboBox window, it is not obvious how to subclass the COMBOLBOX control. Luckily, under the Win32 API, Windows sends a message to the COMBOBOX (notice no 'L') called WM_CTLCOLORLISTBOX before the listbox is drawn. The lParam passed with this message contains the handle of the listbox. For example:

```
LRESULT CFileUpdateCombo::OnCtlColorListBox(WPARAM wParam, LPARAM lParam)
{
    if ( ! m_bSubclassedListBox )
    {
        HWND hWnd = (HWND)lParam;
        CWnd* pWnd = FromHandle(hWnd);
```

```

    if ( pWnd && pWnd != this )
    {
        // m_ListBox is derived from CListBox
        m_ListBox.SubclassWindow(hWnd );
        m_ListBox.SetOwner(this);
        m_bSubclassedListBox = TRUE;
    }
}
return (LRESULT)GetStockObject(WHITE_BRUSH);
}

```

mikem@abelcomputers.com, email, 9/7/95

6.3.14. How do I inherit from a MFC standard control class and provide initialization code that works on both subclassed and non-subclassed controls?

[ed note: Ok, this probably isn't a FAQ, but I thought it sounded pretty cool.]

I have a fix, but you may not like it; however, it takes care of both subclassing methods.

If SubclassWindow() was virtual, all problems would be solved, as SubclassDlgItem calls SubclassWindow(), and common initialization could be called from this point, and from OnCreate(). Even better would be a virtual SetupWindow() function called from all initialization points by Microsoft's code.

C'est la vie. My fix might slow the message loop for the control in question, but so far I haven't seen any performance hits. Over-ride the virtual function WindowProc() for your control something like the following (call SetupWindow() in OnCreate() also):

```

LRESULT CExtendControl::WindowProc( UINT message, WPARAM wParam, LPARAM lParam)
{
    if (!m_bSetup)
        SetupWindow();
    return CEdit::WindowProc(message, wParam, lParam );
}
//This is a virtual function. Use it for Hwnd setup in all inherited
//classes. It will work for a subclassed window.
void CExtendControl::SetupWindow()
{
    ASSERT( m_hWnd );
    m_bSetup = TRUE;
}

```

```
    /*** Insert Initialization Code here!***  
}
```

Jody Power (jodyp@andyne.on.ca)

NEW!! 6.3.15. How do you add controls to a CDialog dynamically, instead of using a dialog resource?

You can add controls to your dialog dynamically by using methods `CWnd::Create()` and `CWnd::CreateEx()` or overridables of `CWnd::Create()` in control window wrapper classes such as `CEdit` or `CListBox`, etc. For example, to create `CEdit` control, you can do the following:

- 1) add a member variable `m_ec_myedit` to your dialog `.h` file;
- 2) I assume that your dialog templete has some control with ID = `IDC_ABOVE_DYNAMIC_EDIT`, and you want your dynamically created edit control to have the same width and be placed under `IDC_ABOVE_DYNAMIC_EDIT`. Then add the following code under the call to `CDialog::OnInitDialog()` in your overridden `OnInitDialog()`:

```
GetDlgItem(IDC_ABOVE_DYNAMIC_EDIT)->GetWindowRect(rect);  
ScreenToClient(rect);  
CRect rectNew(rect.left, rect.bottom+5, rect.right, rect.bottom+35);  
m_myEdit.CreateEx(WS_EX_CLIENTEDGE, "EDIT", NULL  
/*lpszWindowName*/,  
WS_CHILD|WS_VISIBLE|WS_GROUP|WS_TABSTOP|WS_BORDER, rectNew.left, rectNew.top,  
rectNew.Width(), rectNew.Height(), this->GetSafeHwnd(), NULL, NULL);  
m_myEdit.ShowWindow(SW_SHOW);
```

It's that simple. The only thing that differs for different control classes is window styles. Usually, you can find the most important of style and extended style constants in online help.

DEMPSEY@DEMPSEY.COM, mfc-1, 6/12/98

NEW!! 6.3.16. Why is my fixed-height owner drawn listbox's MeasureItem never called?

The `MeasureItem` function for a fixed-height owner drawn is only called once. The problem is that it is called before the MFC listbox object is associated with

the Windows listbox control. The solution is to invoke the listbox's MeasureItem from the OnMeasureItem function of the dialog containing the listbox:

```
void CExampleDlg::OnMeasureItem(int nIDCtl, LPMEASUREITEMSTRUCT lpMeasureItemStruct)
{
    CDialog::OnMeasureItem(nIDCtl, lpMeasureItemStruct);

    if (nIDCtl == IDC_LISTBOX)
        m_ListBox.MeasureItem(lpMeasureItemStruct);
}
```

Eric Bergman-Terrell, ericterrell@juno.com, 5/16/97

6.4. Windows Common Controls (a.k.a. Windows 95 controls)

6.4.1. Can I use these controls under NT or Win32s?

Windows NT 3.50 does not support the common controls, and will not in the future. You must use Windows NT version 3.51 to gain the common controls.

blaszczak@BIX.com, mfc-l, 7/6/95

Version 1.30 of Win32s supports the Common controls.

6.4.2. Where's a demo of these wickedly cool controls?

Check out the MFC sample, fire, it features most of the controls in action: MSVC20\samples\mfc\fire. Under 4.0, there's another one called cmnctrls.

scot@stingray.com, 7/27/95

6.4.3. How do you handle NM_DBLCLK for a CListCtrl?

```
BEGIN_MESSAGE_MAP(CListView, CView)
    ON_NOTIFY( NM_DBLCLK, ID_LISTCTRL, OnDbClick )
END_MESSAGE_MAP()

void CListView::OnDbClick(NMHDR* /*k*/, LRESULT* /*j*/)
{
    int nItem, nFlags;
    char szTest[80];
    nFlags = LVNI_SELECTED;
    nItem = m_ListCtrl->GetNextItem(-1, nFlags);
    if ( nItem != -1 )
    {
        sprintf( szTest, "Selected Item %d", nItem);
        AfxMessageBox(szTest);
    }
}
```

spolyak@interaccess.com, mfc-l, 7/21/95

6.4.4. Does CTreeCtrl support multiple selection?

No. Sorry! (I see this one all the time!)

scot@stingray.com

If it's accessible to you, Microsoft Systems Journal, July 1994, has an extensive look at TreeViews, including a sample program that includes drag & drop. If you're developing for Windows 95, you don't need a shareware implementation, MFC has a class for it.

steven@primenet.com

6.4.5. When I expand a node in my CTreeCtrl, there's no visual feedback, what to do?

Create the control with TVS_SHOWSELALWAYS style.

kfreeman@viewlogic.com

6.4.6. (The FAQ of the devil!) How do I implement multiple selection, tooltips, editable nodes and multiple columns in my tree control?

Unfortunately, since the tree control lives in a binary DLL, it is not very extensible and you are stuck without these features. There have been articles on how to try and "hack" these features, but they do not work reliably.

We have a drop-in tree control replacement that implements all of these features and more in Objective Toolkit. Demos are at <http://www.stingray.com>.

NEW!! 6.4.7. How do I tell a CListCtrl object to select the whole line and not just the left column, when you click on an item?

The easiest way of getting full row select is if you use the updated common controls DLL (versions >= 4.70) supplied with IE. With that you can use the LVM_SETEXTENDEDLISTVIEWSTYLE message to set the LVS_EX_FULLROWSELECT style.

mfc-l, 6/21/98

NEW!! 6.4.8. How do you turn off the sliding animation that happens when a TreeCtrl is expanded or collapsed?

I have used LockWindowUpdate() and UnlockWindowUpdate() in the past to achieve what you are asking using a CListCtrl. The functions are in CWnd and since CTreeCtrl is derived from CWnd this should work the same.

Ted Dasler [daslert@DGABBY.MFLDCLIN.EDU], mfc-l, 6/19/98

NEW!! 6.4.9. When implementing drag/drop operations, how do you get the different cursors: copy, shortcut, circle with the slash, etc. ?

I believe you want to override the OnDragEnter() and OnDragOver() methods. Returning DROPEFFECT_NONE will give you the circle with a slash. DROPEFFECT_LINK will give you the arrow. DROPEFFECT_COPY will give you the plus sign.

Cary Walker [cwalker@showcasecorp.com], VCPP mailing list, 6/29/98

6.5. Dialogs

6.5.1. How do I center my dialog?

Use the `CWnd::CenterWindow` method accomplish this. I usually put it in my `OnInitDialog` overloaded function. Since `CDialog` is an ancestor of `CWnd`, you can call the method directly:

```
BOOL CMyDialog::OnInitDialog()
{
    //Perform any other dialog initialization up here.
    CenterWindow();
    return TRUE;
}
```

scot@stingray.com, 6/1/95.

6.5.2. How do I get the 'old style' common dialogs on win95?

MFC detects if it is running on Win95, and if so, replaces the standard FileOpen Dialog with an explorer version of the FileOpen Dialog. You can prevent MFC from using this "explorer" version by adding the following line to your `CFileDialog` derived class constructor:

```
m_ofn.Flags &= ~OFN_EXPLORER;
```

andyd@andyne.on.ca (Andy DeWolfe), via programmer.win32, 5/10/95

UPD!! 6.5.3. How do I subclass a win95 common dialog?

You can do it but Microsoft has made it much more difficult in Win95. You need to create a "child dialog template" (with the `WS_CHILD` style) and set it to `m_ofn.lpTemplateName` (making sure `m_ofn.hInstance` is set to your app instance). This template must **only** contain the controls that you are adding to the dialog (ie. NOT the whole dialog with the standard controls duplicated as in Win3.x).

When the dialog is invoked, your template will appear (by default) below the regular file dialog controls. If you put a static control with id `stc32` (defined in

include\dlgs.h), the common dialog code will rearrange things so that the original controls will appear wherever you put the stc32 control (you don't have to size it to match the common dlg code will do that for you).

You will need to supply `m_ofn.lpfHook` and handle your additional controls through the hook proc. Note that since the system puts your dialog template ON TOP of the normal dialog, MFC message routing won't get to your controls so you can't code them through a message map in your `CFileDialog` derivative. If anybody has found a way around this, I'd love to hear it!!

This is very messy and Microsoft knows it. They promise a fix in MFC 4.0.[ed. note: This is much nicer in 4.0. There are virtuals to override for getting callbacks, plus it even handles the old and new style templates - pretty clever stuff!]

joej@golddisk.com, Joe Janakovic, via *programmer.win32*, 6/10/95

This is actually a documentation bug ... I haven't found this information elsewhere. If you're subclassing the common file dialog, and you want to close the dialog when the user presses a button, the entry in MSDN on `OFNHookProc` says that you should not use `EndDialog`, but that you should post a `WM_COMMAND` message to the main dialog (which is the parent of your customising dialog), specifying `IDABORT` as the control ID. This works with the old-style, non-Explorer file dialog, but not with the Explorer-style dialog. You now have to use `IDCANCEL`; and this also works with the old-style dialog.

If you're using MFC, you can write the following:

```
if (GetParent() != GetTopLevelParent())
{
    // Explorer-style
    GetParent()->PostMessage(WM_COMMAND,
        MAKEWPARAM(IDCANCEL, BN_CLICKED));
}
else
{
    // old style
    PostMessage(WM_COMMAND,
        MAKEWPARAM(IDCANCEL, BN_CLICKED));
}
```

- rick cameron [Rick.Cameron@Seagatesoftware.com], 6/18/98

6.5.4. CDialog::Create() fails, what could be wrong?

- Invalid HWND passed as a parent
- Invalid dialog resource ID passed (be careful about numeric IDs vs. string IDs -- be careful with #define ID_MYDIALOG 0x1234 -- it is a "string" ID to the resource compiler).
- One or more controls on your dialog could not be created, usually
- Because of the use of a custom control that was not registered.
- Calling EndDialog during the OnInitDialog message (or some other handler called early in the game)!
- NULL HWND passed as parent when dialog has WS_CHILD style

That's about all I can think of right now,

Dean McCrory, MSMFC, 6/16/95

6.5.5. How do I create a toolbar/statusbar in a dialog?

There's a sample in the Microsoft Software Library, DLGCBR, that demonstrates how to do this. Basically there's four steps, outlined and then coded below.....

To add a control bar to a dialog, you must create the control bar as usual, and then make room for the control bar within the client area of the dialog. For the control bar to function properly, the dialog must duplicate some of the functionality of frame windows. If you want ON_UPDATE_COMMAND_UI handlers to work for the control bars, you also need to derive new control bar classes, and handle the WM_IDLEUPDATECMDUI message. If your dialog is not the main window of your application, you will also need to modify its parent frame window to pass the WM_IDLEUPDATECMDUI message on to the dialog's control bars.

To make room for a control bar within the client area of the dialog, follow these steps in your dialog's OnInitDialog() function:

1. Create the control bars.

```
CRect rcClientStart;  
CRect rcClientNow;  
GetClientRect(rcClientStart);  
RepositionBars(AFX_IDW_CONTROLBAR_FIRST,
```

```
AFX_IDW_CONTROLBAR_LAST,0, reposQuery,rcClientNow);
```

2. Figure out how much room the control bars will take by using the reposQuery option of RepositionBars():

```
CPoint ptOffset(rcClientStart.left - rcClientNow.left,  
               rcClientStart.top - rcClientNow.top);  
ptOffset.y += ::GetSystemMetrics(SM_CYMENU);  
CRect rcChild;  
CWnd* pwndChild = GetWindow(GW_CHILD);  
while (pwndChild)  
{  
    pwndChild->GetWindowRect(rcChild);  
    rcChild.OffsetRect(ptOffset);  
    pwndChild->MoveWindow(rcChild, FALSE);  
    pwndChild = pwndChild->GetNextWindow();  
}
```

3. Move all the controls in your dialog to account for space used by control bars at the top or left of the client area. If your dialog contains a menu, you also need to account for the space used by the menu:4. Increase the dialog window dimensions by the amount of space used by the control bars:

```
CRect rcWindow;  
GetWindowRect(rcWindow);  
rcWindow.right += rcClientStart.Width()  
               - rcClientNow.Width();  
rcWindow.bottom += rcClientStart.Height()  
                 - rcClientNow.Height();  
MoveWindow(rcWindow, FALSE);
```

4. Position the control bars using RepositionBars().

To update the first pane of a status bar with menu item text, you must handle WM_MENUSELECT, WM_ENTERIDLE, and WM_SETMESSAGESTRING in your dialog class. You need to duplicate the functionality of the CFrameWnd handlers for these messages. See the CModelessMain class in the sample program for examples of these message handlers.

To allow ON_UPDATE_COMMAND_UI handlers to work for other status bar panes and for toolbar buttons, you must derive new control bar classes and implement a message handler for WM_IDLEUPDATECMDUI. This is necessary because the default control bar implementations of OnUpdateCmdUI() assume the parent window is a frame window. However, it doesn't do anything but pass the parent window pointer on to a function which only requires a

CcmdTarget pointer. Therefore, you can temporarily tell OnUpdateCmdUI() that the parent window pointer you are giving it is a CFrameWnd pointer to meet the compiler requirements. Here's an example:

```
LRESULT CDlgToolBar::OnIdleUpdateCmdUI(WPARAM wParam,LPARAM lParam)
{
    if (IsWindowVisible())
    {
        CFrameWnd* pParent = (CFrameWnd*)GetParent();
        if (pParent)
            OnUpdateCmdUI(pParent, (BOOL)wParam);
    }
    return 0L;
}
```

To pass WM_IDLEUPDATECMDUI messages on to dialogs other than the main window, save dialog pointers in your frame window class and create a WM_IDLEUPDATECMDUI handler in that class. The handler should send the WM_IDLEUPDATECMDUI message on to the dialog child windows by using CWnd::SendMessageToDescendants(). Then perform default processing for the message within the frame window.

