

How to Program Visual Basic 5.0



- 在 Web 上或在客户应用程序中创建、集成及重用组件
- 了解 VB5.0 的新特征及其增强的开发环境
- 使用 ActiveX 控件开发功能齐全、加载迅速的 Web 页面

使用

[美] Jeffrey P. McManus 著
龚杰 秦颂 蔡进 译

Visual Basic 5.0

编程



電子工業出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
URL: <http://www.phei.com.cn>

使用 Visual Basic 5.0 编程

How to Program Visual Basic 5.0

[美] Jeffrey P. McManus 著

龚杰 秦颂 蔡进 译



电子工业出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY

内 容 简 介

本书是一本使用Active X技术进行32位分布式开发的综合性指导书,它是为对此感兴趣的中级至高级Visual Basic程序员设计的。本书为你展示了对象技术及控件技术,你将从中学到如何开发简单的控件并遵循书中的步骤将你感兴趣的内容加入IE或其它ActiveX容器中。

本书编排合理,思路清晰,图文并茂,易学易懂。本书共十七章,包括VB5.0的新内容,控件属性,控件方法,控件发布等内容。



Authorized translation from the English language edition published by Ziff-Davis press an imprint of Macmillan Computer Publishing U.S.A. Copyright© 1997.

All rights reserved. No part of this book may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording or by any information storage retrieval system, without permission from the publisher.

SIMPLIFIED CHINESE language edition published by Publishing House of Electronics Industry. Copyright © 1998.

本书英文版由美国麦克米兰计算机出版公司的Ziff-Davis公司出版,已将中文版版权授予电子工业出版社。未经许可,不得以任何形式和手段复制或抄袭本书内容。版权所有,侵权必究。

原 书 名: How to Program Visual Basic 5.0

书 名: 使用Visual Basic 5.0编程

著 者: [美] Jeffrey P. McManus

译 者: 龚杰 秦颂 蔡进

责任编辑: 范官清

印 刷 者: 天津颖华印刷厂印刷

出版发行: 电子工业出版社出版、发行

北京市海淀区万寿路173信箱 邮编100036 发行部电话 68214070

URL: <http://www.phei.com.cn>

经 销: 各地新华书店经销

开 本: 787×1092 1/16 印张: 16.25 字数: 400千字

版 次: 1998年3月第1版 1998年3月第1次印刷

印 数: 6000册

书 号: ISBN 7-5053-4060-3
TP·1774

定 价: 26.00元

著作权合同登记号 图字: 01-97-0995

凡购买电子工业出版社的图书,如有缺页、倒页、脱页者,本社发行部负责调换

版权所有·翻印必究

引言

尽管每个软件模块都可以与其它软件模块相互交互的日子不会很快到来，但我们都期待着这一天的早日到来。我个人希望有一天可以把软件芯片放置在家庭宠物（比如，一只小猫）的身体里，然后利用Visual Basic的特性和方法来控制它，那样该有多好啊。

不过，我的奢望也许太高了。

自从Visual Basic发布后，Windows开发人员就具备了使用软件组件（component）的能力。业内人士告诉我们，过去仅仅面向VB开发人员的Windows组件市场现在正在向全然不同的应用领域扩展，如FoxPro、Access、Visual C++和Microsoft Word 97等，它已形成了一个相当大的软件产业，其产值高达2亿5千万美元。它们是一些人人都可重复利用的组件。

怎样对付不合作的用户

随着因特网的出现，组件技术肯定将不断发展、成熟并最终成为每个应用、Web页面、旅店里的摇摆床以及家庭宠物的一部分。

然而，这一切还没有实现。尽管如此，通过Web页向用户发送组件的想法依然是很诱人的——它只要求你的用户所使用的浏览器支持你选定的软件组件技术标准。

当然，如果访问你的Web页的参观者没有使用“正确的浏览器”，你可以采用以下方法来对付他们：

- ▶ 弄死他们
- ▶ 用砂纸打磨他们，直到他们难受得狂叫为止
- ▶ 其它各种方法

只有通过全球Web管理员的通力合作，全球性的ActiveX同化才能得以迅速、平稳地实现。

在Web浏览器普遍支持ActiveX之前，我们都有充分的理由来开发ActiveX控件。你或许对这些没有多少印象，也可能你不打算购买本书（并且我也可能不写它）。但我还是认为在以下两个领域里，ActiveX控件的开发将最为流行：纵向应用领域（用Visual Basic或任何一种支持ActiveX控件的其它开发环境编写）和专用Web服务器领域（也被叫做企业内部网）。

