

“本书是 MFC 开发人员的必备参考书籍”

——Rogue Wave 软件公司 Stingray 分公司 Scot Wingo

# MFC 经典问答

## The MFC Answer Book

Eugène Kain 著

健莲科技 译

Solutions for Effective Visual C++ Applications



本书附赠光盘一张



Addison-Wesley



中国电力出版社

www.infopower.com.cn

# MFC 经典问答

## The MFC Answer Book

Eugène Kain 著

健莲科技 译

Solutions for Effective Visual C++ Applications

中国电力出版社

## 内 容 提 要

本书收集了微软基础类库 (Microsoft Foundation Class, MFC) 编程中经常会遇到的问题, 多达 130 多条。本书设计独特, 采用提问—解答的方式, 对经常会遇到的所有问题进行了分类。本书共分为九章, 第 0 章对本书中的基本术语和一些约定进行了说明。第 1 章~第 8 章分别对不同类别的问题进行阐述。

本书适用于 MFC 高级开发程序员阅读使用, 对所有 MFC 编程相关的人员都具有很高的参考价值。

### 图书在版编目 (CIP) 数据

MFC 经典问答 / (美) 凯恩编著; 健莲科技译. —北京: 中国电力出版社, 2001

ISBN 7-5083-0606-6

I. M… II. ①凯… ②健… III. C 语言-程序设计-问答  
IV. TP312.44

中国版本图书馆 CIP 数据核字 (2001) 第 23504 号

著作权合同登记号 图字: 01-2000-3095

本书英文版原名: The MFC Answer Book: Solutions for Effective  
Visual C++ Applications

Published by Arrangement with Addison Wesley Longman, Inc.

All rights reserved.

本书由美国培生集团授权出版

中国电力出版社出版、发行

(北京三里河路 6 号 100044 <http://www.infopower.com.cn>)

三河市实验小学印刷厂印刷

各地新华书店经售

\*

2001 年 7 月第一版 2001 年 7 月北京第一次印刷

787 毫米×1092 毫米 16 开本 33.5 印张 760 千字

定价 59.00 元

版 权 所 有 翻 印 必 究

(本书如有印装质量问题, 我社发行部负责退换)

JS687/02

# 前 言

如果曾经查看过一个 Visual C++/MFC 新闻组或者邮件列表，那么可能会发现那里塞满了这样的问题：

- 怎样使分割窗口不能改变大小？
- 怎样把一个按钮或者一个组合放置到工具栏上？
- 怎样为对话框设置一种不同的背景颜色？

刚开始接触 MFC 的人可能认为这种产品的说明文档很差。但是，真正的问题是 Visual C++ 中的工具，诸如 AppWizard 和 ClassWizard 带给大多数开发人员一种安全错觉。一个初学 MFC 开发环境的开发人员运行 AppWizard、选择某些选项、在五分钟内完成一个工作应用程序是很典型的情况。MFC 开发环境“看”起来如此轻松！

在尝试进行某些操作如定制分割窗口的行为时，问题就产生了——如果没有任何帮助——那么很可能需要在繁多的 MFC 源代码堆中寻找几天，这些源代码装载在 Visual C++ 内。尤其是当在几分钟内就创建了大部分的应用程序时，上面的工作看起来不显得让人懊恼吗？

在这种情形下，许多人错误地抱怨 Visual C++ 的技术文档。MFC 和所有的 C++ 类库一样，设计得相当广泛（对黑客而言，Microsoft 为他们提供了源代码，这样他们能够使用 MFC 无限制的可扩展性）。不幸的是，Microsoft 的开发人员无法预测某个 MFC 用户可能需要怎样扩展该库来实现其应用程序；文档中只包括了 MFC 类的基本接口，这就要求勇敢的开发人员自己解决扩展的细节问题。

在图形用户接口日益复杂的今天，为了使基于 MFC 的应用程序具有竞争力，MFC 的可扩展性就显得非常重要了。也就是说，没有人再创建具有繁杂 Scribble 风格的应用程序。试着运行 Scribble 事例以及接下来的 Quicken、Microsoft Outlook、CorelDraw、Adobe Photoshop，或者从 1997 年开始发行的其他任何一种应用程序，这样会精确地看到未来几个月内潜在额外的 GUI 工作，为了跟上今天的用户接口标准，你需要在这方面投资。