MS FAQ 6/25/95

6.5.6. Why isn't my CDialog::PreCreateWindow() getting called?

PreCreateWindow does not get called when you create a dialog box. If you would like to init some data/controls for a dialog box you have to trap the OnInitDialog message and do your stuff there. PreCreateWindow is used to modify params for a window that you are creating.

ewalker@tezcat.com, mfc-l, 7/12/95

6.5.7. How do I embed a common dialog in a property page?

This question comes up frequently on the "MFC" forum of CompuServe and the simple answer - unfortunately - is that there is no way to do it :-)

chris@chrism.demon.co.uk, programmer.win32, 7/12/95

6.5.8. Why can't I DDX/DDV to initialize my CDialog controls?

You can't do anything with the dialog controls until your dialog is created - which doesn't happen until DoModal(). The standard way of overcoming the problems is to create member variables for the data, initialize them before calling DoModal and then transfer the values in OnInitDialog. Or perhaps in UpdateData(). Much like the ClassWizard member variables does it.

So have your dialog include a CStringList or CStringArray, put the values for the listbox in that and transfer them to the listbox in OnInitDialog. [etc...]

null@diku.dk, programmer.controls, 7/11/95

Init your dialog in OnInitDialog. If necessary pass a pointer to your document to the constructor of your dialog (and save it in a private/protected m_pDoc member).

jhasling@gascad.co.at, programmer.controls, 7/11/95

6.5.9. How do I change the captions of a CPropertyPage?

You can change the label before adding the page to the property sheet in the following way. You have to derive a class from CPropertyPage and add a public function SetCaption which sets the caption.

```
void CPage1::SetCaption(char *str)
{
    m_strCaption = str; // m_strCaption is protected member of
                       //CPropertyPage
}
```

Now you can use the SetCaption() function in the following way.

```
CMySheet my("My PropSheet");
CPage1 p1;
p1.SetCaption(str); // Setting the caption
my.AddPage(&p1);
CAnotherSheet newps("New Sheet");
CPage1 p2;
p2.SetCaption(newstr);
newps.AddPage(&p2);
my.DoModal();
```

Ramesh, NetQuest., MSMFC 8/3/95

6.5.10. How do I trap F1 in my dialog?

The following Knowledge Base Article explains a way to trap the WM_KEYDOWN messages in the dialog box.

ID: Q117563, TITLE: How to Trap WM_KEYDOWN Messages in a CDialog

The next article explains how to provide context sensitive help in a dialog. It also points to sample code.

ID: Q110506, SAMPLE: Context Sensitive Help in a CDialog

Ramesh, NetQuest., MSMFC, 8/31/95

6.5.11. How do I change the icon for a dialog-only MFC application?

Add the following code to the InitInstance() for the CWinApp derived class:

```
BOOL CDialogTestApp::InitInstance()
{
    //...
    #if(_MFC_VER >= 0x0300)
        SetClassLong(m_pMainWnd->m_hWnd, GCL_HICON,
            (LONG)LoadIcon(IDC_ICONDIALOGAPP));
    #else
        SetClassWord(m_pMainWnd->m_hWnd, GCW_HICON,
            (WORD)LoadIcon(IDC_ICONDIALOGAPP));
    #endif
    //...
    m_pMainWnd->ShowWindow(m_nCmdShow);
    return TRUE;
}
```

NEW!! 6.5.12. How to I get rid of the Help and Apply buttons on my CPropertySheet-derivative?

In the constructor for your CPropertySheet-derivative add the following lines:

```
m_psh.dwFlags |= PSH_NOAPPLYNOW;
m_psh.dwFlags &= ~PSH_HASHELP;
```

In the constructor for every CPropertyPage-derivative included in the Property Sheet, add the following line:

```
m_psp.dwFlags &= ~PSP_HASHELP;
```

Eric Bergman-Terrell, erictterrell@juno.com, 5/16/97

NEW!! 6.5.13. How do you add a resize 'handle' to a dialog?

The way I would handle this is to draw the control using DrawFrameControl(...) function (CDC:: and Win32 API):

```
void CMyDialog::OnPaint( )
{
    CDialog::OnPaint();
    CRect rc;
    GetClientRect( rc );
    rc.left = rc.right - GetSystemMetrics(SM_CXHSCROLL);
    rc.top = rc.bottom - GetSystemMetrics(SM_CYVSCROLL);
    CClientDC dc( this );
    dc.DrawFrameControl( rc, DFC_SCROLL, DFCS_SCROLLSIZEGRIP );
}
```

Don't forget to add the code for WM_NCHITTEST to return HTBOTTOMRIGHT in the area of your sizegrip.

```
UINT CMyDialog::OnNcHitTest(CPoint point)
{
    UINT ht = CDialog::OnNcHitTest(point);
    if( ht == HTCLIENT )
    {
        CRect rc;
        GetWindowRect( rc );
        rc.left = rc.right - GetSystemMetrics(SM_CXHSCROLL);
        rc.top = rc.bottom - GetSystemMetrics(SM_CYVSCROLL);
        if( rc.PtInRect( point ) )
            ht = HTBOTTOMRIGHT;
    }
    return ht;
}
```

Glenn Carr, [gcarr@LGC.COM], mfc-l, 6/27/98

7. Toolbars, Status Bars, Control Bars, and Dialog Bars

- 7.1. [How do I add a combobox to my toolbar?](#)
 - 7.2. [How do I update the text of a pane in a status bar?](#)
 - 7.3. [How do I make my CToolBar customizable at run-time?](#)
 - 7.4. [How do I turn off the toolbar or status bar?](#)
 - 7.5. [How do I create a toolbar/statusbar in a dialog?](#)
 - 7.6. [Controls not support by MFC](#)
 - 7.6.1. [Why doesn't MFC support the new controls provided by IE?](#)
 - 7.6.2. [Why doesn't MFC have toolbars like the Office 97 application?](#)
 - 7.6.3. [Why doesn't MFC support command bar menus like the Visual C++ IDE?](#)
-

7.1. How do I add a combobox to my toolbar?

You can do this using the `CToolBar::SetButtonInfo()` method. The MFC sample `ctrlbars` shows how to do this in file `mainfrm.cpp`. Basically you call `SetButtonInfo` to create a blank space (`TBBS_SEPARATOR`) for the combo box, with the resource ID of the help and tool tip for the combobox. You then use `GetItemRect` to get the bounding rectangle of the combobox, and create a combobox window of your own in that space.

scot@stingray.com, 6/1/95 martynl@cix.compulink.co.uk - updated.

7.2. How do I update the text of a pane in a status bar?

By default, a `CStatusBar` pane is not enabled when the pane is created. To activate a pane, you must call the `ON_UPDATE_COMMAND_UI()` macro for each pane on the status bar and update the panes. Because panes do not send `WM_COMMAND` messages, you cannot use ClassWizard to activate panes; you must type the code manually. For example, suppose one pane has `ID_INDICATOR_PAGE` as its identifier and that it contains the current page number in a document. To make the `ID_INDICATOR_PAGE` pane display text, add the following to a header file (probably the `MAINFRM.H` file):

```
afx_msg void OnUpdatePage(CCmdUI *pCmdUI);
```

Add the following to the application message map:

```
ON_UPDATE_COMMAND_UI(ID_INDICATOR_PAGE, OnUpdatePage)
```

Add the following to a source code file (probably `MAINFRM.CPP`):

```
void CMainFrame::OnUpdatePage(CCmdUI *pCmdUI)
{
    pCmdUI->Enable();
}
```

To display text in the panes, either call `SetPaneText()` or call `CCmdUI::SetText()` in the `OnUpdate()` function. For example, you might want to set up an integer variable `m_nPage` that contains the current page number. Then, the `OnUpdatePage()` function might read as follows:

```
void CMainFrame::OnUpdatePage(CCmdUI *pCmdUI)
{
    pCmdUI->Enable();
    char szPage[16];
    wsprintf((LPSTR)szPage, "Page %d", m_nPage);
    pCmdUI->SetText((LPSTR)szPage);
}
```

This technique causes the page number to appear in the pane during idle processing in the same manner that the application updates other indicators.

MSVC Knowledge Base 6/4/94

7.3. How do I make my CToolBar customizable at run-time?

You might consider reading article "CToolBarCtrl :Handling Customization Notifications" in the Product Documentation of VC++ 2.1.

Here is the relevant extract :

"A Windows toolbar common control has built-in customization features, including a system-defined customization dialog box, which allow the user to insert, delete, or rearrange toolbar buttons. The application determines whether the customization features are available and controls the extent to which the user can customize the toolbar. These customization features are available in the CToolBarCtrl class but not in the current CToolBar class.

You can make these customization features available to the user by giving the toolbar the CCS_ADJUSTABLE style. The customization features allow the user to drag a button to a new position or to remove a button by dragging it off the toolbar. In addition, the user can double-click the toolbar to display the Customize Toolbar dialog box, which allows the user to add, delete, and rearrange toolbar buttons. The application can display the dialog box by using the Customize member function."

R.Rajendran (NetQuest), 76041.2245@compuserve.com,

MSMFC Forum, May-9-95

If you want to make a standard MFC CToolBar customizable, you can download CUSBAR.ZIP from the MSMFC library on CompuServe. This package implements CCustomTolbar, the run-time customizable toolbar and also provides the necessary user tools (customization dialog box including the code for a bitmapped listbox). Freeware.

Patrick Philippot, 8/3/95 via email on CSERVE

7.4. How do I turn off the toolbar or status bar?

You can turn the status bar off in any of your views (i.e. in the OnViewStatusBar() method you describe above) with the following code:

```
if( ((CMainFrame*)GetParent()->m_wndToolBar.IsWindowVisible() )
{
    GetParent()->SendMessage(WM_COMMAND, ID_VIEW_TOOLBAR, 0L);
}
if( ((CMainFrame*)GetParent()->m_wndStatusBar.IsWindowVisible() )
{
    GetParent()->SendMessage(WM_COMMAND, ID_VIEW_STATUS_BAR, 0L);
}
```

Use 1L instead of 0L for the SendMessage's lParam to turn the bars on.

JKBenjamin@aol.com via mfc-l, 5/16/95

7.5. How do I create a toolbar/statusbar in a dialog?

See [section 6.5.5](#) of this FAQ

7.6. Controls not support by MFC

7.6.1. Why doesn't MFC support the new controls provided by IE?

7.6.2. Why doesn't MFC have toolbars like the Office 97 applications?

7.6.3. Why doesn't MFC support command bar menus like the Visual C++ IDE?

A: The new user interface style provided by IE is implemented by a version of COMCTL32.DLL that's still in beta. The beta import library, DLL, and documentation is available in the ActiveX 3.01 SDK and is available for download from Microsoft's web site. Since the controls are still in beta, MFC doesn't support them. When they're available in their final version, you can assume that a version of MFC that supports them will be forthcoming.

The IDE and the Office applications do not use the COMCTL32.DLL

implementations of those controls, and therefore aren't concerned with the release state of the system implementation library.

[-B ekiM - mikeblas@microsoft.com.](mailto:MikeBlas@microsoft.com)

Note: the IE control is called Rebar and there was a very good MSJ article about it. It is useful ONLY if you want to implement IE3/4 style toolbars. The toolbars in VC++ 5.x and Office 97 are different - they are called Command bars. THEY ARE NOT REBARS - the differences?

- Rebars "slide" around.
- Rebars aren't customizable (to my knowledge)
- Command bars have cool floating menus too (rebar may do this eventually)
- Command bars are less clunky.
- Command bars don't have a transparent background.
- Command bars are implemented inside of Office97/DevStudio - they are

not available for developers to use, Rebar will be as part of a common control DLL.

- Both have a "cool" or "flat" look and feel where the button reacts to the mouse movement by displaying a border.
- I don't think Rebars are vertically dockable.

If you want to play with rebar you need the files COMMCTRL.H and COMCTL32.LIB that comes with the ActiveX SDK available at

<http://www.microsoft.com/intedev/sdk/>.

Also, for more information take a look at the REBAR-Sample available at <http://www.microsoft.com/win32dev/ui/rebar.htm>

FYI, Stingray Software has implemented Office97/DevStudio style command bars in Objective Toolkit - check out the demo at <http://www.stingray.com>.

8. Documents, Views and Frame Class Questions

8.1. [Views](#)

- 8.1.1. [How do I size a view?](#)
- 8.1.2. [How do I size a CFormView?](#)
- 8.1.3. [How do I use new views with a doc template?](#)
- 8.1.4. [How do I change the background color of a view?](#)
- 8.1.5. [How do I get the current View?](#)
- 8.1.6. [How do I create multiple views on one document?](#)
- 8.1.7. [How do I get all the views in an MDI app?](#)
- 8.1.8. [How do I make a CScrollView "mouse scrollable"?](#)

NEW!! 8.1.9. [How do I limit the maximum and minimum sizing of a view?](#)

8.2. [Documents](#)

- 8.2.1. [Do I have to use the Document/View architecture?](#)
 - 8.2.2. [How do I get the current Document?](#)
 - 8.2.3. [When are documents destroyed?](#)
 - 8.2.4. [How do I create multiple documents?](#)
 - 8.2.5. [How do I get a list of open documents?](#)
 - 8.2.6. [How do I keep my application from creating a new document at startup?](#)
-

8.1. Views

8.1.1. How do I size a view?

Normally, you can change the size of a window by calling `MoveWindow()`. In an application developed with the Microsoft Foundation Class (MFC) Library, the view window is a child window of the frame window that surrounds the view. To change the size of the view window, retrieve a pointer to the frame window of the view by calling `GetParentFrame()`, then call `MoveWindow()` to change the size of the parent. When the parent frame window changes size, it automatically changes the size of the view window to fit in the parent frame.

MSVC Knowledge Base. 6/4/94

8.1.2. How do I size a CFormView?

See MS Knowledge Base article Q98598 for a very long answer. Basically, you need to override `OnInitialUpdate()` in a `CFormView` derived class. There's other details to deriving from `CFormView` that the article goes into.

MSVC Knowledge Base. 6/7/95

In the view `ClikethisView` declaration:

```
virtual void OnInitialUpdate();
```

In the `ClikethisView` code:

```
void ClikethisView::OnInitialUpdate()
{
    // make the window the size of the main dialog
    CFormView::OnInitialUpdate();
    GetParentFrame()->RecalcLayout();
    ResizeParentToFit( /*FALSE*/ );
}
```

andyr@gate.net, programmer.misc, 8/11/95

8.1.3. How do I use new views with a doc template?

In an application created with AppWizard, you have two options: change the derivation of the current view, or create a new view and use the new view in your MDI application along with the original view.

To create a new view, use ClassWizard to create a new class derived from CView. After the class has been created, the steps to use the new view or to modify the view provided by App Wizard are the same.

1. Modify the header file for the view class to change all references to CView to the name of the desired view class. In this example, the class is derived from CScrollView. Usually, this step involves changing the class the view class is derived from as follows:

```
class CMyView : public CScrollView
```

2. Modify the implementation file for the view class to change all references to CView to the name of the desired view class. This involves changing the IMPLEMENT_DYNCREATE line as follows:

```
IMPLEMENT_DYNCREATE(CMyView, CScrollView)
```

changing the BEGIN_MESSAGE_MAP as follows:

```
BEGIN_MESSAGE_MAP(CMyView, CScrollView)
```

and changing any other references to CView to CScrollView.

3. No further modifications are required if you are modifying a view created by App Wizard. If you create a new view, find the AddDocTemplate() call in the CWinApp::InitInstance() function. The third parameter to AddDocTemplate() is RUNTIME_CLASS(CSomeView). To replace the current view with the new view class, change CSomeView to CMyView. In an MDI application, you can use multiple view types by adding a second AddDocTemplate() call that changes RUNTIME_CLASS(CSomeView) to RUNTIME_CLASS(CMyView).