了解Microsoft Office 97对ActiveX的支持是非常有趣的，虽然我编写本

书，但我却无法告诉你它是如何实现的，因为Office 97此时尚未发布。

Visual Basic是创建控件的最佳环境

从理论上讲，你可以用任何一种编程语言编写并调用ActiveX组件，这是ActiveX最重要的特性之一，这一点与其它跨平台组件技术（比如Java）不同。

VB程序员不再需要依赖第三方去使用C++编写的控件。事实上，VB作为一种控件开发环境的出现意味着Visual Basic开发人员现在可以转换表格，为用Visual C++开发应用程序的开发人员提供ActiveX组件。

随着时间的推移，一定会有越来越多的语言和开发环境支持ActiveX控件的创建。事实上，在本书写作期间，InterGroup Technologies公司已经发布了创建ActiveX控件的新工具——VisualX的beta版本。它虽然没有Visual Basic的Control Creation Edition好，但它还是值得一提的。你可以从<http://www.intergrouptech.com/form.html>处免费下载它。

Johnny无法创建控件的原因

多年来，Visual Basic程序员一直在抱怨，说他们无法在Visual Basic中创建ActiveX控件。他们总是问，“要VB提供这种能力到底有多大困难？”。现在，一切都解决了，微软除了给你提供了在Visual Basic中创建控件的方法外，还允许你从微软的Web网站上免费下载它。

因为用Visual Basic创建控件是如此容易（容易是指一旦你购买了该产品并阅读了有关信息，你就能创建控件），我们很可能在今后几年里看到许多与ActiveX控件有关的新的、有趣的、迷人的产品。

好啦，现在你已经有了Visual Basic Control Creation Edition（即本书）。只是希望你要利用它做好事而不是干坏事啊。

本书不讨论的内容

现在，你对本书将要讨论的内容已经有了一个初步的概念，我要谈谈本书不讨论的问题。（顺便提一下，我最讨厌作者说，“噢，这个问题超出了本书的范围”，这好像是在逃避，因此，如果有些专题超出了本书的范围，我将给出深入讨论该专题的参考资料。）

除了直接与在Visual Basic中创建控件有关的地方外，本书将不讨论支持ActiveX控件的背景技术，如COM、DCOM和OLE。我在编写本书时就考虑了这一点，你不需要了解这些初级问题，就可以创建很好的控件，而不像创建好的Visual Basic应用就必须了解有关Windows API的方方面面。如果你还想了解

更多的信息，我建议你查阅ActiveX 软件开发工具包中的文档与工具，你可以从以下网站上下载它们：<http://www.microsoft.com/activex>。

你还将注意到，我在本书中有意避免提到Java。这并不是因为我无法对ActiveX和Java进行评价；而是因为我不想对此妄加评论。关于ActiveX与Java的优劣之比较，我的意见如下：ActiveX与Java在因特网上的竞争实在难分伯仲。

如欲得到免费的Visual Basic 5.0 Control Creation Edition，请访问微软的Web网点，地址为<http://www.microsoft.com/vbasic/controls/>。

本书读者对象

本书是专为那些对在Visual Basic中创建ActiveX控件感兴趣的中高级Visual Basic程序员设计的。你也许对创建驻留在Web页面中的ActiveX控件感兴趣，你也可能对创建驻留在编译后的Visual Basic应用中的ActiveX控件感兴趣。不管是哪种情况，本书都介绍了在VB中创建一个控件时，你需要采取的全部步骤。

尽管本书使用了Control Edition来创建书中的全部例子，但你实际上可以使用Visual Basic 5.0的任何一种版本来创建控件。无论哪种版本，其基本概念都是一样的。

请注意，本书并不是一本完整的Visual Basic编程手册，然而，它可以作为通用的VB编程手册的补充材料。如果你是一位初学者，并且你想学习怎样创建ActiveX控件，那么在你创建控件之前有必要了解VB的工作方式。这样，你可以节省时间并减少麻烦。

最后，我预祝你学习愉快。请不要忘了经常去访问<http://www.well.com/user/jeffreyp/activex/>网点。我会及时更新该网点，使它能够提供尽量多的控件创建信息并且提供尽可能多的指向VB资源的链接。

目 录

引言

第1章 Visual Basic 5.0中的新内容	1
在这个集成开发环境中有哪些新内容	1
新的语言特性	10
第2章 控件基础知识	14
ActiveX控件概述	14
控件接口	18
第3章 开发你的第一个控件	21
起步	21
项目管理	22
控件设计	23
控件测试	24
剖析HappyHour控件的Caption属性	26
利用ActiveX 控件接口向导设置属性	32
第4章 控件属性	36
创建属性过程	36
创建定制的属性页	51
第5章 处理和引发事件	58
创建定制的事件	58
UserControl对象的事件	66
第6章 控件方法	68
创建一个方法	68
创建一个带参量的方法	71
创建带返回值的方法	72