从 MFC 的第一个版本开始，我就一直在使用 MFC 工作，并且开始接触 MFC FAQ 的联机文档，一学习更多有关 MFC 的知识，同时帮助人们更有效地使用这个类库。大约是在 3 年前，我开始编写 Stingray Software，其目的是开发更容易创建 MFC 应用程序的产品。George Shepherd 和我编写了视为珍宝的《MFC Internal》和《Inside Visual C++》两本书，第五版——出现在许多杂志的标题中——它是我们继续为这一目标而奋斗的标志。

在我使用 MFC 工作的这些年中，还没有发现哪一本书像你手里的这本书一样是必备的参考工具。通读这本书，就会得到明确的帮助，掌握更多有关 MFC 的知识，并且可以利用该类库进行非常有创造性的开发。我所希望的是，大多数读者深入阅读本书一星期后，将能够编写更为先进的应用程序 GUI，并且获得提升，还能都受到他们使用 MFC 开发程序的同伴的敬畏（这些只有当他们发现了你是如何创造出那些让人惊奇的增强效果时，才可能

发生)。

如果计划使用 MFC 开发所有重大的应用程序，那么收集本书中的 FAQ 肯定会使你节省花费在艰难地深究源代码上的几个月的时间。对于任何一个 MFC 开发人员来说，这本书都是一本“必备”的参考工具。

——Scot Wingo

MFC FAQ 的维护者

Rogue Wave Stingray Division 公司副总裁

《MFC Internal》和《Inside Visual C++》的合著者

本书的网址：

[http://www.stingray.com/mfc\\_faq](http://www.stingray.com/mfc_faq)

<http://www.stingray.com>

## 致 谢

在开始探索本书的主题前，请允许我占用一点时间来向许多人表达我的感激之情。在他们无私的奉献和努力下，本书才得以出版。

评论家们阅读本书的手稿图表，他们提出了改进意见，提高了本书的整体质量。我要感谢 Desiree Christensen、Charles Couchoud、Jack Mathews、Monika Weikel 和 Don Willits，他们为这本书投入了大量的时间和精力。

尤其要感谢我的朋友 Christophe Nasarre，他几乎对我手稿中的每一章都提出了深思熟虑的建议。可以肯定地说他是我遇到的最认真的审稿人！

我还要深深地感谢 Addison-Wesley 的小组成员，没有他们的帮助，这本书就不可能完成，现在也不会您的手中了。对 Ben Ryan 来说，他居住在另外的一个大洲，从来不认识我，但他非常相信我，甘愿冒为我出版该书的风险。后来，Ben 去开拓另外的事业，J.Carter Shanklin 和 Krysia Bebick 负责了本书完成的总监督。John Fuller 和他的同事把我的手稿翻译成了电子版以进行正确的排版和装订。复印编辑 Betsy Hardinger 为我的手稿做了非常优秀的编辑工作。Robin Bruce、Ben Ames 和 Chand Leary 负责营销，他们做得非常好。最后，Elizabeth Spinhour 高效地协调了这些人员之间的合作分工。

最后，非常重要的一点是，我要感谢我的妻子，她忍受了作为一个作家妻子的孤独，当我在计算机前度过夜晚和周末时，她慷慨地整理家务，并精心地照顾我们的两个孩子。同时也给我的父母和岳父母以同样的感谢，在完成本书的过程中，他们给了我无私的支持，还要感谢我妹妹的丈夫 Herve，他不时来探望我们，他的风趣幽默给我们带来了欢乐。

# 译 者 序

毫无疑问，Visual C++是目前最流行且功能强大的软件开发工具之一。其中的 MFC (Microsoft Foundation Class, 微软的基础类库) 已经成为程序人员的得力助手。但是由于 MFC 庞大的体系结构，程序人员在开发过程中常常会遇到这样或那样的问题，而这些问题又是多种多样且难于查找。

本书正是从这一点出发，收集了 MFC 编程中经常遇到的问题，多达 130 多条。本书采用了独特的提问—解答方式，对所有的问题进行分类，这样读者能够快速而准确地查找到所需要的每一个问题。

另外，本书中有大量的示例工程，包括源代码文件和可执行文件的清单。读者可以尝试使用这些示例工程去理解并掌握 MFC 编程的精髓。当然，也可以借鉴这些代码，改善自己的应用程序项目。