For more information, please see Knowledge Base article Q99562.

MSVC Knowledge Base 6/7/95

8.1.4. How do I change the background color of a view?

To change the background color for a CView, CFrameWnd, or CWnd object, process the WM_ERASEBKGD message. The following code shows how:

```
BOOL CSampleView::OnEraseBkgnd(CDC* pDC)
{
    // Set brush to desired background color
    CBrush backBrush(RGB(255, 128, 128));

    // Save old brush
    CBrush* pOldBrush = pDC->SelectObject(&backBrush);
    CRect rect;
    pDC->GetClipBox(&rect);

    // Erase the area needed
    pDC->PatBlt(rect.left, rect.top, rect.Width(), rect.Height(), PATCOPY);
    pDC->SelectObject(pOldBrush);

    return TRUE;
}
```

I solved the problem like this->

```
HBRUSH dlgtest::OnCtlColor(CDC* pDC, CWnd* pWnd, UINT nCtlColor)
{
    switch (nCtlColor)
    {
        case CTLCOLOR_BTN:
        case CTLCOLOR_STATIC:
        {
            pDC->SetBkMode(TRANSPARENT);
        }
        case CTLCOLOR_DLG:
        {
            CBrush* back_brush;
            COLORREF color;
            color = (COLORREF) GetSysColor(COLOR_BTNFACE);
            back_brush = new CBrush(color);
            return (HBRUSH) (back_brush->m_hObject);
        }
    }
    return(CFormView::OnCtlColor(pDC, pWnd, nCtlColor));
}
```

Tim, tfiner@vrli.com, email, 9/10/95

8.1.5. How do I get the current View?

The best thing to do is to pass the view along as a parameter. If this is impractical, you can get the view if you KNOW, that it is the currently active document and the currently active view. For details, see :

Microsoft KB, article Q108587, "Get Current CDocument or CView from Anywhere".

In brief, use

```
((CFrameWnd*) AfxGetApp()->m_pMainWnd)->GetActiveDocument()
```

and

```
((CFrameWnd*)(AfxGetApp()->m_pMainWnd))->GetActiveView()
```

to get the document and the view. It might be a good idea to wrap them in static functions in your CMyDoc and CMyView and check that they are of the correct RUNTIME_CLASS.

If the view isn't the currently active view or if you can run OLE in-place, this won't work however.

null@diku.dk, Niels Ull Jacobsen, programmer.misc, 6/8/95

8.1.6. How do I create multiple views on one document?

The CDocTemplate::CreateNewFrame() function creates additional views of a document in an MDI application written with MFC. To call this function, specify a pointer to a CDocument object (the document for which the function will create a view) and a pointer to a frame window that has the properties to duplicate. Typically, the second parameter of this function is NULL.

When an application calls CreateNewFrame(), the function creates a new frame window and a view in the frame window. The frame window type and view type depend on the document template (CDocTemplate) associated with the document specified in the CreateNewFrame() call.

The CHKBOOK MFC sample application that ships with Visual C++ also demonstrates creating additional frames and views for documents. Check out CHKBOOK.CPP, the CChkBookApp::OpenDocumentfile() function.

Another example of using `CreateNewFrame()` is the MULTVIEW sample application. Also, Dale Rogerson's article, "Multiple Views for a Single Document" located on the Microsoft Developer Network Development Library CD-ROM, which explains in detail how to add additional views to an existing document, is an excellent source of information.

`CreateNewFrame()` creates both a frame and a view; not only a view. If, for some reason, `CreateNewFrame()` does not quite address your situation, the source code for `CreateNewFrame()` is quite useful to demonstrate the steps required to create frames and views.

MS FAQ with mods, 6/25/95

8.1.7. How do I get all the views in an MDI app?

You need to use some functions which are undocumented:

- `CDocument::GetFirstViewPosition(); // DOCCORE.CPP`
- `CDocument::GetNextView(); // DOCCORE.CPP`
- `CMultiDocTemplate::GetFirstDocPosition(); // DOCMULTI.CPP`
- `CMultiDocTemplate::GetNextDoc(); // DOCMULTI.CPP`

You'll also need to mess with the `m_templateList` member of `CWinApp`.

blaszczak@Bix.com, mfc-1, 7/11/95

Note: This has changed with MFC 4.0. There's now a class called `CDocManager` which can get to all of the views/docs for you. Check out [MFC Internals](#) for details.

scot@stingray.com

8.1.8. How do I make a CScrollView "mouse scrollable"?

Download AUTOSV.LZH from the MSMFC library on CIS. This code shows you how to implement a secondary message loop taking care of the mouse activity. Hooks are provided to customize the code. Freeware.

Patrick Philippot, CIS email, 8/3/95

NEW!! 8.1.9. How do I limit the maximum and minimum sizing of a view?

To limit the maximum and minimum size of a view, you need to do two things:

1. Handle the WM_GETMINMAXINFO message in a CFrameWnd derived class. This is because when the user resizes a view, they are actually grabbing the frame of the view. The MINMAXINFO struct sets limits on the entire window, so remember to take into account toolbars, scroll bars, etc.

```
// Min and Mix size in pixels - values choosen are just examples
#define MINX 200
#define MINY 300
#define MAXX 300
#define MAXY 400

void CMyFrameWnd::OnGetMinMaxInfo(MINMAXINFO FAR* lpMMI)
{
    CRect rectWindow;
    GetWindowRect(&rectWindow);

    CRect rectClient;
    GetClientRect(&rectClient);

    // get offset of toolbars, scrollbars, etc.
    int nWidthOffset = rectWindow.Width() - rectClient.Width();
    int nHeightOffset = rectWindow.Height() - rectClient.Height();

    lpMMI->ptMinTrackSize.x = MINX + nWidthOffset;
    lpMMI->ptMinTrackSize.y = MINY + nHeightOffset;
    lpMMI->ptMaxTrackSize.x = MAXX + nWidthOffset;
    lpMMI->ptMaxTrackSize.y = MAXY + nHeightOffset;
}
```

2. Also, be sure to remove the WS_MAXIMIZEBOX in your CFrameWnd derivative's PreCreateWindow function. If you don't, when the view is maximized, the frame will not behave as you'd expect when maximized. To understand what I'm getting at, try leaving the WM_MAXIMIZEBOX flag and see what happens when you maximize the view.

```
BOOL CMyFrameWnd::PreCreateWindow(CREATESTRUCT& cs)
{
    cs.style &= ~WS_MAXIMIZEBOX;
    return CFrameWnd::PreCreateWindow(cs);
}
```

}



8.2. Documents

8.2.1. Do I have to use the Document/View architecture?

MFC does not force you to use document/views. Check out hello, mdi, and helloapp samples, they don't use it at all. Most MFC features can be used in non-document/view applications. You do lose features like print preview and of many OLE features when you don't go document/view.

scot@stingray.com 6/7/95

8.2.2. How do I get the current Document?

See section [8.1.5](#) for the details.

8.2.3. When are documents destroyed?

In SDI applications, the document is deleted when the application exits. In MDI applications, the document is deleted when the last view on the document is closed. To help keep your document SDI/MDI compatible, you should delete the document's data in the virtual DeleteContents() function, not in the destructor.

Richard Hazenberg, drmtime@euronet.nl, programmer.misc, 6/24/95

8.2.4. How do I create multiple documents?

To add support for additional document types, you can create and register additional CMultiDocTemplate objects with your CWinApp derived object. This technique is illustrated in the MULTDOCS sample application. The general steps needed to add an additional document type to an MFC application are listed below:

1. Use AppWizard to create a new document class and a new view class.
2. Use the Resource Editor to add a new resource string to support the new document class. For more information on the format of the document template string resource, see the topic How to Interpret a Document

Template String.

3. Use the Resource Editor to add an additional icon and menu resource to the application. Note, the ID for each of these resources needs to be the same ID as the resource ID used for the document template string created in step 2. This ID is used by the CMultiDocTemplate class to identify the resources associated with the additional document type.
4. In the applications InitInstance() function, create another CMultiDocTemplate object and register it with the CWinApp::AddDocTemplate() function. For example:

```
CMultiDocTemplate* pDocTemplate2 = new CMultiDocTemplate( IDR_DOC2TYPE,  
RUNTIME_CLASS(CDoc2), RUNTIME_CLASS(CMDIChildWnd),  
RUNTIME_CLASS(CView2)); AddDocTemplate(pDocTemplate2);
```

5. And finally, add the custom serialization and painting code to your new document and view classes.

MS FAQ, 6/25/95

8.2.5. How do I get a list of open documents?

The code below demonstrates how to retrieve a list of pointers to all CDocuments that were created using a CDocTemplate object.

In the code below, CMyApp is derived from CWinApp. The variable m_templateList is a CPtrList object that is a member of CWinApp, and it contains a list of pointers to all of the document templates (CDocTemplates). The CDocTemplate functions GetFirstDocPosition() and GetNextDoc() are used to iterate through the list of documents for each document template.

```
void CMyApp::GetDocumentList(CObList * pDocList)  
{  
    ASSERT(pDocList->IsEmpty());  
    POSITION pos = m_templateList.GetHeadPosition();  
    while (pos)  
    {  
        CDocTemplate* pTemplate = (CDocTemplate*)m_templateList.GetNext(pos);  
        POSITION pos2 = pTemplate->GetFirstDocPosition();  
        while (pos2)  
        {  
            CDocument* pDocument;  
            if ((pDocument=pTemplate->GetNextDoc(pos2)) != NULL)  
                pDocList->AddHead(pDocument);  
        }  
    }  
}
```

```
}  
}
```

There are two public member functions of the CDocTemplate class that are not documented in the reference manual or the online help. However, these are public member functions defined in the CDocTemplate class and provide simple functionality for traversing the list of open documents. These functions operate as follows:

Function: virtual POSITION GetFirstDocPosition() const;

Remarks: Call this function to get the position of the first document in the list of open documents associated with the template.

Return Value: A POSITION value that can be used for iteration with the GetNextDoc member function.

Function: virtual CDocument* GetNextDoc(POSITION& rPosition) const;

rPosition: A reference to a POSITION value returned by a previous call to the GetNextDoc or GetFirstDocPosition member function. This value must not be NULL.

Remarks: Call this function to iterate through all of the document template's open documents. The function returns the document identified by rPosition and then sets rPosition to the POSITION value of the next document in the list. If the retrieved document is the last in the list, then rPosition is set to NULL.

Return Value: A pointer to the view identified by rPosition.

MS FAQ, 6/25/95

NOTE: This is only valid for MFC 3.2 and lower, MFC 4.0 version follows->

```
void CMyApp::DoSomethingToAllDocs()  
{  
    CObList pDocList;  
    POSITION pos = GetFirstDocTemplatePosition();  
    while(pos)  
    {  
        CDocTemplate* pTemplate = GetNextDocTemplate(pos);  
        POSITION pos2 = pTemplate->GetFirstDocPosition();  
        while(pos2)  
        {  
            CDocument* pDocument;  
            if(pDocument = pTemplate->GetNextDoc(pos2))  
                pDocList.AddHead(pDocument);  
        }  
    }  
}
```

```
if(!pDocList.IsEmpty())
{
    pos = pDocList.GetHeadPosition();
    while(pos)
    {
        // Call some CDocument function for each document
        ((CDocument*)pDocList.GetNext(pos))->UpdateAllViews(NULL);
    }
}
```

mcontest@universal.com, email, 9/22/95

8.2.6. How do I keep my application from creating a new document at startup?

Add this call:

```
cmdInfo.m_nShellCommand = CCommandLineInfo::FileNothing
```

just before the call to ProcessShellCommand in the app's InitInstance.

lechner-cos1@kaman.com, email, 1/6/96

9. Menus and Message Handling

9.1. [Menus](#)

- 9.1.1. [How do I get a pointer to the menu bar in a MDI app?](#)
- 9.1.2. [How do I implement a right-mouse popup-menu?](#)
- 9.1.3. [How do I dynamically change the mainframe menu?](#)
- 9.1.4. [How do I 'attach' a menu to a window's creation/destruction?](#)
- 9.1.5. [How do I add "What's This" menus to my application - like Win95 hip apps have?](#)

9.2. [Message Handling](#)

- 9.2.1. [How do I intercept WM_SETTEXT messages?](#)
 - 9.2.2. [How do I handle my own registered messages?](#)
 - NEW!!** 9.2.3. [Is it possible to process a range of messages without creating a message-map entry for each message in the range?](#)
 - NEW!!** 9.2.4. [Is it possible to enable/disable a range of menu items without creating a message-map entry for each message in the range?](#)
 - NEW!!** 9.2.5. [How do I capture a windows message from any window fired?](#)
-

9.1. Menus

9.1.1. How do I get a pointer to the menu bar in a MDI app?

QUESTION:

I'm writing a MDI application and I have problems to get a pointer to the actual menu bar. The normal construction doesn't seem to work in MDI:

```
CMenu *menu;  
menu = GetMenu()->GetSubMenu(0);
```

How can I get a pointer to the menu bar to update the menu?**ANSWER:**

```
AfxGetApp()->m_pMainWnd->GetMenu()->GetSubMenu(n);
```

mlinar@pollux.usc.edu, Mitch Mlinar, 6/8/95

9.1.2. How do I implement a right-mouse popup-menu?

```
////////////////////////////////////  
// WM_RBUTTONDOWN handler.  
//  
// Trap this message and display the button properties popup menu.  
// The main frame receives the popup menu messages. This allows the  
// status bar to be updated with the help text.  
// //////////////////////////////////////  
void CAppButton::OnRButtonDown(UINT flags, CPoint point)  
{  
    CMenu menu;  
    CMenu *submenu;  
    // load the menu  
    menu.LoadMenu(IDR_LAUNCH);  
    // get the popup menu  
    submenu = menu.GetSubMenu(0);  
    // convert to screen coordinates  
    ClientToScreen(&point);  
    // post the menu  
    submenu->TrackPopupMenu(TPM_LEFTALIGN | TPM_RIGHTBUTTON,  
        point.x, point.y,  
        AfxGetApp()->m_pMainWnd, NULL);  
}
```

johnm@unipalm.co.uk, programmer.win32, 7/12/95

It's better to use RBUTTONUP instead, however right-clicking on dialog controls doesn't generate RBUTTONUP and RBUTTONDOWN messages.

If it's necessary to treat this situation too, a program has to catch WM_PARENTNOTIFY message in Win3.x and WinNT and WM_CONTEXTMENU in Windows 95. Here's a code:

```
// May be dialog too:
BEGIN_MESSAGE_MAP(CMyPropertyPage, CPropertyPage)
   //{{AFX_MSG_MAP(CMyPropertyPage)
    ON_WM_RBUTTONUP()
    ON_WM_PARENTNOTIFY()
    ON_MESSAGE(WM_CONTEXTMENU, OnContextMenu)
    //}}AFX_MSG_MAP
END_MESSAGE_MAP()

void CMyPropertyPage::OnRButtonUp(UINT nFlags, CPoint point)=20
{
    PopupMenu (&point);
}

void CMyPropertyPage::OnParentNotify(UINT message, LPARAM lParam)
{
    if (message != WM_RBUTTONDOWN)
        CPropertyPage::OnParentNotify(message, lParam);
    else
    {
        CPoint pt(LOWORD(lParam),HIWORD(lParam));
        PopupMenu (&pt);
    }
}

LONG CMyPropertyPage::OnContextMenu (UINT wParam, LONG lParam)
{
    CPoint pt(LOWORD(lParam),HIWORD(lParam));
    ScreenToClient (&pt);
    PopupMenu (&pt);
    return 0;
}

//*****

void CMyPropertyPage::PopupMenu(CPoint* pt)
{
    ASSERT(m_idContextMenu != 0);
    ASSERT(nSubMenu >= 0);
    ClientToScreen (pt);
    CMenu FloatingMenu;
    VERIFY(FloatingMenu.LoadMenu(ID_POPUP_MENU));
    CMenu* pPopupMenu = FloatingMenu.GetSubMenu (0);
}
```

```

    ASSERT(pPopupMenu != 3D NULL);
    pPopupMenu->TrackPopupMenu (TPM_LEFTALIGN | TPM_RIGHTBUTTON,=20
        pt->x, pt->y, this);
}

```

PaulACost@msn.com - via email, 10/15/95

9.1.3. How do I dynamically change the mainframe menu?

```

CMenu newMenu;
newMenu.LoadMenu (IDR_MENU1);
AfxGetMainWnd()->SetMenu( &newMenu );
AfxGetMainWnd()->DrawMenuBar();
newMenu.Detach ();

```

Arun Rao, MSMFC, 6/27/95

9.1.4. How do I 'attach' a menu to a window's creation/destruction?

{Note the original question talked about dialogs, but you can interpolate this code to any kind of window that you want to have change the menu.}

One of the ways to do this is as follows.....