标准方法的实现	75
使用容器提供的方法	75
第7章 使用容器	77
容器概述	77
使用Extender对象	78
第8章 使用结构控件	87
固有控件	87
对控件中的所有结构控件进行操作	88
第三厂家的控件	89
找寻商用控件	90
第9章 用户绘制的控件	95
图形方法	95
Print事件	100
用户绘制的控件: Hexagon Control	101
Refresh方法	102
以禁止方式显示控件	103
显示Default属性	104
表明你的控件是聚焦的	106
第10章 窗口装饰	107
给控件加上透明背景	107
设计时、运行时及只读属性	107
利用控件充当容器	109
访问键的实现	112
运行时不可见的控件	115
与底表边缘对齐的控件	116
增加一个About框	118
项目模板	120

增加一个工具箱位图	123
第11章 建立Windows API和DLL调用	124
DLL调用说明	124
项目举例: SoundButton控件	125
项目举例: MenuPic控件	130
第12章 控件发布	137
控件编译	137
二进制兼容性	138
给控件命名	139
项目属性	139
在Microsoft Access下测试编译后的控件	142
使用Setup向导	144
二进制许可授权	146
本地化	147
为控件登记和取消登记	148
使用Regclean	148
第13章 在Web上部署控件	150
拥有Web网点的用户	150
对于不上网的用户	150
获取ActiveX Control Pad	151
HTML和VBScript强化教程	151
使用ActiveX Control Pad	152
在Web浏览器中测试你的控件	154
使它与Netscape一起工作	158
在Internet上使用Setup 向导	159
CAB文件和CODEBASE标记	160
安全性考虑	162

Web上的许可证发放问题	163
签署你的代码	165
使用Signcode签署你的.CAB文件	166
第14章 与因特网交互的控件	169
属性值的异步下载	169
作为结构控件的因特网控件	176
在控件中增添Web浏览功能	183
第15章 程序调试与错误陷阱设置	193
程序调试	193
错误处理	202
第16章 面向对象编程	209
面向对象的基本原理	209
类	209
集合	213
将类结合到控件中	218
利用Implements（实施）关键字建立接口	220
使用Friend（友好）方法	223
第17章 数据库访问	226
Visual Basic的数据访问	226
在ActiveX控件中增加数据认知功能	230
词汇表	235

第1章 Visual Basic 5.0中的新增内容

本章描述 Visual Basic 5.0中重要的新的特性，包括界面的改变和Visual Basic (VB) 编程语言的改变。

乍一瞥，界面的改变似乎不尽人意。然而幸运的是，Visual Basic 5.0的设计人员已经为你提供了许多崭新的可定制的特性。你可以定制界面，使它看起来像Visual Basic 4.0的界面，随后在需要新特性时再增加它们，直至使这个界面变得面目全非。这一切全由你决定。

在这个集成开发环境中有什么新内容

当你开发一个Visual Basic项目时，你将在IDE中消磨大部分时间。你需要在IDE中编写代码、给项目增加组件（component）并为你的应用程序设计可视化界面。

Visual Basic 5.0给IDE增加了许多新特性，使用户不必多费神就能更容易、更快速地开发应用程序。

■ 项目、文件和项目管理器（Project Explorer）

项目管理器是你管理组成Visual Basic 5.0项目的文件的地方；它类似于Visual Basic早期版本中的Project窗口。

Visual Basic 5.0中之所以需要项目管理器，其主要原因是因为你现在可以同时打开多个项目。这对于Visual Basic行家来说可能是多余的，但是一旦你了解了控件的创建过程，你就会明白其存在的意义。由于EXE项目和ActiveX项目是彼此独立的，并且你需要用EXE项目来测试你创建的ActiveX控件，因此你很可能经常要用到多个项目。

有了这些新文件和多个项目，要想随时追踪每一项任务就变得很困难了。图1.1显示了文件和文件名后缀的层次结构。

请注意，ActiveX项目文件和EXE项目文件都是以.VBP结尾的。这意味着你可能需要给你的项目文件增加某些其它识别特征以使它们能相互区别。例如，你可能想用含有单词“Control”的文件名表示包含ActiveX控件的项目文件。

例如，假设你有一个叫做StickShift的项目。这个项目是为一个图形推送按钮建立的ActiveX控件，该按钮的形状很像1989年产的五速“牧马人”吉普车的手动变速器。对这个项目，你至少需要一个描述StickShift控件本身的文件（StickShift.ctl）和一个项目文件（StickShiftControl.vbp）。如果StickShift有一个定制的属性表单（property sheet），那么你可以给该文件取名为StickShift.pag。