我们相信，本书将帮助你成为真正高效的 MFC 开发人员。

参加本书翻译工作的人员有：李兴娟、殷爱茹、郭玉红和苏永刚等。

因为本书涉及的知识和技术非常广泛，故难免存在一些翻译不当之处，欢迎读者批评指正。

译 者

# 序

## 为什么需要另外的 Microsoft Foundation Classes (MFC) 书籍来指导编程?

要回答这个为题, 让我们先来看一个典型的 MFC 编程情景。首先, 需要参加过一次 MFC 培训讲座, 或者阅读过一些介绍 MFC 编程的书籍。这样, 很快就能够书写或者定制小型的指导性应用程序。AppWizard 和 ClassWizard 使您达到一种空前强大的创造水平。应用程序支持多文档接口 (MDI) 并且具有专业外观型的用户接口, 如移动工具栏、状态栏、打印和打印预览等等。

接着, 就该转回来使用 MFC 产生界面友好的应用程序了。代码轻松地 从键盘间流出来, 向导勤奋地工作, 在 MFC 阳光的照射下显得如此美妙。

有一天, 您开始考虑如何实现新的特征, 而这些特征是培训讲座中没有具体讲解的。例如:

- 使应用程序保存上一次激活的文档, 自动地重新打开。
- 在同一个文档内支持多种类型的视图, 并且允许用户明确地打开任何一个视图。
- 在一个表单视图内增加工具提示 (ToolTip)。
- 动态切换到在窗口内显示的某个视图并且使其他的视图覆盖该视图。
- 实现一个扩展的对话框。
- 在另外的窗口如表单视图、对话框或者微型框架窗口内嵌入一个属性表页 (制表对话框)。
- 在状态栏面板内显示一个进度指示器。
- 当用户点击工具栏或者对话框上的某个按钮时弹出一个菜单。
- 在打印和打印预览中支持页眉和页脚。
- 利用进度指示器显示一个 Printing...对话框。

感觉起来实现这些功能没有什么难处: 毕竟, 您已经在其他的 Windows 应用程序中看到过它们。但是又将从哪里下手解决这些问题呢?

当您知道了产生预定效果所必须重载的特殊 MFC 虚函数, 或者知道了必须捕获并要适当处理的 Windows 消息的话, 这些问题可能会容易得到解决。然而, 对于某些特征, 需要涉及到更多的技术——甚至需要跟踪 MFC 的源代码来理解应该在什么地方进行什么操作来修改应用程序的默认行为。

让人懊恼的一件事是, 您要找的特殊问题的答案可能被搁置在某个地方, 也许埋藏在某本讲述 MFC 编程的书或杂志的文章中, 也许在 Microsoft Developer's Network 光盘中, 或者在 Microsoft Knowledge Base 中, 或者在从 Internet 上获得的各种线程和邮件列表中, 甚至在联机书籍或者 Visual C++ 光盘包括的例子中。问题是: 将如何在这些资源中查找最有用、最可靠的信息? 又将如何找到当前就需要的解决方案?



## 本书概述

编写本书的目的是提供随时可以使用的技术来解答 MFC 开发人员在现实中遇到的最普遍的问题。本书独特的结构设计使您能够迅速地查找到答案并且把相关的解决方案结合到程序中。

本书的 FAQ 采用一种理想的格式，这样开发人员能够迅速找到需要紧急解决的问题的答案。同时，您会发现许多技术可以帮助您更好地理解 MFC 应用程序的内部工作，并提高 MFC 编程技巧。尤其特殊的是，解释（Explanation）和附加注释（Additional Comment）部分通常都深入到 MFC 源代码或者没有归档的函数内部，从而解释所讨论的技术如何工作、它们与 MFC 的默认行为有何不同以及是如何与其默认行为结合起来的。

## 本书特点

尽管大部分描述 Visual C++ 和 MFC 编程的书籍都给出了有关 MFC 编程问题的正确答案，如果一页一页地阅读这些书，也会得到有用的提示。然而，其中绝大部分书籍的结构都无法迅速找到某个给定问题的答案。而且，即使找到了该答案，它很可能是隐藏在一大堆讨论之中，并不能立即给您一步步地技术指导，所以不能简单地把它们加入到当前项目中来增加所要求的特征。