Declare a variable CMenu pNewMenu in one of the dialog class.
Handle the WM_INITDIALOG and WM_CLOSE messages in the dialog class as follows.

```

BOOL CMydlg::OnInitDialog()
{
    CDialog::OnInitDialog();
    // Load the IDR_MYFRAME menu
    pNewMenu = new CMenu;
    pNewMenu->LoadMenu(IDR_MYFRAME);
    // Set the mainframe menu to mainframe.
    ((CMainFrame *)AfxGetMainWnd()->SetMenu(pNewMenu);
    return TRUE;
}

```

And

```

void CMydlg::OnClose()
{
    // Detach the previous HMenu handle from the object.
}

```



```

pNewMenu->Detach();
pNewMenu->LoadMenu(IDR_MAINFRAME);
// Restore the mainframe menu.
((CMainFrame *)AfxGetMainWnd())->SetMenu(pNewMenu);
CDialog::OnClose();
}

```

If there are other methods of closing the dialog (example- By clicking a button in the Dialog), then The code given above in OnClose handler, must be put in the button click handler.

Sanjeev Kumar, MSMFC, 6/23/95

9.1.5. How do I add 'What's this' menus to my application - like Win95 hip apps have?

Here's some steps to get you started->

1. Put the following menu into a resource script:

```

IDR_WHAT_IS_THIS_MENU  MENU  DISCARDABLE
BEGIN
  BEGIN
    POPUP "a"
    BEGIN
      MENUITEM "What's this?", ID_WHAT_IS_THIS
    END
  END
END
END

```

2. Add to your dialog a right-click handler (OnRButtonDown) with menu IDR_WHAT_IS_THIS_MENU. You need to store the point of the last click in some variable - for example

```
CPoint m_cLastRClickPoint;
```

and store here the client coordinates of the last right click.

3. Put the following code into your dialog class (or probably the parent class of all your dialogs):

```

BEGIN_MESSAGE_MAP(CMyDialog, CDialog)
 //{{AFX_MSG_MAP(CMyDialog)
  // whatever

```

```
//}}
ON_COMMAND(ID_WHAT_IS_THIS, OnWhatIsThis)
END_MESSAGE_MAP()

void CMyDialog::OnWhatIsThis()
{
    CWnd* pControl = ChildWindowFromPoint (m_cLastRClickPoint);

    // if the click wasn't on one of the controls - open help for dialog
    if (pControl == NULL || pControl->m_hWnd == m_hWnd)
        WinHelp (HID_BASE_RESOURCE + m_nIDHelp, HELP_CONTEXTPOPUP);
    else
        WinHelp (HID_BASE_CONTROL + pControl->GetDlgCtrlID(), HELP_CONTEXTPOPUP);
}
```

- and finally add the following lines to the makehelp.bat file:

```
echo. >>hlp\wr.hm
echo // Controls (IDC_*) >>hlp\wr.hm
makehm IDC_,HIDC_,0x50000 resource.h >>hlp\wr.hm
```

This wires everything to your help system.

Poul A. Costinsky (PoulACost@msn.com).

9.2. Message Handling

9.2.1. How do I intercept WM_SETTEXT messages?

Because MFC didn't provide generic support for overriding WM_SETTEXT, you can use OnMessage(WM_SETTEXT, OnSetText) in the message map and then define your own method:

```
LRESULT CMyClass::OnSetText(wParam, lParam); //
```

jfriend@collabra.com, programmer.tools, 8/17/95

9.2.2. How do I handle my own registered messages?

[Editor Note: In this FAQ, Dean is telling a guy how to handle WM_CHKTBLTOGGLE, which is some message that dude is trying to handle. The guy was doing it a wrong way before, I've left that in for educational reasons.]

Use ON_MESSAGE:

```
ON_MESSAGE(WM_CHKTBLTOGGLE, OnChkTblToggle)
```

In your class definition:

```
afx_msg LRESULT OnChkTblToggle(WPARAM wParam, LPARAM lParam);
```

In your message map:

```
#define ON_WM_CHKTBLTOGGLE()  
{  
    WM_CHKTBLTOGGLE, 0, AfxSig_vwp, (AFX_PMSG)(AFX_PMSGW)(BOOL  
    (AFX_MSG_CALL CWnd::*)(BYTE, BYTE))OnChkTblToggle  
}
```

In your code:

```
LRESULT CMyView::OnChkTblToggle(WPARAM wParam, LPARAM lParam)  
{  
    // TODO: write your code here
```

```
}
```

You told MFC that your function is:

```
void CMYView::OnChkTbIToggle(UINT, CPoint)
```

That's what the signature `AfxSig_vwp` means... and definitely not what you want.

`ON_MESSAGE` and `ON_REGISTERED_MESSAGE` are intended to allow you to extend the message handlers to your own custom message handlers. Please don't rely on specific `AfxSig_*` values or on the message map structure -- it may change without notice.

Dean McCrory, mfc-l, 8/19/95

NEW!! 9.2.3. Is it possible to process a range of messages without creating a message-map entry for each message in the range?

NEW!! 9.2.4. Is it possible to enable/disable a range of menu items without creating a message-map entry for each message in the range?

Yes and yes, respectively. See the article "Message Map: Ranges of Messages" in the VC++ on-line help. This article explains how to use the following message-map macros:

```
ON_COMMAND_RANGE  
ON_UPDATE_COMMAND_RANGE  
ON_CONTROL_RANGE
```

Eric Bergman-Terrell, ericterrell@juno.com, 5/16/97

NEW!! 9.2.5. How do I capture a windows message from any window fired?

For windows belonging to the calling thread, use `GetMessage()`.
For windows outside the calling thread, use `SetWindowsHookEx()`.

Darren Ford [darren@cybrtyme.com], VCPP mailing list, 6/17/98

10. OLE Class Questions

10.1. [Structured Storage / Compound Files](#)

10.1.1. [When I upgraded, I could not read my files generated by the old version of MFC?](#)

10.2. [OLE Controls \(Previously called OCX's\)](#)

10.2.1. [What is an OLE control?](#)

10.2.2. [How do I write OLE controls?](#)

10.2.3. [What versions of MFC support OLE control containment?](#)

10.2.4. [How do I get application specific control bars to disappear when in-placed editing.](#)

10.1. Structured Storage/Compound Files

10.1.1. When I upgraded, I could not read my files generated by the old version of MFC. What do I do?

There may be an easier way, but here is how I did it:

1. Use `VERSIONABLE_SCHEMA` and `GetObjectSchema` to identify the version. If old version, then it is stored in OLE 1.0 format. To read this format, first read a `WORD` and a `CString` (type and name info). Next is the OLE data itself (the hard part). Use the following steps:
2. Use `StgCreateDocfile` to create a temporary Compound File.
3. Use `OleConvertOLESTREAMToIStorage` to copy the data to the Compound file, converting to OLE 2 format.
4. Call `OleLoad` using the `IStorage` to get an `IUnknown` pointer, then call `QueryInterface` to set the `COleClientItem::m_lpObject` member, and set `m_nDrawAspect` to `DVASPECT_CONTENT`.
5. Release the `IUnknown` pointer and the `IStorage` pointer
6. The `OleConvertOLESTREAMToIStorage` code requires an `OLESTREAM` implementation. Copy the code from MSVC 1.0 MFC code.

This actually works. There are some issues with messing with the schema parameter, and I have not implemented writing OLE 1/MFC files. Code available on request.

ronjones@xnet.com, Ron Jones, via programmer.tools, 5/9/95

10.2. OLE Controls (previously called OCX's)

10.2.1. What is an OLE control?

OLE controls are the 32-bit successor to 16-bit VBX controls. Instead of being stored in a plain DLL and having functional interfaces, OLE controls rely on OLE automation. Hopefully this will make the interface more flexible and easier to use than VBXs.

Although OLE controls use OLE, they are not object oriented. There is a flat set of properties you have access to and you can not apply OO techniques such as inheritance, polymorphism, etc.. to them. In my opinion, this can be pretty frustrating to the MFC programmer who is used to the OO techniques.

scot@stingray.com, 6/25/95

10.2.2. How do I write OLE controls?

In VC++ 2.x, Microsoft released the CDK (OLE control developer kit), you use that kit and it's tools to write OLE controls. In VC++ 4.0, this is no longer separate and you now just run the Control Wizard. Check your VC++ documentation for more info.

scot@stingray.com, 6/25/95

10.2.3. What versions of MFC support OLE control containment?

MFC 4.0 supports OLE control containment. Earlier versions of MFC don't unless you roll your own. (yuck!)

10.2.4. How do I get application specific control bars to disappear when in-placed editing.

Add `CBRS_HIDE_INPLACE` to `Create()` of the toolbar.

Example:

m_wndFormBar.Create(this, WS_CHILD | WS_VISIBLE | CBRS_TOP | CBRS_HIDE_INPLACE,
IDW_FORMBAR)

Paul Rony, 102615.601@compuserve.com

11. WOSA Class Questions

11.1. [CRecordset](#)

11.1.1. [When I add a CRecordset class, I get tons of linker errors..](#)

11.1.2. [I need a CRecordset::Find\(\), what should I do?](#)

11.2. [WinSock](#)

11.2.1. [I'm having problems with CSocket blocking, what's up?](#)

11.1. CRecordset

11.1.1. When I add a CRecordset class, I get tons of linker errors..

If you haven't told the AppWizard to use Database Support when you created your project, the AppWizard doesn't include the database headers. If you add "#include " in your stdafx.h, CRecordset will be "legalized". Then open the Linker Options window and add the library: "odbc". This avoids tons of linker errors.

bessler@highland.swb.de (Wolfgang Bessler) 06/05/95, comp.lang.c++

11.1.2. I need a CRecordset::Find(), what should I do?

Use the WHERE clause in SQL.

mikeblas@interserv.com - mfc-l, 10/10/95

11.2. WinSock

11.2.1. I'm having problems with CSocket blocking, what's up?

I've worked around this by creating a dialog box that I pop up after calling Connect()... i.e.:

```
CMySocket MySocket;  
MySocket.Create();  
MySocket.Connect("mysmtphost",25);  
DummyDlg DummyDialog;  
MySocket.m_pDialog=&DummyDialog;  
DummyDialog.DoModal();
```

This suspends the thread until MySocket sends a WM_CLOSE message to DummyDialog. I'd rather it look like:

```
MySocket.m_hEvent=CreateEvent(...);  
WaitForSingleObject(MySocket.m_hEvent,INFINITE);
```

But that stops all processing of MySocket.

Cynthia Jennings (idlewild@is.net), programmer.win32, 6/19/95

12. DLL and Build Questions

If you have questions about extension and user DLLs, be sure to read technical notes 11 and 33. Volume 2 of the documentation has more info too. Finally, try searching on the sample names dllhusk (extension DLL) and dlltrace (user DLL) in books on-line.

- 12.1. [Do I need a CWinApp object in a DLL?](#)
 - 12.2. [How should I define the WEP in a MFC DLL?](#)
 - 12.3. [How do I build an 'Extension DLL'?](#)
 - 12.4. [How can I manage resources in a 'Resource only DLL' and still benefit from ClassWizard?](#)
 - 12.5. [I am having \[some problem\] with exports/imports and extension DLLs.](#)
 - 12.6. [What DLL's should I distribute with my MFC app?](#)
-

12.1. Do I need a CWinApp object in a DLL?

The Microsoft Foundation Class Library supports two types of DLLs: `_USRDLL` and `_AFXDLL`. The `_USRDLL` model requires one `CWinApp` object to perform the initialization and cleanup of the Microsoft Foundation Class Library Windows classes that the DLL uses. This requirement is described in MFC Tech Note 11; the `DLLTRACE` sample demonstrates a `_USRDLL` that contains a `CWinApp` object.

An `_AFXDLL` does not require a `CWinApp` object. Because it shares the Microsoft Foundation Class Library classes with the application, it does not require a `CWinApp` to provide initialization and cleanup. Instead, an `_AFXDLL` requires a special version of `LibMain()` and a DLL initialization function.

MSVC Knowledge Base 6/4/95

12.2. How should I define the WEP in a MFC DLL?

In a dynamic-link library (DLL) built with Microsoft Foundation Class Library version 2.0, the _USRDLL model uses the WEP() (Windows exit procedure) function provided in the C run-time library. Because the code uses the C library WEP() function, the destructors for static and global objects in the DLL are called and the CWinApp::ExitInstance() function for the DLL application object is called.

See MSVC Knowledge Base article Q98374 and Tech notes 11 and 33.

MSVC Knowledge Base 6/7/95

12.3. How do I build an 'extension DLL'?

1. When you're building a 32-bit extension DLL, define `_AFXEXT` on the compiler command line. If you look in `AFXVER_.H`, you'll see that this forces `_AFXDLL` to also be defined. So an "AFXEXT" DLL is an AFXDLL.
2. When `_AFXDLL` is defined, `AfxGetResourceHandle` returns a value stored in MFC's global data, which is shared by the EXE, the extension DLL and the MFC DLL. The handle returned identifies the module which will be searched first when looking for a resource.

(See the source code for `AfxFindResourceHandle()` if you're curious about the order of the search.)

3. Strictly speaking, what we need to load a resource is a module handle rather than an instance handle. (Instances share modules --- e.g., code and resources --- but have different data.) A DLL has a module handle which is distinct from the handle of the EXE.
4. You can use `::GetModuleHandle` to get the handle for your DLL, then pass it to `AfxSetResourceHandle` so that your DLL is the first place searched for resources. But note that this removes the EXE module from of modules searched. You'll probably want to save a copy of the handle returned by `AfxGetResourceHandle` before calling `AfxSetResourceHandle`, then restore it once you're done loading the DLL resource.

Charlie Kester, Microsoft Developer Support, MSMFC, 7/19/95

12.4. How can I manage resources in a resource only DLL and still benefit from ClassWizard?

The following text is available as RESDLL.ZIP in the MSMFC library on CompuServe (applies to MSVC20):

How To Manage An MFC Project Storing Its Resources Into A Resource-Only DLL

Software localization is much easier when your project stores its resources in a resource-only DLL. There are also many situations where storing the project's resources in a DLL can be a good idea.

However, if this project is an MFC project, doing so will generate a major drawback: you will not benefit from the Class Wizard capabilities any longer because the resources will be managed in a separate project.