相比之下，本书设计独特，它帮助 MFC 开发人员以最有效的途径解决他们在编程中遇到的问题：

- 本书具有组织化的体系，内容列表将帮助您迅速地找到解答所提问题的 FAQ。
- 我尽了一切努力来建立方便且容易理解的索引，它将指导您在本书中找到与某个关键字或者函数的相应页码。
- 所有 FAQ 的书写方式都很简洁，一开始就给出逐步的解答方法。解释和附加注释规则地放在稍后的部分，这样它们既不会妨碍解决方案，而且为那些不满足表面而希望进一步理解底层工作原理的人们提供了方便。
- 所有的解释都带有测试过的并可重用的实例代码，可以在几分钟内把这些代码插入到 MFC 应用程序中，很快地实现所要求的功能。

总结：本书旨在提供最短的途径来找到问题的逐步解决方案，这样可以立即把它们集成到当前项目中。

## 读者对象

本书适合的对象是所有希望解决有关 MFC 编程问题，尤其是要掌握高级 MFC 技术的 MFC 开发人员，本书有助于他们为其应用程序添加大量精巧复杂的特性。

本书假设读者掌握基本的 C++ 语言以及 MFC 编程技术，同时了解如何使用 Visual C++ 集成编程环境和工具，如 AppWizard 和 ClassWizard。只有为了特殊结果而以非标准的方式使用 Visual C++ 向导时，才对其进行讨论。

为了从本书获得最大的收益，你应该已经理解了在 Visual C++ 文档中描述的以 Scribble 指南表达的基本 MFC 概念：文档/视图结构、消息类型转换、UPDATA\_COMMAND\_UI 机制、对话数据交换（DDX）等等。比如说，你已经学习了 Scribble 教学指南，参加过 MFC

编程培训讲座，或者阅读过一本介绍该主题的有关书籍。

当然，拥有大量的 MFC 编程背景不会造成任何损失！而恰恰相反。根据读者和同事们的反馈，我发现本书还能帮助那些经验丰富的 MFC 开发人员，他们将在本书中找到许多有用的技术并把它们增加到他们的 MFC 编程工具包中。

总之，阅读本书将帮助所有的 MFC 开发人员加深对基础 MFC 概念的理解，并提高他们的 MFC 编程技巧。

## 如何使用本书

本书重点介绍在 Windows 95 和 Windows NT 下运行的 32 位 MFC 4.x 版本。然而，书中所讨论的大部分技术和概念也适用于 MFC 的早期版本。对更先进的 MFC 版本，这些技术和概念也应该是有用的，因为它们基于核心的 MFC 类和行为，并不会影响现有代码的发展演变。

我尽了最大的努力来写这本书，如果你愿意使用，它将成为你的很灵活的工具。这意味着，你可以一页一页地阅读本书——如果你这样做，我将非常感激——或者作为参考手册，只查阅感兴趣的特定主题。大部分 FAQ 都是交叉引用的，这样即使阅读到本书的中间部分，也可以帮助找到所需要的所有相关信息。

然而，我建议开始寻找 MFC 问题的答案之前，先花几分钟阅读第 0 章（术语和内容）和第 1 章（文档/视图结构的背景），以保证我们在术语和基础的文档/视图结构概念方面是约定好的。

## 光盘中的内容

光盘中包括源代码和本书中实例程序的可执行文件。文件夹的组织形式首先是章节号码，接着是项目名称。这样，第 2 章的 `AutoSaveDoc` 项目就位于 `d:\Chap02\AutoSaveDoc` 文件夹中，这里“d:”是光盘驱动器的盘符号。

所有的可执行文件都位于相应章节的文件夹中。例如，第 2 章中所有的可执行实例程序都位于 `d:\Chap02` 文件夹中。`EkUtil.h` 和 `EkUtil.cpp` 文件位于包括各种帮助 `Ek...` 函数和类的根目录下，这些函数和类出现在整本书中。

可以选择把光盘上的整个文件夹拷贝到硬盘上，也可以只拷贝感兴趣的实例，或者直接在光盘上访问这些文件。如果光盘上的文件拷贝到了硬盘上，则记着把硬盘上这些文件的只读属性去掉。

所有的实例程序都在 Visual C++ 5.0 和 Visual C++ 6.0 下编译并测试过。它们也可以在 Visual C++ 4.x 下运行，但是需要手工创建 `appropriate.mdp` 项目文件。注意，光盘上的 `.asp` 项目文件是 Visual C++ 5.0 的格式；如果使用 Visual C++ 6.0 打开这些文件，则当对话框询问是否把这些文件转换成新的格式时，只需回答 Yes。

光盘中还包括大量编译过的 Stingray Software 的 MFC（Microsoft Foundation Class）扩展库的实例，可以在 `Stingray Demos` 文件夹中找到这些文件。因为 Stingray 扩展库是专门为满足 MFC 开发人员的需求而设计的，所以对本书中提及的主题是一种理想的补充。

Stingray 扩展库提供了帮助类和完整的面向对象框架，它们显著地扩展了核心 MFC 类

的功能和容量。它们允许 MFC 开发人员创建具有复杂用户接口的短期应用程序，和那些当前成功的商业用产品一样。要得到更多的消息，请访问 Stingray Web 站点 <http://www.stingray.com>。

## 欢迎反馈

我尽最大的努力来精确地阐释那些我认为大部分 MFC 开发人员都感兴趣的主体。然而，如果您认为某个主题有不同的解释或者应该使用另外的技术，请给我发电子邮件 [ekain@awl.com](mailto:ekain@awl.com)。如果想要提出某种主题观点或者所遇到问题的自己的解决方案，或者发现了本书的错误以及其他任何问题，或者有某些建议要和我讨论，都欢迎给我发电子邮件。

我可以保证阅读所有的电子邮件，并仔细地考虑它们，尽可能早地回复每一封。但是，要注意，我可能没有时间回复具体的 MFC 编程问题。还可以访问我的 Web 站点 <http://www.mfcfaq.com> 来获得最新的资料。

我希望这本书能让您比以前更喜欢 MFC 编程，并帮助您完成宏伟的应用程序。

预祝您的 MFC 项目开发成功！

——EUGENE KAIN

# 目 录

前 言  
致 谢  
译者序  
序

第 0 章 术语和约定说明	1
第 1 节 本书中使用的术语	1
第 2 节 本书实例代码中的约定说明	2
第 1 章 文档/视图结构的背景	3
FAQ 1.1 什么是文档/视图结构?	3
FAQ 1.2 使用文档/视图结构的好处是什么?	3
FAQ 1.3 文档/视图结构的组件有哪些? 它们彼此间如何联系?	4
FAQ 1.4 文档的作用是什么?	7
FAQ 1.5 视图的作用是什么?	8
FAQ 1.6 视图框架窗口的作用是什么?	9
FAQ 1.7 文档模板的作用是什么?	9
FAQ 1.8 文档模板资源 ID 的作用是什么?	10
FAQ 1.9 如何创建文档/视图结构的组件对象?	12
FAQ 1.10 MFC 如何发送命令消息?	13
FAQ 1.11 MFC 如何执行标准的菜单命令?	16
第 2 章 文档和文档模板	20
第 1 节 管理文档模板	20
FAQ 2.1 在应用程序中如何创建并引用多文档模板?	20
FAQ 2.2 当用户每次创建一个新文档时, 如何管理多文档模板而不用弹出对话框?	23
FAQ 2.3 如何创建具有一个新的文档模板资源 ID 的资源?	28
FAQ 2.4 当打开一个文档时, MFC 如何选择所使用的文档模板对象?	30
FAQ 2.5 如何把多个文档扩展名与同一个文档或视图类关联?	34
第 2 节 管理文件	38
FAQ 2.6 如何防止一个 MDI 应用程序在启动时打开一个新(空)文档?	38