However, there's a trick that you can use to develop your project as if it were a standard project while being able to quickly switch to the resource-only DLL model. Here's how to proceed:

METHOD 1 :

1. Create your project as usual using AppWizard (we'll name it TEST).
2. Close the project and create a new DLL project in the same directory (call it RESDLL). When you click on the Create button, VC++ opens the Add file dialog. Take this opportunity to add the resource file of the previous project (TEST.RC) to this new project.
3. Before being able to compile the resources of the TEST project as a resource-only DLL, you must add the /NOENTRY option to the linker. Unfortunately, the settings dialog box of VC++ doesn't allow to do that in a simple way:
 1. Select Project|Settings from the main menu.
 2. Click on the Link tab.
 3. Select General from the category combobox.
 4. In the Object/Library Modules field, remove all references to any .LIB

file (they are useless) and add /NOENTRY. This option should then appear in the Common Options display area.

5. Click OK and compile. You now have a DLL containing only the resources for your project.
4. Do not open the TEST.MAK project. Instead, copy TEST.MAK to TEST_RES.MAK in your project directory.
5. Open TEST_RES.MAK and remove TEST.RC from the project files.
6. Select Project|Settings, click on the General tab and add "USE_RESDLL" to the list of Preprocessor Definitions.
7. Open TEST.H and modify the class declaration of CTestApp this way:

```
public:
    CTestApp();

#ifdef USE_RESDLL
public:
    virtual int ExitInstance();
private:
    HINSTANCE m_hInstDLL;
#endif //USE_RESDLL
```

8. Open TEST.CPP and modify CTestApp::InitInstance as follows. Also, add the newly declared ExitInstance member function:

```
BOOL CTestApp::InitInstance()
{
    // Standard initialization
    // If you are not using these features and wish to reduce the size
    // of your final executable, you should remove from the following
    // the specific initialization routines you do not need.
#ifdef USE_RESDLL
    if ((m_hInstDLL = LoadLibrary("resdll.dll")) == NULL)
    {
        return FALSE; // failed to load the localized resources
    }
    else
    {
        AfxSetResourceHandle(m_hInstDLL); // get resources from the DLL
    }
#endif
    //...#ifdef USE_RESDLL
    int CTestApp::ExitInstance()
    {
        FreeLibrary(m_hInstDLL);
        return CWinApp::ExitInstance();
    }
}
```

#endif

9. Compile. TEST_RES.EXE should work very nicely, loading its resources from the DLL.
10. Close the project and open TEST.MAK. Compile. TEST.EXE should also work very well but this time, the resources are loaded from the .EXE file because you had not defined USE_RES_DLL in this version of the project.

BEWARE: When switching from one model to another, you must either Rebuild All or touch TEST.CPP. Otherwise, you'll have problems.

Now, you are able to modify your resources and to use Class Wizard when you work with TEST.MAK. To build the resource-only DLL version of the project, just switch to TEST_RES.MAK after recompiling RESDLL (RESDLL.MAK) if changes have been made to the resources.

It's that easy!

METHOD 2 :

1. Proceed like above for steps 1, 2 and 3
2. Instead of creating a new project, copy TEST.RC to RESDLL.RC. Remove TEST.RC from your project and add RESDLL.RC. A different name is somewhat safer.
3. Double-click RESDLL.RC to trigger a rebuild of the .CLW file.
4. Apply the "Exclude File from Build" command to RESDLL.RC. (Project|Settings, General Page, "Exclude File From Build")
5. Delete TEST.CLW and TEST.RC.

Now TEST and RESDLL will work together through the ClassWizard. The resources defined in RESDLL.RC will not be added to TEST.EXE. This method is simpler but doesn't allow you to test your program in both cases (resources in the DLL or resources linked to the .EXE file). I prefer the latter when developing because I do not have to bother about orphaned DLLs when a GPF occurs (although this is less a problem under NT).

Don't forget to recompile the DLL before testing your program each time you modify the resources. A batch file calling NMAKE installed in the Tools menu will certainly be faster than switching from project to project. Using the "New Target" facility won't help in that case.

It would be nice if VC++ had the same capability as Borland C++: a project can contain multiple independent targets. So, the developer is able to manage DLL and EXE creation from within the same project.

Patick Philippot, CIS Email, 8/3/95

12.5. I am having [some problem] with exports/imports and extension DLLs.

This is a very complex subject. I actually wrote an article on this and chapter 9 of MFC internals which covers MFC and DLLs (extension and regular) in great detail. The article is on-line at <http://www.vcdj.com>.

-scot@stingray.com - Scot Wingo

12.6. What DLLs should I distribute with my MFC app?

You should review the file REDISTRB.WRI on your Visual C++ CD. This file explains what files are necessary for different kinds of applications; it explains what to do about both Windows files and MFC files.

blaszczak@BIX.com, programmer.tools, 7/9/95

`dumpbin /imports myapp.exe dll_one.dll dll_two.dll | grep -i dll` And recurse through the dll's you find. Won't find dll's loaded by LoadLibrary() however.

null@diku.dk, programmer.tools, 7/10/95

13. MFC Tips, Tricks, and Caveats

13.1. [Macro Questions](#)

- 13.1.1. [What's the difference between IMPLEMENT_DYNAMIC, IMPLEMENT_DYNCREATE, and IMPLEMENT_SERIAL?](#)
- 13.1.2. [How can I declare an abstract base class to be IMPLEMENT_SERIAL?](#)
- 13.1.3. [I can't create an instance because of DECLARE_DYNCREATE!](#)
- 13.1.4. [How do I enable TRACE macros in my app?](#)

NEW!! 13.1.5. [Alternative to the TRACE macro](#)

13.2. [Application Specific Questions](#)

- 13.2.1. [How can I create an application that is initially maximized?](#)
- 13.2.2. [How do I limit my MFC application to one instance?](#)
- 13.2.3. [How do I get my MFC app to use the Registry on Win32 platforms?](#)
- 13.2.4. [How do I programmatically terminate my MFC application?](#)

13.3. [Thread Related Questions](#)

- 13.3.1. [How do I perform background processing in my application?](#)
- 13.3.2. [How do I send a message to another thread?](#)

13.4. [MFC on other Platforms](#)

- 13.4.1. [Is MFC available on the Macintosh?](#)
- 13.4.2. [Is MFC available on OS/2?](#)

13.5. [Miscellaneous](#)

- 13.5.1. [Is MFC DEAD?](#)
- 13.5.2. [Does ATL replace MFC?](#)
- 13.5.3. [Will the next version of MFC support \[some particular feature\]?](#)
- 13.5.4. [Does Microsoft use MFC in their Products? Which ones?](#)
- 13.5.5. [How should I learn / start-learning MFC?](#)
- 13.5.6. [What's the best way to convert my C Window's app to MFC?](#)
- 13.5.7. [Why is my MFC application running slow?](#)
- 13.5.8. [What is AFX.INL and AFXWIN1.INL, etc..?](#)

- 13.5.9. [What the heck is this _t\(\) thing I keep seeing?](#)
 - 13.5.10. [How do I use CMemoryState?](#)
 - 13.5.11. [How do I customize the MFC idle time processing?](#)
 - 13.5.12. [How do I display a "Choose directory" dialog, instead of a "Choose file" dialog?](#)
 - 13.5.13. [I'm having problems using MFC 4.0 and the STL, what could be wrong?](#)
 - NEW!!** 13.5.14. [What is MFC's Y2K compliance for CTimeSpan, COleDateTimeSpan, CTime, and COleDateTime?](#)
-

13.1. Macro Questions

13.1.1. What's the difference between IMPLEMENT_DYNAMIC, IMPLEMENT_DYNCREATE and IMPLEMENT_SERIAL?

IMPLEMENT_DYNAMIC provides run time type information to support macros such as IsKindOf and GetRuntimeClass.

IMPLEMENT_DYNCREATE adds the ability to allow MFC to create the type on the fly. This is required for any concrete data type that will be serialized to a file.

IMPLEMENT_SERIAL also provides a version number for the class and adds the ability to use the >> operator to read the type from a file.

As an example, if a derived class Dog uses IMPLEMENT_DYNCREATE and a base class Animal uses IMPLEMENT_SERIAL, then a Dog can be written with a pointer to either but can only be read by a pointer to Animal.

MFC 3.x provides a constant called VERSIONABLE_SCHEMA to be used with IMPLEMENT_SERIAL to support multiple versions at the same time. The implementation in MFC is broken and fails at runtime.

jimb@turningpoint.com, email, 7/11/95

[**MFC 4.0 NOTE:** It is not true that the versionable schema is broken anymore, and has been fixed in MFC 4.0.]

13.1.2. How can I declare an abstract base class to be IMPLEMENT_SERIAL?

You need a special form of IMPLEMENT_SERIAL that looks like this: Use the regular DECLARE_SERIAL but use IMPLEMENT_SERIAL_ABC shown below instead of IMPLEMENT_SERIAL.

```
#define IMPLEMENT_SERIAL_ABC(class_name, base_class_name, wSchema)
```

```

_IMPLEMENT_RUNTIMECLASS(class_name, base_class_name, wSchema, NULL)
CArchive& AFXAPI operator>>(CArchive& ar, class_name* &pOb)
{
    pOb = (class_name*) ar.ReadObject(RUNTIME_CLASS(class_name));
    return ar;
}

```

anonymous

13.1.3. I can't create an instance because of DECLARE_DYNCREATE!

Question : ClassWizard generated a class for me but used DECLARE_DYNCREATE(...) and declared my constructor as protected. Now, when a try to create an instance of the class I get the compiler error:

error C2248: 'CChkTbl::~~CChkTbl' : cannot access protected member declared in class 'CChkTbl'

Answer : ClassWizard does this because the framework normally handles instantiation for you. i.e. if this is a CView derived class, normally the CDocumentTemplate instantiates the view during the default OnFileNew(), or when you call CxxxDocTemplate->OpenDocumentFile() or something similar. The framework does this so that it will give you an error message if you accidentally try to instantiate it yourself. If you really need to do instantiation outside of the CDocTempate framework, simply change the constructor to be public.

chucks@skypoint.com, programmer.tools, 8/12/95

Answer : The answer is that the DECLARE_DYNCREATE macro lays down a "protected:" directive and leaves it in place. One needs to make sure that anything following DECLARE_DYNCREATE should be "protected" too; if not, one needs to declare "public:" or "private:", as needed.

duane@anasazi.com, email, 8/15/95

13.1.4. How do I enable TRACE macros in my app?

If you use Visual C++ 1.0, run the TRACER application from your Microsoft Visual C++ program group (its icon has the title "MFC Trace Options"). Select

"Enable Tracing," then choose OK.

If you use Microsoft C/C++ 7.0, you must copy the AFX.INI file from the Microsoft Foundation Class Library source directory (by default, C:\C700\MFC\SRC) to your Windows directory (by default, C:\WINDOWS). This file should contain a section such as the following:

```
[Diagnostics]
TraceEnabled = 1
TraceFlags = 0
```

As long as TraceEnabled is set to 1, tracing is enabled.

This AFX.INI file is the same for both C/C++ 7.0 and Visual C++ 1.0. In MSVC 2.x, the TRACE output automatically goes to the 'Debug' window when you run in the debugger.

VC+ Knowledge Base, 6/7/95

NEW!! 13.1.5. Alternative to the TRACE macro

If you create the following pre-processor macro:

```
#ifdef _DEBUG
#define INFO ::AfxTrace("%s(%i): ", __FILE__, __LINE__); ::AfxTrace
#else
#define INFO ((void)0)
#endif
```

and then use this in place of a TRACE then the output is such that if you double click on the line in the output window it takes you to the line of code where the INFO statement is. It is great for navigating code quickly while debugging.

Colin Mackay [colin.angus.mackay@dial.pipex.com],
Sirius Seven Software Ltd.

13.2. Application Specific Questions

13.2.1. How can I create an application that is initially maximized?

For new applications, this can be done with AppWizard during Step 4. During Step 4, choose Advanced..., and then select the Main Frame tab. Check the Maximized option.

For an MDI application, in the `CWinApp::InitInstance()` function, set `CWinApp::m_nCmdShow` to `SW_SHOWMAXIMIZED` before calling `pMainFrame->ShowWindow(m_nCmdShow)`. In an application generated by AppWizard, the code is as follows:

```
// create main MDI Frame window
CMainFrame* pMainFrame = new CMainFrame;
if (!pMainFrame->LoadFrame(IDR_MAINFRAME))
    return FALSE;
m_nCmdShow = SW_SHOWMAXIMIZED; // ADD THIS LINE!
pMainFrame->ShowWindow(m_nCmdShow);
pMainFrame->UpdateWindow();
m_pMainWnd = pMainFrame;
```

In an SDI application, in the `CWinApp::InitInstance()` function, set `CWinApp::m_nCmdShow` to `SW_SHOWMAXIMIZED` before calling `OnFileNew()`. For example, in an application generated by AppWizard, the code is as follows:

```
m_nCmdShow = SW_SHOWMAXIMIZED;

// create a new (empty) document
OnFileNew();
```

MSVC Knowledge Base, 6/4/95

13.2.2. How do I limit my MFC application to one instance?

Look at the Microsoft sample `ONETIME.EXE`, (MSDN CD or <ftp.microsoft.com>). In Brief:

```

const char* MyMainWndClassName = "MyMainWndXQW";
BOOL CMyApp::InitApplication()
{
    //Call base class. Default version does nothing.
    CWinApp::InitApplication();

    WNDCLASS wndcls;

    // start with NULL defaults
    memset(&wndcls, 0, sizeof(WNDCLASS));

    // Get class information for default window class.
    ::GetClassInfo(AfxGetInstanceHandle(),
        "AfxFrameOrView",
        &wndcls);

    // Substitute unique class name for new class
    wndcls.lpszClassName = MyMainWndClassName;

    //Register new class and return the result code
    return ::RegisterClass(&wndcls);
}

```

And:

```

BOOL CMyApp::FirstInstance()
{
    CWnd *PrevCWnd, *ChildCWnd;

    // Determine if another window with our class name exists...
    PrevCWnd = CWnd::FindWindow(MyMainWndClassName, NULL);
    if (PrevCWnd != NULL)
    {
        // if so, does it have any popups?
        ChildCWnd=PrevCWnd->GetLastActivePopup();

        // Bring the main window to the top
        PrevCWnd->BringWindowToTop();

        // If iconic, restore the main window
        if (PrevCWnd->IsIconic())
            PrevCWnd->ShowWindow(SW_RESTORE);

        // If there are popups, bring them along too!
        if (PrevCWnd != ChildCWnd)
            ChildCWnd->BringWindowToTop();

        // Return FALSE. This isn't the first instance
        // and we are done activating the previous one.
        return FALSE;
    }
    else

```

```

        // First instance. Proceed as normal.
        return TRUE;
    }
    CMyApp::InitInstance()
    {
        if (!FirstInstance())
            return FALSE;
        //...
    }

```

Niels Ull Jacobsen (null@diku.dk), programmer.tools, 6/19/95

See also MS Knowledge base article Q124134 ("Allowing Only One Application Instance on Win32s") and Advanced Windows NT , chapter 7, "Prohibiting Multiple Instances of an Application from Running: The MultInst Sample Application" (available on the MSDN).

Niels Ull Jacobsen (null@diku.dk), email, 8/8/95

1. *update - these were posted to mfc-l:*

I have each InitApplication() create a semaphore. If GetLastError() returns ERROR_ALREADY_EXISTS then I know that some other application is already running and has gotten that far so I bail.

```

Yourapp::InitInstance()
{
    hMutexOneInstance = CreateMutex(NULL, TRUE,
        _T("PreventSecondInstance"));
    if(GetLastError() == ERROR_ALREADY_EXISTS)
        bFound = TRUE;
    if(hMutexOneInstance)
        ReleaseMutex(hMutexOneInstance);
    return (bFound == TRUE) ? FALSE : TRUE;
}

```

mcontest@universal.com

There is a nice section in Jeffrey Richter's book Advanced Windows NT about this. Essentially, it uses shared data segments between processes.

Step1:

=====

In your main file, add:

```
#pragma data_seg(".SharedData")
LONG nUsageCount = -1;
#pragma data_seg()
```

Step 2 :

=====

In your Application's InitInstance(), call:
InterlockedIncrement (&nUsageCount);

This function returns the incremented value of the variable. If it is non-zero, you know that you are not the first App.

In your Application's ExitInstance() call:
InterlockedDecrement(&nUsageCount);

Step3:

=====

In your DEF file, have the following lines: (Note that the segment name you give here should match the one in the application's main file.)

```
SEGMENTS
.SharedData shared
```

abalakri@us.oracle.com

You'd better use one of the built-in synchronization methods. See Q124134 : Allowing Only One Application Instance on Win32s for a sample of using a memory mapped file for synchronization. It doesn't include starting the previous instance, but if you detect that you're not the only one running, it should be pretty simple: if CreateFileMapping fails, try to find the previous instance from the window class name. If it's not found, sleep for a while and start over (with CreateFileMapping). In this way, either you will find the other instance when it gets around to creating it's window or CreateFileMapping will eventually succeed. The advantage of using CreateFileMapping instead of CreateObject is that it also works on Win32s.

- nuj@kruger.dk

Note: *There's a sample of this that was contributed by john@jing.com (John Xu) called onetime4.zip that is in the MFC FAQ archive (see section 2.2.6 for MFC FAQ archive details).*

13.2.3. How do I get my MFC app to use the Registry on Win32

platforms?

Just make a call to `SetRegistryKey("MyCompany")` in your `CWinApp` class and pass it a string (typically your company name). This string will define where in the registry the data goes: `HKEY_CURRENT_USER\Software\MyCompany\`

After making this call just use the normal `WriteProfilexxx()` routines and the data will go to the registry instead of to an INI file. It works well and is simple!

Brett Robichaud, brett_robichaud@tallysys.com, 6/23/95, programmer.win32

13.2.4. How do I programmatically terminate my MFC application?

MFC does not provide a public function to gracefully exit an application. A method for dealing with this is to create a function in your application like the following:

```
void ExitApp()
{
    // same as double-clicking on main window close box
    ASSERT(AfxGetApp()->m_pMainWnd != NULL);
    AfxGetApp()->m_pMainWnd->SendMessage(WM_CLOSE);
}
```

As you can see, this is implemented as a global function, which can be called from anywhere in your application. It simply sends a `WM_CLOSE` message to your application's mainframe window. This initiates an orderly shutdown of the application.

If you are using MFC, version 2.5 or later, you can take advantage of a new global MFC function, "`AfxGetMainWnd`", to simplify the code:

```
void ExitMFCApp()
{
    // same as double-clicking on main window close box
    ASSERT(AfxGetMainWnd() != NULL);
    AfxGetMainWnd()->SendMessage(WM_CLOSE);
}
```

NOTE: Always call `CDocument::SetModifiedFlag()` after changing your document data. This will ensure that the framework prompts the user to save

before shutdown. If you need more extensive control over the shutdown procedure, you can override CDocument::SaveModified().

MS FAQ, 6/25/95

13.3. Thread Related Questions

13.3.1. How do I perform background processing in my application?

Many applications perform lengthy processing "in the background" during intervals that the user is not otherwise interacting with the application. In an application developed for the Microsoft Windows operating system, an application can perform background processing by splitting a lengthy process into many small fragments. After processing each fragment, the application yields execution control to Windows using a PeekMessage() loop.

An application developed with the Microsoft Foundation Class Library can perform background processing either by using the PeekMessage() loop in the library code's main message loop or by embedding another PeekMessage() loop in the application.

See MSVC Knowledge Base Q99999 for more info.

MSVC Knowledge Base 6/7/95.

13.3.2. How do I send a message to another thread?

Try using SendNotifyMessage() to send messages to another thread.

null@diku.dk, Neils Ull Jacobsen via programmer.misc, 6/18/95

13.4. MFC on other Platforms

13.4.1. Is MFC available on the Macintosh?

Yes! Microsoft has something called the Visual C++ cross-platform edition. This product lets you cross-compile for the 68000 based Macs and I think they have a PowerPC version coming out soon. Contact Microsoft directly for more info.

scot@stingray.com, 7/27/95

13.4.2. Is MFC available on OS/2?

No!

13.5. Miscellaneous (1/3)

13.5.1. Is MFC Dead?

I thought this would be a great time to address a VERY WRONG rumor/opinion that MFC is dead that I've been seeing on the newsgroups for the last several months. There have been so many rumors that I thought it would be fun to take a look at the top ten rumors behind MFC's ill reported demise:

They are:

10. Microsoft will replace MFC with ATL.
9. I read that MFC is in "maintenance mode" in a magazine (it's true that Windows Tech Journal did incorrectly report this. The irony is that Oakley actually "died" and MFC is really alive.)
8. VC++ 5.0 didn't have a major release of MFC, so it is dead.
7. There are just no more features that could be added to MFC.
6. Java will kill C++, thus killing MFC.
5. There is nobody on the MFC team anymore - they have all moved to different areas of Microsoft.
4. Microsoft stopped the Visual C++ subscription program, so MFC is dead.
3. I read a posting on a newsgroup from this guy in Portugal that heard from a friend at Microsoft that MFC is dead.
2. Stingray has done such a great job filling the holes in MFC that Microsoft decided they couldn't keep up.

And finally, the number one reason that people think MFC is dead....

1. Aliens have abducted the MFC team and they are now hiding behind a Comet touring the Universe, instead of working on MFC.

The truth of the matter is that MFC is NOT DEAD! Don't believe me? Well maybe you will believe it from the horses mouth, here is an MFC FAQ submission from Mike Blaszcak, who is the current MFC Team Lead, on the subject of MFC's incorrectly rumored demise:

Q: Is MFC dead?

A: No. MFC continues to grow and change as the operating system grows and changes, and as the needs of our users continue to grow and mature. During the summer of 1997, some MFC team members will provide presentations under strict non-disclosure agreements that explain some of the planning that's happening for the next version of the product. But the caveat in the above question still strongly applies: the development of the product might change directions to fill an unforeseen pressing need, but the product is by no means dead.

-Mike Blaszcak (mikeblas@microsoft.com)

So there you have it folks, those of you that were genuinely concerned (like me!) that MFC was in maintenance mode, or never going to see another major revision, you can sleep again. For those of you that continue to spread the incorrect rumor that MFC is dead - stop it! If you see anyone posting to newsgroups that MFC is dead, please help set the record straight and feel free to copy this portion of the MFC FAQ and paste it right into a reply.

I think that this was a big conspiracy by the old Borland OWL team (or maybe JavaSoft) to scare people away from MFC.

Together we can stop the madness! LONG LIVE MFC!!! (Wow - I'm glad I finally was able to get that off my chest <g>)

Another piece of exciting news is that one of the old-time MFC team members is back on the MFC Team! After a tour through one of Microsoft's Java teams, Dean has rejoined the ranks of the MFC Team. FYI, Dean wrote large chunks of the OLE support in MFC, etc.. Rumor also has it that Dean has a girlfriend now, so he probably won't be pulling too many all-nighters, at least on MFC <g>

scot@stingray.com

13.5.2. Does ATL Replace MFC?

No. ATL and MFC are designed to solve two very different problems.

- mikeblas@microsoft.com

13.5.3. Will the next version of MFC support [some particular feature]?

The MFC Team grows MFC's feature set as time and resource allow.

MFC features are added when the MFC team feels that MFC can provide a clear win for C++ programmers over using some API or some system feature directly. MFC tracks features in the operating system as well as new strategic technologies that Microsoft introduces.

Unfortunately, some people in the industry have irresponsibly begun predicting what MFC will or won't support. If someone not on the MFC Team claims to know whether MFC will or won't support a particular feature, they're wrong: they don't know. If someone who is on the MFC Team says that the team is planning support for a feature, they're probably right--at that particular moment in time. Because software development schedules change and are influenced by all sorts of surprises, the feature may or may not make the final version of the product despite the best efforts of the team members.

- mikeblas@microsoft.com

13.5.4. Does Microsoft use MFC in their products? Which ones?

There are many Microsoft apps written in MFC. Sometimes its just not obvious... (to name a few: Bookshelf, Bob!, WordArt OLE server, Visual C++ (of course), Win95 paint, Win95 WordPad, some portions of Win95 FAX software, some Win95 games I know of...).

In the future, there are more apps coming out using MFC. I don't have a way to track all of these uses, so there is certainly more that I'm not aware of or can't remember. I don't expect Word or Excel to ever use MFC -- they have way too much legacy code and they don't see any customer benefit to rewriting to MFC. But my point is -- definitely for new code, Microsoft is using MFC. Even some "old" code is taking advantage of MFC in future versions.

Dean McCrory, MSMFC, 6/8/95

[NEXT...](#)

14. Wizard Questions

14.1. [How can I change the AppWizard options I selected for my application?](#)

14.1. How can I change the AppWizard options I selected for my application?

Unfortunately, AppWizard will not modify already created and modified applications. You need to generate a new skeleton application using AppWizard and either merge your code into that or compare that skeleton with your original to see what's different. The good news is that it's mostly just a minor tweak to add/remove OLE support or MDI or something like that. It usually is just changing your CWinApp derivation, your frame window, or something like that.

scot@stingray.com, 5/31/95

15. Visual C++ Questions

I'm trying to keep these as generic as possible, but these types of questions tend to be very release-specific, so where possible, I've included the version that's being questioned.

15.1. [Windows 95 Questions](#)

- 15.1.1. [I'm running MSVC 2.0 and I can't get Windows 95 look and feel in my dialogs.](#)
- 15.1.2. [When I compile under Windows 95, it flashes in and out of DOS mode.](#)
- 15.1.3. [Can MSVC 1.5 or 2.0 be used for Windows 95?](#)

15.2. [Configuration Questions](#)

- 15.2.1. [What are the memory requirements of MSVC?](#)
- 15.2.2. [How can I get MSVC 2.1, I can't find it in stores?](#)
- 15.2.3. [Should I run under Win 95 or NT?](#)
- 15.2.4. [How do I set the include/lib/exe directories?](#)
- 15.2.5. [I can't compile any programs! Help!](#)
- 15.2.6. [Can I setup a project to search different paths for source?](#)
- 15.2.7. [How do I build a browser file for all of MFC?](#)
- 15.2.8. [What does the "unexpected end of file while looking for Precompiled header" message mean?](#)
- NEW!!** 15.2.9. [Is there any way of seeing UNICODE strings interpreted correctly in VC++ debug watch window?](#)

15.3. [Language Feature Questions](#)

- 15.3.1. [Does MSVC support templates and exception handling?](#)
- 15.3.2. [Does MSVC support the Standard Template Library?](#)
- 15.3.3. [Does VC++ support RTTI?](#)

15.4. [Religious Questions](#)

- 15.4.1. [Which is better OWL or MFC, BC++ or MS MSVC?](#)
- 15.4.2. [Should I use Visual C++ or Visual Basic?](#)
- 15.4.3. [MFC philosophy \(from a microsoftie!\)](#)
- 15.4.4. [Should I use the MFC collections or the STL collections?](#)

15.5. [Advanced Visual C++ Tips and Tricks](#)

- 15.5.1. [How do I see the MSVC 'Easter Eggs'?!](#)
- 15.5.2. [What are the command line options to MSVC?](#)
- 15.5.3. [How to change the color of ClassWizard generated code!](#)
- 15.5.4. [Ten cool things you can do in the VC++ editor!](#)
- 15.5.5. [How do I speed up VC++ compiles?](#)
- 15.5.6. [Make all MFC keywords a different color!](#)
- 15.5.7. [Easy way to launch 4.x projects without mdp files.](#)

15.6. [Miscellaneous Visual C++ Questions](#)

- UPD!! 15.6.1. [What are all these files that MSVC is creating?!](#)
- 15.6.2. [How do I change from using MFCXX.DLL to using a statically linked library?](#)
- 15.6.3. [How do I report a bug to Microsoft?](#)

15.7. [Visual C++/MFC Version Specific Hints, Problems, etc..](#)

- 15.7.1. [Why aren't my windows being created in MFC 4.0?](#)
 - 15.7.2. [How do I convert a VC++ 2.x makefile over to 4.x?](#)
 - 15.7.3. [ClassWizard has problems wiring controls to members in international 4.0 versions, what the heck do I do?](#)
 - 15.7.4. [VC++ 4.0 crashes all the time on me!!! HELP!?!?!?!?!](#)
 - 15.7.5. [How do I add components \(not just classes\) to the Component Gallery?](#)
 - UPD!! 15.7.6. [What is up with the Visual C++ 5.0 help system!?](#)
-

15.1. Windows 95 Questions

15.1.1. I'm running MSVC 2.0 and I can't get Windows 95 look and feel in my dialogs, what's wrong?

In MSVC 2.0, go to the Project/Options dialog, select "Linker" and change the linker command line to include: /subsystem:windows,4.0 This is the default in MSVC 2.1.

Chris Marriott, chris@chrism.demon.co.uk, 5/25/95, via programmer.misc

15.1.2. When I compile under Windows 95, it flashes in and out of DOS mode.

Remove or rename any dosprmt.pif files in your win95 and windows directories. If this doesn't work, do the same with all pif files.

scot@stingray.com, 5/25/95

In the Win95\system directory there is a pif for Conagent. It is this pif file that a person has to change to Window screen mode instead of Full Screen mode. Worked like a charm on mine after running about a month with the screen blanking...

RockyMoore@aol.com, email, 7/16/95

15.1.3. Can MSVC 1.5 or 2.0 be used for Windows 95?

You can run both versions under Windows 95. Visual C++ 1.5x can ONLY generate 16-bit applications. You 'can' run these on Windows 95, but it's more desirable to have true 32-bit applications, which MSVC 2.0 generates. These same rules apply to NT.

scot@stingray.com 6/5/95

15.2. Configuration Questions

15.2.1. What are the memory requirements of MSVC?

MSVC 1.5.x will work on a 486 class machine with a minimum of 4MB of RAM. 8MB will make your life much better. MSVC 2.x needs at least 16MB. Microsoft recommends 20MB on the box.

Note: VC 1.5 takes a big (3x) performance hit under all operating systems if you have less than 16 Meg. I published an article about this (and other performance enhancements) in Windows/DOS Developers Journal in May, 95.

jimb@turningpoint.com, email, 7/11/95

15.2.2. How can I get MSVC 2.1, I can't find it in stores?

The point releases are part of the MSVC subscription (See section 2.1.1.). Only major releases are sold through retail, so you'll need to buy a subscription to get the point releases. Subscriptions are available from Microsoft, Programmer's Paradise and Programmer's shop. (See section 2.1 for info)

15.2.3. Should I run under Win 95 or NT?

Life is a bunch of trade off's. The trade off's here are if you run under 95, you won't need as much memory, but the OS is not as robust as NT, so you will probably crash more. On the flip side, if you run under NT (a full 32-bit OS), you won't crash as much, but alas, you will need more memory. Personally, I prefer the NT route. NT catches all memory problems and handles them nicely for you without destabilizing the system.

scot@stingray.com, 6/18/95

15.2.4. How do I set the include/lib/exe directories?

MSVC versions < 2.0 used environment variables INCLUDE/LIBDIR/EXEDIR. MSVC versions > 2.0 uses settings in the Visual C++ Tools menu. Select "Tools"

and then "Options" menu item. A dialog opens, and select "Directories" tab button. There you can set Include directories, Library directories, Executable directories...

lee@sam1.info.samsung.co.kr (Lee, Jin Woo), via programmer.win32, 6/10/95

15.2.5. I can't compile any programs! Help!

Question: I just installed VC1.51 on my system. Installation proceeded smoothly without any errors. However, I can't compile any programs! Even when I try to compile the simplest and shortest programs, I get this error message:

Can't open the compilation response file

Answer : A common cause of this type of error is having a TMP or TEMP environment variable which isn't pointing at a valid directory. E.g. your AUTOEXEC.BAT file says:

```
set TMP=C:\TMP
```

and you don't have a "C:\TMP" directory.