FAQ 2.7	如何编程来创建一个新(空)文档? .....	43
FAQ 2.8	如何编程来打开一个已存在的文档文件? .....	47
FAQ 2.9	如何使应用程序保存上一次的活动文档,并自动地再次打开它? .....	48
FAQ 2.10	如何使文档自动地保存而不用提示用户? .....	53
FAQ 2.11	如何实现“Save all documents”菜单命令,该命令在保存每个修改过的文档时不用提示用户? .....	54
FAQ 2.12	如何编程来关闭一个文档? .....	59
FAQ 2.13	如何实现“Close all documents”菜单命令? .....	61
第3节	管理当前文档列表(Recent Files List, MRU) .....	66
FAQ 2.14	如何定制在“File”菜单内显示的当前文档数? .....	66
FAQ 2.15	如何编程在Recent Files表中增加一个特殊的字符串? .....	70
FAQ 2.16	如何截取用户选择的Recent Files项? .....	73
FAQ 2.17	如何使应用程序自动地打开在Recent File表顶部列出的文档? .....	76
第4节	其他条目 .....	77
FAQ 2.18	如何得到指向当前活动文档的指针? .....	77
FAQ 2.19	如何遍历当前打开的所有文档列表? .....	81
FAQ 2.20	为什么我的应用程序没有在Windows Explorer中注册它的文档文件,如何解决这一问题? .....	87
第3章	视图和框架窗口 .....	90
第1节	综合主题 .....	90
FAQ 3.1	如何在我的应用程序中增加一个新类型的视图或框架窗口? .....	90
FAQ 3.2	如何选择是在视图类还是在视图的框架窗口类内实现某种特殊行为? .....	95
第2节	打开和关闭视图以及框架窗口 .....	96
FAQ 3.3	如何编程打开一个基于特殊文档模板的视图? .....	96
FAQ 3.4	如何在每次创建新文档时打开两个(或多个)特定视图? .....	99
FAQ 3.5	如何编程来关闭一个视图? .....	104
FAQ 3.6	如何编程来关闭一个打开文档的所有视图? .....	104
FAQ 3.7	如何防止用户关闭一个视图? .....	105
第3节	管理位置和大小 .....	110
FAQ 3.8	如何编程来改变视图的大小和位置? .....	110
FAQ 3.9	如何设置一个视图的初始位置和大小? .....	112
FAQ 3.10	如何使主应用程序窗口居中? .....	115
FAQ 3.11	如何使视图居中? .....	118
FAQ 3.12	如何使框架窗口的大小不可改变? .....	120
FAQ 3.13	如何限制视图的最大或最小尺寸? .....	121
FAQ 3.14	如何限制视图的重新定位? .....	124
FAQ 3.15	如何使视图初始显示的规模为最小或最大? .....	128
FAQ 3.16	如何使应用程序以最大化或者最小化状态启动? .....	131

FAQ 3.17	如何使主应用程序窗口总是位于其他窗口的上面? .....	135
FAQ 3.18	如何使某个视图总位于其他视图的上面? .....	136
FAQ 3.19	如何实现一个像 Visual C++和 Word 那样的全屏视图模式? .....	140
第 4 节	管理标题、图标、光标和背景.....	146
FAQ 3.20	如何定制视图的标题? .....	146
FAQ 3.21	如何在视图的标题中显示一个“修改过”的指示符, 与某个“修改过的”文档关联? .....	151
FAQ 3.22	在应用程序中如何修改与各个类型的窗口相关联的图标? .....	154
FAQ 3.23	如何动态地改变主应用程序窗口的图标? .....	155
FAQ 3.24	如何动态地改变某个视图的光标? .....	158
FAQ 3.25	如何在某个漫长的操作中显示一个沙漏光标? .....	161
FAQ 3.26	如何改变一个视图的背景颜色? .....	163
FAQ 3.27	如何在主应用程序窗口的背景上画图? .....	165
第 5 节	程序清单视图 .....	172
FAQ 3.28	如何使我的程序清单视图与其他的视图同步? .....	172
FAQ 3.29	如何使一个程序清单视图初始显示时的尺寸恰好为所关联的对话框资源的大小? .....	175
FAQ 3.30	如何在程序清单视图内使用 UPDATE_COMMAND_UI 机制? .....	176
FAQ 3.31	如何改变程序清单视图的背景颜色? .....	180
FAQ 3.32	如何在程序清单视图的控件上增加工具提示 (ToolTip)? .....	183
第 6 节	分割窗口 .....	186
FAQ 3.33	如何编写一个具有水平和垂直静态分割器面板 (三线分割) 的窗口? .....	186
FAQ 3.34	如何锁定一个分割器 (Splitter) 窗口, 从而使用户不能移动分割线? .....	190
FAQ 3.35	如何编程改变某个分割器窗口内面板的大小? .....	192
FAQ 3.36	如何向用户可见地显示包括活动视图的分割器面板? .....	195
第 7 节	切换视图 .....	199
FAQ 3.37	如何动态地切换在 MDI 子窗口或者在 SDI 主应用程序窗口显示的视图? .....	199
FAQ 3.38	如何动态地切换在分割器窗口内显示的视图? .....	203
第 8 节	其他条目 .....	208
FAQ 3.39	如何得到活动视图的指针? .....	208
FAQ 3.40	如何得到活动框架窗口的指针? .....	210
FAQ 3.41	如何遍历与某个文档关联的所有视图程序清单? .....	211
第 4 章	对话框.....	213
第 1 节	综合主题 .....	213
FAQ 4.1	如何设置一个对话框的起始位置? .....	213
FAQ 4.2	如何使对话框在另一个窗口里居中? .....	217
FAQ 4.3	对话框如何访问活动文档 (或视图) 对象? .....	220
FAQ 4.4	如何控制对话框的背景颜色? .....	223