Chris Marriott, chris@chrism.demon.co.uk, programmer.misc, 6/18/95

15.2.6. Can I setup a project to search different paths for source?

I would like to have my projects first search for the source in a local directory, and if it can't find it there, search a network drive.

TODO: Find answer to this one, seems good. Asked on tools 6/21 by

Joe Kinsella, jkinsella@procd.com

15.2.7. How do I build a browser file for all of MFC?

It's nice to be able to browse the MFC classes and source code, the best way to do this is via some Microsoft supplied .BAT files for the Microsoft Software Library:

BLDBSC15.BAT - for VC++ 1.5

BLDBSC20.BAT - for VC++ 2.0

BLDBSC21.BAT - for VC++ 2.1

Basically, you want to invoke the compiler on every file with /Zs as the option, which means generate the .sbr file and no .obj. Next, you want to run bscmake.exe on the generated .sbr files to create your .bsc file. The MFC makefile will complain about the obj's being missing, but you can ignore that warning. It will take significantly longer and more time to create both .sbrs and .objs.

scot@stingray.com, 6/25/95

It is worth noting that VC 2.1 has a pre-built browser file that is on the CD-ROM but is not installed. It can be loaded at any time into the IDE with the File/Open command.

jimb@turningpoint.com, email, 7/11/95

If you are using the 32-bit versions, you can get more information in the README.TXT file in the \msvc20\mfc\src directory. What it tells you is to move to that directory and from the dos prompt run:

nmake DEBUG=1 BROWSEONLY=1

billb@microsoft.com, mfc-l, 7/17/95

15.2.8. When I compile I get: "fatal error C1010: unexpected end of file while looking for precompiled header directive" - why?

You need to #include "stdafx.h" at the top of your file.

- [hrubin@nyx.net](mailto:hruhin@nyx.net)

NEW!! 15.2.9. Is there any way of seeing UNICODE strings interpreted correctly in VC++ debug watch window?

Under Tools|Options Goto the Debug Tab, there is an option 'Display unicode strings' turn it on and it should display unicode strings correctly.

Chris Depetris [chris@CYBERNEX.NET], VCPP mailing list, 6/30/98

15.3. Language feature questions

15.3.1. Does MSVC support templates and exception handling?

The 32-bit Visual C++ 2.0 release introduced templates and exception handling. There is no support for these C++ features in 16-bit versions of the product.

It is worth noting that the 16 bit implementation of THROW and CATCH are brain-damaged. They do not clean up the stack as it unwinds.

jimb@turningpoint.com, email, 7/11/95

15.3.2. Does MSVC support the Standard Template Library?

Yes! See FAQ 11.27 for some hints. This support was added with 4.0

15.3.3. Does VC++ support RTTI?

Yes, version 4.0 does.

scot@stingray.com, 6/7/95

15.4. Religious questions

15.4.1. Which is better OWL or MFC, BC++ or MS MSVC?

This is constantly getting asked on Usenet. The answer really depends on what you are doing. If you demand the latest C++ features such as templates, RTTI, etc.. BC++ usually has these first. But be careful, when Borland went from OWL 2 to 3 (or was it 1 to 2?) they made everybody completely rewrite their applications. Applications that were written with MFC 1.0 still work with the newer versions of MFC today! The best advice is to get the features from each vendor and see what's best for your situation.

scot@stingray.com 6/8/95, comp.lang.c++

15.4.2. Should I use Visual C++ or Visual Basic?

(Here's a funny analogy I saw posted->) Well, the principle is the same (create interface, add code to respond to events, add code to do something useful). But programming in VB is like riding a kiddy bike, while programming in C++ is like driving a Formula 1 racing car - be prepared for accidents.

But seriously, VB is interpreted, MSVC is compiled, therefore MSVC programs should be much faster (if written well). It's generally believed that VB is the best way to 'prototype' a user-interface, but when it comes to code reuse and solving larger problems, VB breaks down. Visual C++ on the other hand is not as easy to use as VB, but once you learn it, you can write much more complex programs than with VB. The readers digest version of all this is: It depends on the application you are writing.

scot@stingray.com, 6/18/95

15.4.3. MFC philosophy (from a microsoftie!)

Just a note about the classes provided in MFC... keep in mind that these classes are designed as "general-purpose" classes, not "all-purpose" classes. No one implementation can have performance characteristics that are ideal for *all* possible applications. It is up to you, the programmer, to evaluate a provided

implementation and decide whether it is appropriate for the task at hand. If it is not, you have several alternatives: modify or subclass the provided implementation; find a more appropriate class from another source; or write one from scratch.

elsbree@msn.com, mfc-l, 7/15/95

15.4.4. Should I use the MFC collections or the STL collections?

In my opinion, STL containers are much more intuitive to use. However, one reason to use the MFC container classes is that they port to UNIX via Bristol's Wind/U more reliably than their STL counterparts. Ironically, the "standard" template library is not currently very standardized in the UNIX environment.

-- John.Cook@waii.com

15.5. Advanced Visual C++ Tips and Tricks

15.5.1. How do I see the MSVC 'Easter Eggs'?!

FYI: Easter eggs are hidden screens that usually have secret initials, development team names, etc..

- To see the spinning PLUS with the MSVC team members:

In MSVC 2.0 or 2.1:

A. Bring up the about box: help/about

B. Press control-tab and double click in the middle of the box at the same time.

NOTE: You must have the MSVC CD loaded for this to work, I think.

If above does not work, try this-> copy the MSVCCRD.DLL from the CD to msvc\bin. CTRL+Double-click the picture in the dialog.

note: release the CTRL key at the same time as the second click.

ohallorj@pwc-tpc.ca, Jeff O'Halloran

- To see the Visual C++ 4.0 about box, double click in the about box while the CD-ROM is in the drive and you are holding the control key down. The OK button should go away. Wait about 3-5 minutes and you should see a pretty neat flight simulator! You can browse the picture by opening msdevcrd.dll which lives in msdev\bin.

scot@stingray.com

15.5.2. What are the command line options to MSVC?

- MSVC /V - Runs the compiler in a dos box (version 1.5 only?)

mark@techop.demon.co.uk

- MSVC /NOLOGO - Runs MSVC without the about box, saves second or 2.
- MSVC /bppassc:yes - Enables a cool breakpoint counter in the 'breakpoints'

dialog (under the Debug menu).

jimb@turningpoint.com, email 7/11/95

- MSVC -p - attaches VC++ to a process.

15.5.3. How to change the color of ClassWizard generated code!

The default color for text modified by the ClassWizard is black print on a white background. Visual C++ is aware of the code that the ClassWizard modifies and can highlight these changes for you using a different color.

Highlighting the text ClassWizard modifies makes the new modifications very obvious and easy to find. To do this, from the Tools menu, choose Options. Then select the Format tab in the Options dialog box. From the Colors Listbox, select Wizard Code. Then define the colors for text and the background.

NOTE: Only applies to VC++ 2.x (and 4.x), but it's waaaay cool!

MS FAQ, 6/25/95

15.5.4. Ten cool things you can do in the VC++ editor!

1. CTRL+ALT+T shows your tab characters
2. SHIFT+ESC closes the active dockable window
3. CTRL+F3 finds the next occurrence of the current word
4. CTRL+M finds the matching bracket
5. CTRL+> and CTRL+< find the next/prev matching #ifdef/#else/#endif
6. CTRL+SHIFT+R starts/stops macro recording
7. CTRL+SHIFT+P plays recorded macro
8. Right-click #includes to open the header file
9. TAB/SHIFT+TAB indent or

un-indent selected lines

- Hold ALT key down to select columns

Jeff Henshaw, borrowed of MFC PDC slide, 6/25/95

NOTE: Not all of these work in VC++ 1.5x, most do in 2.x and 4.x

15.5.5. How do I speed up VC++ compiles?

Use precompiled headers, turn off the browser (Options:Project:Compiler:Listing Files:Browser Information), increase and/or decrease the size of your disk cache, tinker with the INCLUDE and LIB environment variable directory orders. That's about it.

ebarb@iadfw.net, programmer.tools, 8/19/95

15.5.6. Make all MFC keywords a different color!

Download the file USERTYPE.DAT from the FAQ Archive (See 2.2.6) and follow the directions in there. Try it, you'll like it!

NOTE: This has been updated for 4.0

scot@stingray.com, 8/31/95

15.5.7. Easy way to launch 4.x projects without mdp files.

For very large projects you might not bank or put mdp files into the version control system. You just want to grab the makefile and go. The problem is that Visual C++ does not easily recognize .mak files. Here's an easy way to launch right from a makefile to VC4.0 and have a .mdp file created.

1. Bring up the registry edit (registry.exe for Windows 95 and regedt32.exe for Window NT 3.51)
2. Activate the HKEY_CLASSES_ROOT window
3. Add a new key

`.mak`

4. Add a new string value

`mdpfile`

5. Close the registry

That's it. Now you can just select any Visual C++ 4.0 makefile from the file

manager or the explorer and the Visual C++ IDE will come up and a project workspace (.mdp) will be create for that makefile. You no longer have to go into the visual environment do a "file open", enter a filename, and select "open as...makefile". This is definitely a time saver.

G_LEDONNE@msn.com

15.6. Miscellaneous Visual C++ questions

UPD!! 15.6.1. What are all these files that MSVC is creating?!

Here's a table that explains the common extensions and what MSVC uses them for:

<u>Extension</u>	<u>Use</u>	<u>Notes</u>
APS	App Studio File	
BSC	Browser File	SBR's combined into 1 BSC
CLW	Class Wizard File	
DSP		

Project file for DevStudio Contains information specific to a project. Used to be in the .mak file. DSW Workspace file for DevStudio It organizes all the projects into a single workspace. ILK Incremental Linker File NCB Contains the ClassView information information generated by the parser which is used by ClassView, WizardBar, and Component Gallery. Added in 4.0, delete if you have problems. ODL COM interface definition Used to generate a type library OPT Options file for DevStudio It stores all the user options you create for your workspace PCH Pre-Compiled Header File PDB Debugging Info /Z7 Affects This RES RC 'object' File Linked Into EXE SBR Browser Info for 1 File Combined Into BSC VCP Visual C++ State File

Thanks to [Kevin L. McCarthy@iacnet.com](mailto:Kevin.L.McCarthy@iacnet.com) for additional entries .

%%TODO: Fill this table in more completely.

15.6.2. How do I change from using MFCXX.DLL to using a statically linked library?

You need to flick a few settings in the build settings windows:

In the C/C++ tab, select the "General" Category:

1. Remove "_AFXDLL" from the list of preprocessor definitions.
2. In the "Code Generation" tab, select "Multithreaded" or "Debug Multithreaded", which ever is appropriate for your build. I
3. n the "Resources" tab, make sure that "_AFXDLL" doesn't appear in the list that's in the "Preprocessor Definitions" edit.

-Mike B - blaszczak@BIX.com

15.6.3. How do I report a bug to Microsoft?

[ed. note: Here's a fun answer from Mike B on this one, seems Mike gets lots of "how do I", or "is this a bug" emails. This FAQ actually has good info, I kept some of the personal stuff because it livens it up somewhat.]

I think need to explain something about bug reporting.

While everyone on this list knows that I work at Microsoft, and most people know that I work on MFC, it seems that very few people understand how bugs should be reported to Microsoft about Microsoft products.

If you're a Microsoft Select customer, or you're involved with Microsoft Consulting Services, you should talk to your Corporate Account Manager or Engagement Manager about how you should get information about bugs to and from Microsoft.

If you're a user who has bought Visual C++, you need to use the "Technical Support" command in the "Help" menu of Visual C++. This help topic explains a lot about contacting PSS, Microsoft's Product Support Services.

If you have a support agreement with PSS, or if you don't have an agreement with PSS, you can call to report a bug without charge. You will have to pay for your own long-distance phone call, but you'll not be debited for the time you spend making the report by Microsoft.

Everyone should have understood this, but I guess many don't. Further:

Microsoft certainly encourages third-party and peer support for Microsoft products. You should be able to find certified Microsoft consultants near where you work, and you should be able to find peer-support places to talk about Microsoft products. MFC-L is an example of one, and the MSMFC forum on CompuServe is another.

The problem is, though, that a few people have written to me (and, I suppose, other MS-people on the list) personally requesting that I handle a bug report or requesting that I provide one-on-one help.

Frankly, I just don't have the time to do that. I try to answer lots of questions in public places (like on MFC-L, or on MSMFC, or in the occasional book) to help out and encourage the peer-support communities. But I don't have time to hand-hold people through problems, and I don't have the time to accept bug reports.

"Accepting a bug report", if you don't realize it, means that I would get from you a complete reproduction case, figure out what is wrong, reproduce it myself, and report it to the appropriate folks. I can't skip those steps because it makes noise for the appropriate folks if they've already heard about the problem before.

I really don't mean to be a prick: I just can't make (or even imply) promises that I know I'm not capable of keeping.

Now, on the other hand, if you report a problem to a public place (like MFC-L or to CompuServe) and I see it, I might try to reproduce it. I'll steal it from the list and take it to work, where I can see if it was reported before. And when I have time, I can fool with it. But I can NOT imply any obligation--I can't promise that I'll do even this much... but since

I'm interested in MFC, I usually try. It's unofficial--all it is some guy who wants to go out of his way to help out.

So, if you find what you think is a bug, here's what you should do:

- 1) Panic.
- 2) Threaten to nuke my neighborhood.
- 3) Try to consolidate the problem.

4) Take that consolidation to another project on another machine and try it

5) If it still is a problem, step through the code. THINK..

6) THINK some more..

If you're still sure its a bug, do one or more of these things:

a) Call PSS.

b) Talk to a support professional, like a certified consultant.

c) Write to MFC-L or the MSFMC group, or the MFC bulletin board on MSN.

but realize that only (A) is the official, guaranteed way to get help.

My job is to write software, and I try to be pretty good at that. (I'm pretty good at going to hockey games, too.) But I really can't also do the job of product support on a full-time, full-responsibility basis. Sending bug reports directly to me will do nothing more than make me feel awkward and say "Sorry, can't help" to you.

Thanks for understanding,

- mikeblas@interserv.com

15.7. Visual C++/MFC 4.0 Specific hints, problems, etc..

15.7.1. Why aren't my windows being created in MFC 4.0?

The problem is probably that you are relying on one of the MFC 3.x and earlier registered window classes for the class name passed to your `::Create()`. For example, a very popular way to create a window with a white background is to use "AfxViewOrFrame". In MFC 4.0, not only have all of the window class names been changed, but they also happen dynamically when needed!

The solution is to use `AfxRegisterWndClass()` and pass the results to `Create()`. Be sure to specify the right flags to get the desired results. Also check out tech note number 1, which has been revised for 4.0.

- *scot@stingray.com, 1/20/96*

15.7.2. How do I convert a VC++ 2.x makefile over to 4.x?

1. Select File/Open...
2. Select your *.mak file
3. Select Open as: Makefile
4. Select Open, then VC will offer you to convert it to a VC workspace - this is what you want.

15.7.3. ClassWizard has problems wiring controls to members in international 4.0 versions, what the heck do I do?

To test and see if you have the problem:

1. In VC version 4.0 use the app wizard to create a dialog based application.
2. Then use the dialog resource editor to add controls etc onto the dialog box generated above.
3. Then fire up the class wizard to associate member variables to the added controls. The control ID's are NOT listed in the command ID windows (the IDCANCEL and IDOK are there and that's it!) of the member variables

property page.

The control ID's ARE in the message map property page but not on the member variables property page! Huh?

The fix:

The fix is to change the language property of the dialog template (highlight the name of the dialog box in the resource window and click right mouse button). For me I had to change the language property from English (US) to English (Australian), only after doing this do the member variables of control's appear. I assume this property must match your Win95 chosen language, in any case the change fixed it for me and a German user who had the same problem.

doug@psy.uwa.edu.au, email, 12/13/95

15.7.4. VC++ 4.0 crashes all the time on me!!! HELP!?!?!?!?!?

I haven't personally experienced these problems, but here's some solutions (some are probably wives tales) for solving the problems:

1. Make sure you have downloaded the service update from:
<http://www.microsoft.com/visualc>.
2. If you're using subprojects, this is probably the culprit, they're very unstable.
3. If you're using 200+ files, try and carve your project into DLLs.
4. Try removing the .ncb file, this is where the ClassView info is stored.
5. If you converted your project over from 2.x and are having problems - start fresh with a 4.0 project and insert your files into it.

15.7.5. How do I add components (not just classes) to the Component Gallery?

The APIs used for the Component Gallery have not been made public yet by Microsoft.

UPD!! 15.7.6. What is up with the Visual C++ 5.0 help system!?

It is widely known (at least everyone I talk to) that the VC++ 5.0 help system is

very very bad! Microsoft replaced the wonderful, speedy RTF based WinHelp viewer system with an HTML based IE viewer system. Why is it slow? Well now you have to basically have a Alta Vista on your desktop to search since it is basically HTML. There is an indexing system, it is suppose to get better, but it is really bad right now.

Here's what I do...

1. Remove all of the help files.
2. Install Visual C++ 4.2, choose only the IDE and help file options. Install into some unique directory like c:\vc42_help.
3. Now use VC++ 4.2 for help and VC++ 5.0 for compiles.

Sounds cheesy and it is, but the 5.0 help is really that bad if you are a heavy user like I am (not sure if that is a good or bad thing).

-scot

I use the InfoViewer as a separate application. The InfoViewer can be found on the MSDN CD-ROM as well as part of the Windows CE Handheld SDK. It is much easier to hit the button on the taskbar for InfoViewer and pop it open rather than click the tab for the InfoViewer pane (I should say pain..) in the Developer Studio. It is about as easy to copy and paste code than to say have both open in equally sized windows and type the stuff I see. We should all be getting used to copy and paste anyway by now.

Mark S. Roush <jsbach@kuhub.cc.ukans.edu>

16. Third Party MFC Products

This section lists products that make the MFC developer's life easy. No OCX/VBX type products are listed here, go check some catalog. Where possible, I've put a review by someone!

Stingray Software, Inc.

(800) 924-4223

(919) 461-0672

sales@stingray.com

<http://www.stingray.com>

Company that develops and markets MFC extension class libraries. Co-founded by Scot Wingo - the original MFC FAQ dude. We have a full family of MFC extensions that should save you tons of time in your MFC projects. Many of our products were inspired by FAQs!

%%TODO: get a review

Premia CodeWright Fusion

(800) 547-9902

(503) 641-6000

Great add-in editor for MSVC, they have both 16 and 32-bit versions.

%%TODO: Get a review, anyone?

Nu-Mega Bounds-Checker

(800) 468-6342

(603) 889-2386

<http://www.numega.com>

Lets you know if you've clobbered memory, has a special MFC feature so you can see class names and paths through message maps.

%%TODO: Review? We use it at Stingray all the time. Still not as good as

Purify on Unix, but all we've got right now.

17. MFC FAQ Revision History

- 1.0 Beta 5/25/95

Did heavy work on TOC, more info, history. Not many FAQs yet.

- 1.0 Final Beta 6/2/95

Put in some FAQs I keep seeing. Integrated feedback from mfc-l posting. Posted one last time to mfc-l for final feedback before FCS...

- 1.0 FCS! 6/8/95

Put in some FAQs from MSVC KB
Got lots of good feedback from mfc-l
Got some new FAQs from Usenet
Integrated news from MFC Conference, MSVC 4/2.2, etc..
Posted to Usenet and CompuServe for first time.

- 1.1 6/25/95

Lot's of new stuff from MFC PDC
Lot's of cserve and Usenet FAQs too
redid VC++ section
Lots of good book reviews coming out.
Grabbed some stuff from new MS FAQ too.

- 1.2 7/30/95

Updated from 1.1
Lots of book reviews got emailed and added.
Windows common controls section added.
11.x - probably needs new hierarchy soon.
Introducing "Instant-FAQ (TM)"
Added article section to complement book section.
Updated VC++ 2.2 information.

- 1.3 8/30/95

Updated from 1.2
Distribution for 1.2:
Instant-FAQ: 241
Web hits for Aug: 2966
ftp: ???
CIS: 684
Total circ: 3500-4000

- 4.0 1/15/96

Updated from 1.3
Converted over to Word format - using Word Internet Assist for html output.
Instant-FAQ: 700
Web hits: 5000/month+
Total circ: 10,000+
Coming soon: MSDN and 'Mastering VC++ CD-ROM' appearances of the MFC FAQ.

- 5.0 - 5/15/97

It's been over a year, I freshened up major sections most heavily in 2/3.
Added some cool stuff from .B ekiM.
Added many cool FAQs that were emailed or posted over the last couple of months.
Turning this beast over to Justin Rudd - the new MFC stud-puppy.
Instant-FAQ - 4000 (yikes!)
Ciao!

- 5.5 - 6/15/98

Yet, another year has passed and I, Michael Pickens, have taken over as the new MFC FAQ guy.
I've reformatted the old 'page' (if opened in MSWord it was a whopping 120 pages long!!) into 200+ pages for a new easier look-and-feel.
Added links, links, and more links for a much improved navigation.
Scoured the newsgroups for new FAQs to add.

- 5.6 - 7/01/98

- * Added two new menu items, "Toolbars, Statusbars, etc.." and "Menu's and Message Handling".

- * Changed section 2 from "Juicy MFC Information" to "MFC / VC++ Resource Guide."

- * Added more than 20 new FAQs!! Search for NEW!! and UPD!! for fast viewing.

- * Reformatted "MFC Tips, Tricks, & Caveats" from 30+ independant links to 5 sections. Much easier to view now.

13.5. Miscellaneous Continued (2/3)

13.5.5. How should I learn/start-learning MFC?

I see this posted lots. To start with, check out [section 2.4.2](#) of this FAQ, books always help. Another good point is that you can't really get down and dirty in MFC without knowing and understanding the underlying Windows API, so you might want to start with an overview of that. (The 'Petzold' book is the classic here) At the very least get a feel of what windows does and what it's capable of.

Another hint I have is don't rely too much on the Wizards, these are great, but they sometimes shield the newbie too much. Take a look at what AppWizard/ClassWizard has done for you, run the debugger through a mainfrm.cpp, etc.. You won't understand what the heck is going on until you understand what your magically generated code is doing.

I thought the 'writing windows apps with MFC' book was a good merge between Petzold and intro MFC stuff. 'Inside Visual C++' may move too fast for someone that doesn't know Windows already.

scot@stingray.com, 7/27/95

- ***There's no substitute for paying your dues.***
- ***Unfortunately MFC is set up so that you're never done paying your dues. (HAHAHAHA!!!)***

raymond@btw.com, mfc-l, 7/23/95

Start with the VC++ User's Guide and work your way through the Scribble Tutorial. You get an intro to VC++ and MFC and see how they work together. Then there are numerous articles on the MSDN CD to help with basic concepts. Try searching on MFC and Architecture. Don't shy away from reading the MFC Encyclopedia articles (they take more of a 'how-to' approach) and by all means become familiar with the MFC Technical Notes. Also (hint), stay focused on the 32-bit stuff if you're just starting out. There are also a number of excellent journals out there to help.

billb@microsoft.com, mfc-l, 7/23/95

The important points (of the top of my head) are:

If you're a windows programmer, don't worry about graphics. The CDC member functions are an almost direct encapsulation of the Windows functions. Also CWnd encapsulates much of the window manipulation functions.

Learn the Document-View architecture: document templates, the interaction between the application, the documents and views (AfxGetApp, GetDocument, UpdateAllViews etc).

Learn serialization (scribble makes a good intro to it). DDX and DDV for dialog boxes: there is a good tech note on this. WM_COMMAND handlers, COMMAND_UI handlers (which allow you to enable/disable menu items as well as applying check mark and radio button marks on menu items, it also gives you one way to use status bars).

During all this you'll also learn how to use ClassWizard and see how it is integrated with the editor and resource editor. Perhaps as part of the MFC FAQ we could create a list of topics (ordered in terms of importance) to learn? {Sure, this is a good start, no? -Scot}

grimes@parallax.co.uk, mfc-l, 7/24/95

See also: [Want to learn more about MFC?](#)

13.5.6. What's the best way to convert my C Windows App to MFC?

Microsoft and Symantec have jointly developed a tool that helps you make the move from a C Windows application to MFC. It's called the MFC migration kit. You can find it on the MSVC 2.x CD-ROM in the MFCKIT directory. The kit is also available from the Microsoft Software Library (See section 2.1.4) and from Symantec.

13.5.7. Why is my MFC application running slow?

MFC Apps should be nice and snappy. Make sure you are not building the debug

version and that you have the trace options off (MFCTRACE.EXE). If your application continues to be slow, try doing some quick profiling to see if you are making any redundant calls.

scot@stingray.com, 5/31/95

13.5.8. What is `afx.inl` and `afxwin1.inl`, etc..?

These files live in the `msvcXX\mfc\include` directory and include inline functions. These functions are only 'inline' during non-debug (`_DEBUG` is not defined) builds of MFC. They are prefaced with a special `_AFX_INLINE` directive which gets turned into 'inline' for non-debug builds and `"` for debug builds.

MFC does this so you can debug into the functions in debug mode, but get the benefits of inlining during release builds. That's one of the reasons the library shrinks so much in release build mode.

scot@stingray.com, 7/20/95

13.5.9. What the heck is this `_T()` thing I keep seeing?

`_T` is a macro to expand the string literal to support unicode.

Mike Oliver, MSMFC, 8/1/95

[BACK...](#) / [NEXT...](#)

13.5. Miscellaneous Continued (3/3)

13.5.10. How do I use CMemoryState?

In MFC 2.0 and above built in diagnostic facilities are provided. Hence it is not necessary to include CMemoryState explicitly in your app. The MFC version 2.0 debugging libraries automatically perform memory leak detection. The detection code is in AFXMEM.CPP. This code detects the case in which an application dynamically allocates an object and fails to delete the object before the program terminates.

In fact,CMemoryState may not function properly. You can refer to the following article in KB(Go MDKB in CIS) for more details:

ID:Q99022 "Foundation Class Debug Library Detects Memory Leaks"

#define new DEBUG_NEW should be defined in every .CPP source file. Also, insert the following code in InitInstance of the application:

```
#ifdef _DEBUG
    afxMemDF |= checkAlwaysMemDF;
#endif
```

You can refer to the following article in KB for more details:

ID: Q117326, Foundation Classes Common Asserts, Causes and Solutions.

Muniraju, netquest, MSMFC, 8/1/95

13.5.11. How do I customize the MFC idle time processing?

In my opinion, MFC's idle processing happens too often. Unfortunately, we can't change it because it might break apps when not necessary. But.. we have built in to the code a way for applications to customize the times in which they "enter idle". The secret? CWinThread::IsIdleMessage!!

Here's an example which disables idle entry for mouse & timer messages:

```

BOOL CMyApp::IsIdleMessage(MSG* pMsg)
{
    if (!CWinApp::IsIdleMessage(pMsg))
        return FALSE;
    if ((pMsg->message >= WM_MOUSEFIRST &&
        pMsg->message <= WM_MOUSELAST) ||
        (pMsg->message >= WM_NCMOUSEMOVE
        && pMsg->message <= WM_NCMBUTTONDBLCLK))
        return FALSE;
    if (pMsg->message == WM_TIMER)
        return FALSE;
    // otherwise msg must kick in idle processing...
    return TRUE;
}

```

Now, what would happen if you had a clock on your status bar, and were relying on a WM_TIMER to kick in a call to your update handler for that pane? Well... as you might have guessed, the above code would break you. There is an easy way to fix that, as there is a backdoor method of forcing an idle to kick in: WM_KICKIDLE defined in afxpriv.h can be used to make this happen.

So, in your timer handler (most likely a handler in your derived CStatusBar class), you would add some code which does:

```

if (TimeIsDifferentEnough()) PostMessage(WM_KICKIDLE);

```

If you take a look at CWinThread::IsIdleMessage, you'll see that we go through a fair amount of pane to avoid unnecessary OnIdle calls. Like checking for mouse moves to the same location as last time (I don't know why NT does that, but it does... every time the caret flashes). We are doing just about everything we can without breaking backward compatibility.

-deanm@microsoft.com, via email.

13.5.12. How do I display a choose directory dialog, instead of a choose file dialog?

```

/* Work's only if we're 95 capable */
if (afxData.bWin4)
{
    LPMALLOC pMalloc;

    /* Get's the Shell's default allocator */
    if (::SHGetMalloc(&pMalloc) == NOERROR)
    {

```

```

BROWSEINFO bi;
char pszBuffer[MAX_PATH];
LPITEMIDLIST pidl;

// Get help on BROWSEINFO struct - it's got all the bit settings
bi.hwndOwner = GetSafeHwnd();
bi.pidlRoot = NULL;
bi.pszDisplayName = pszBuffer;
bi.lpszTitle = _T("Select a Starting Directory");
bi.ulFlags = BIF_RETURNFSANCESTORS | BIF_RETURNONLYFSDIRS;
bi.lpfm = NULL; bi.lParam = 0;

// This next call issues the dialog box
if ((pidl = ::SHBrowseForFolder(&bi)) != NULL)
{
    if (::SHGetPathFromIDList(pidl, pszBuffer))
    {
        //At this point pszBuffer contains the selected path
        DoingSomethingUseful(pszBuffer);
    }

    // Free the PIDL allocated by SHBrowseForFolder
    pMalloc->Free(pidl);
}

// Release the shell's allocator
pMalloc->Release();
}
}

```

NOTE: This code will work on Win95 only - it's part of the shell.

bradw@netnet.net mfc-1, 9/9/95

13.5.13. I'm having problems using MFC 4.0 and the STL, what could be wrong?

The trick is to include "new.h" (and also "iostream.h" for similar reasons) before you include any stl headers. For example:

```

#include <new.h>
#include <iostream.h>

namespace std
{
    #include <map.h>
}

```


<dave_bickford@compuware.com>, mfc-l

NEW!! 13.5.14. What is MFC's Y2K compliance for CTimeSpan, COleDateTimeSpan, CTime, and COleDateTime?

If you read the online documentation you will see that all of these classes are Y2K compliant. The first problem for these classes will be in 2038 for CTime (underlying C runtimes fails) and 9999 for COleDateTime. The time span classes will fail by the fact that the date classes fail. If you are looking for date classes with an even larger range then have a look at DTime, a collection of MFC classes which I have developed. The URL is <http://indigo.ie/~pjn>.

The MFC documentation for CTime::CTime states that the valid range for the nYear parameter is 1970 to 2038 (actual range is midnight, January 1, 1970, to January 18, 19:14:07, 2038). These limits are based on the underlying time_t value that mktime creates from the struct tm. Most, if not all, ANSI C systems use 1970 as the reference date and overflow in 2038 (time_t is a long int that holds the number of seconds since the reference date). Microsoft TechNet <http://www.microsoft.com/ithome/topics/year2k/product/ctime.htm>

PJ Naughter, pjn@indigo.ie, mfc-l, 6/23/1998

[BACK...](#)