FAQ 4.5	如何给一个对话框加入预览区域（在对话框内绘制）？	226
FAQ 4.6	如何为对话框增加一个图标？	229
FAQ 4.7	如何实现一个展开的对话框？	230
FAQ 4.8	如何为对话框增加一个工具栏？	237
第 2 节	处理对话框内的控件	248
FAQ 4.9	当显示一个对话框时，如何选择最初拥有焦点的控件？	248
FAQ 4.10	如何在对话框中定制确认规则？	248
FAQ 4.11	如何改变对话框内的控件的字体和颜色？	251
FAQ 4.12	如何在对话框中使用 UPDATE_COMMAND_UI 机制？	254
FAQ 4.13	如何在对话框中增加工具提示以支持控件？	259
<b>第 5 章</b>	<b>属性单</b>	<b>262</b>
第 1 节	综合主题	262
FAQ 5.1	如何建立和使用一个属性单？	262
FAQ 5.2	如何设置属性单的 Apply 按钮？	267
FAQ 5.3	如何通过编程改变属性单内的活动页？	272
FAQ 5.4	如何定制标准的属性单按钮？	274
FAQ 5.5	如何控制属性单窗口的尺寸？	279
第 2 节	管理标签 (Tab)	283
FAQ 5.6	是选择标签的层叠行方式还是选择标签的带有滚动条单一行的方式？	283
FAQ 5.7	如何改变属性单标签的标题？	284
FAQ 5.8	如何为属性页的标签 (Tab) 加入图标？	287
第 3 节	嵌入属性单	288
FAQ 5.9	如何在对话框里嵌入属性单？	288
FAQ 5.10	如何在表单视图里嵌入属性单？	294
FAQ 5.11	如何在分割器窗口里嵌入属性单？	297
FAQ 5.12	如何在小框架窗口里嵌入属性单？	299
<b>第 6 章</b>	<b>工具栏和状态栏</b>	<b>302</b>
第 1 节	工具栏	302
FAQ 6.1	如何在主框架窗口或视图的框架窗口加入一个或多个工具栏？	302
FAQ 6.2	用于控制工具栏位置、方位以及一般行为的各种操作是什么？	308
FAQ 6.3	如何通过编程来实现将一个工具栏紧随另一个工具栏停靠？	313
FAQ 6.4	如何为工具栏添加一个组合框？	316
FAQ 6.5	如何为工具栏按钮添加文本标签？	324
FAQ 6.6	如何实现用户使用菜单选择所要的工具栏？	327
FAQ 6.7	如何通过选择右击停靠区而显示的弹出式菜单，允许用户选择所要的工具栏？	332
FAQ 6.8	如何动态地在不同的工具栏之间切换？	334

第 2 节 状态栏 .....	342
FAQ 6.9 如何为视图的框架窗口添加一个定制的状态栏? .....	342
FAQ 6.10 如何更新状态栏格的文本? .....	344
FAQ 6.11 如何定制属性条的字体? .....	350
FAQ 6.12 如何在状态栏内绘画? .....	353
FAQ 6.13 如何在显示菜单提示时最大化消息格? .....	358
FAQ 6.14 如何在状态栏内显示一个进度指示器? .....	365
第 3 节 常见控制栏问题 .....	370
FAQ 6.15 如何编程显示或隐藏控制栏? .....	370
FAQ 6.16 如何设置浮动控制栏的标题? .....	373
FAQ 6.17 如何保存和恢复控制栏的位置和状态? .....	375
FAQ 6.18 如何创建自定义控制栏? .....	378
第 7 章 菜单 .....	398
FAQ 7.1 如何用一个单一函数处理几个不同的菜单命令? .....	398
FAQ 7.2 如何实现一个右键弹出式菜单(上下文菜单)? .....	403
FAQ 7.3 如何在当用户在一个工具栏上单击一个按钮时显示一个弹出式菜单? .....	406
FAQ 7.4 如何在当用户在一个对话框中单击鼠标右键时显示一个弹出式菜单? .....	409
FAQ 7.5 如何实现一个属主绘式菜单? .....	415
第 8 章 打印和打印预览 .....	428
FAQ 8.1 MFC 的打印体系结构是怎样的? .....	428
FAQ 8.2 如何在 MFC 应用程序中实现打印和打印预览? .....	434
FAQ 8.3 如何打印页数事先未知的文档(动态分页)? .....	442
FAQ 8.4 如何像 Microsoft Word 那样预测分页符? .....	447
FAQ 8.5 如何编程动态实现打印方向(横排或竖排)的改变? .....	449
FAQ 8.6 如何不向用户显示标准的打印对话框而直接打印? .....	451
FAQ 8.7 如何自定义标准的打印对话框并获取用户选项? .....	455
FAQ 8.8 如何自定义“打印...”对话框? .....	459
FAQ 8.9 如何停止或中断打印作业? .....	466
FAQ 8.10 如何为单色打印机实现灰度打印预览? .....	467
FAQ 8.11 如何在代码中区分画图、打印预览和实际打印? .....	469
附录 A 公用函数和类 .....	472
附录 B 参考文献和其他资源 .....	516



## 第 0 章 术语和约定说明

每一个技术领域都要制定其使用的特殊行话，MFC 编程也不例外。尽管这些术语有助于人们书写简练而精确的说明，但是，如果某个特定的词汇或说明存在一些歧义的话，也会带来不必要的麻烦。

因此，我建议在阅读后面的章节之前，先花几分钟的时间读一下接下来的内容，以确保我们对本书中使用的术语和编码约定是一致的。如果某些定义看起来非常简短，也不用担心：当它们用在某个特定技术的上下文中时，就能很容易地理解它们并和所需要的联系起来。因此，当阅读后面章节中的 FAQ 时，没有必要再回到这一部分。

### 第 1 节 本书中使用的术语

**应用程序类：**CWinApp 的派生类，它使用 InitInstance() 函数来完成应用程序的初始化代码。

**应用程序对象：**应用程序类在全局中惟一的实例。

**子框架类：**项目中的一个 CMDIChildWnd 派生类，它和一个视图的框架窗口相联系，在 MDI 应用程序中是必需的（在一个 SDI 应用程序中，主应用程序窗口同时也是视图的框架窗口）。

**子框架窗口：**见“视图的框架窗口。”

**文档类：**项目中的一个 CDocument 派生类。

**文档对象：**某个文档类的一个实例。

**文档模板对象：**CMultiDoc Template 或者 CSingleDoc Template 类的一个实例。

**文档模板资源 ID：**资源 ID 把一个文档模板对象和三种资源联系起来：菜单资源、图标资源和复杂的字符串资源（文档模板字符串）。在“FAQ 1.8 文档模板资源 ID 的作用是什么”中讨论了这些主题。

**文档模板字符串：**由文档模板资源 ID 说明的字符串资源——它定义了 7 个与文档模板对象相关联的子字符串。在“FAQ 1.8 文档模板资源 ID 的作用是什么”中讨论了该主题。

**文档窗口：**泛指用户想要作为一个“视图”观察的文档内容，当我们谈论的是子框架窗口或者实际的视图窗口时，不必做特殊说明。

**主应用程序窗口：**应用程序位于顶层的 Windows 窗口。该窗口通常和一个 C++ 对象联系在一起，这个对象是 CMDIFrameWnd（对 MDI 应用程序而言）或者 CFrameWnd（对 SDI 应用程序而言）派生类的一个实例。

**主框架窗口类：**CMDIFramWnd 或者 CFrameWnd 的派生类，管理应用程序的主框架窗口。

**主框架窗口对象：**应用程序中与主框架窗口有关的 C++ 对象，它是惟一的。

**视图类：**项目中 CView 的一个派生类。