为了测试你的ActiveX控件项目，你几乎总要用到EXE项目。对这个StickShift项目，这意味着你要增加一个项目文件（TestStickShift.vbp）和一个底表（TestStickShift.frm）。最后，你的项目还需要一个Project组文件StickShift.vpg，用来容纳EXE项目和ActiveX项目。

提示：最好把你创建的每个项目存入它自己的文件夹，以便把你需要的全部文件存储在一个地方。本书附带的CD-ROM上的项目就是以这种方式组织的。

单文档/多文档界面选项

当你首次启动Visual Basic 5.0时，你注意到的第一件事可能是窗口和菜单的布局有所改变。Visual Basic集成式开发环境（IDE）现在有一个多文档界面（MDI）。MDI的优点是：所有的子窗口（有许多这样的窗口）都出现在背景上的一个大窗口中（这个大窗口被称为父窗口）。

MDI的缺点是移动和放大、缩小窗口的灵活性不够。此外，如果你多年来一直使用Visual Basic，则你可能会觉得这个新界面有点不和谐。幸运的是，有一些方法可以使Visual Basic IDE的外观看起来像其早期版本的外观，即被称为做单文档界面（SDI，Single Document Interface）的式样。为了把Visual Basic变成SDI界面，你可以按如下步骤操作：

- 1 在该菜单中选定Tools，然后选Options。
- 2 Options对话框出现。选定Advanced标签。
- 3 选取标有SDI Development Environment的小框。
- 4 单击OK。

完成这些步骤之后，你必须重新启动Visual Basic，但相对于回到我们多年来一直熟悉和喜爱的Visual Basic界面来说，这种代价是微不足道的。

注 本书中的所有实例都假定你正在使用SDI界面。这两种界面的过程指令稍有不同。即使你坚持使用新的MDI界面，你也能推断出要做什么，但是对某些

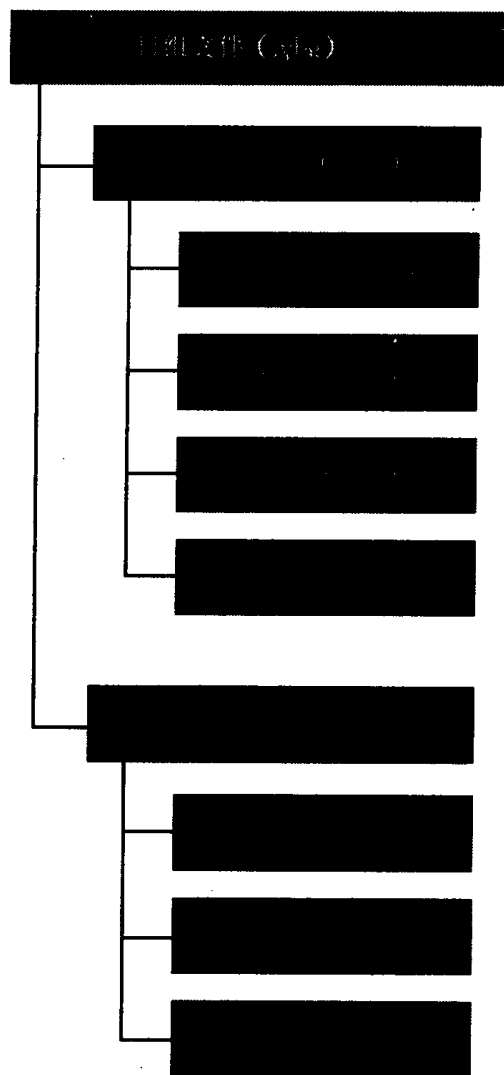


图1.1 Visual Basic 5.0中文件和项目的层次结构

任务，比如关闭和调整窗口尺寸，SDI界面使用起来会更容易些。

Form Layout（底表布局）窗口

Form Layout窗口恰好位于屏幕右边Properties窗口之下。它使你通过在屏幕上直观地定位底表来随时了解运行时底表在屏幕上出现的位置。现在，如果想要使底表在屏幕上居中，你再也不必编写令人心烦的程序代码，你可以利用Form Layout窗口来完成此类任务。

通过右击Form Layout窗口，你可以显示分辨率数值。它将向你展示你的表格布局在分辨率为640 x 480的标准VGA屏幕上的外观。这样做是很有意义的，因为许多开发人员已经在使用高分辨率的屏幕了，也有一些用户还仍然沉溺于低分辨率的世界之中。图1.2显示了Form Layout窗口在屏幕上的位置。

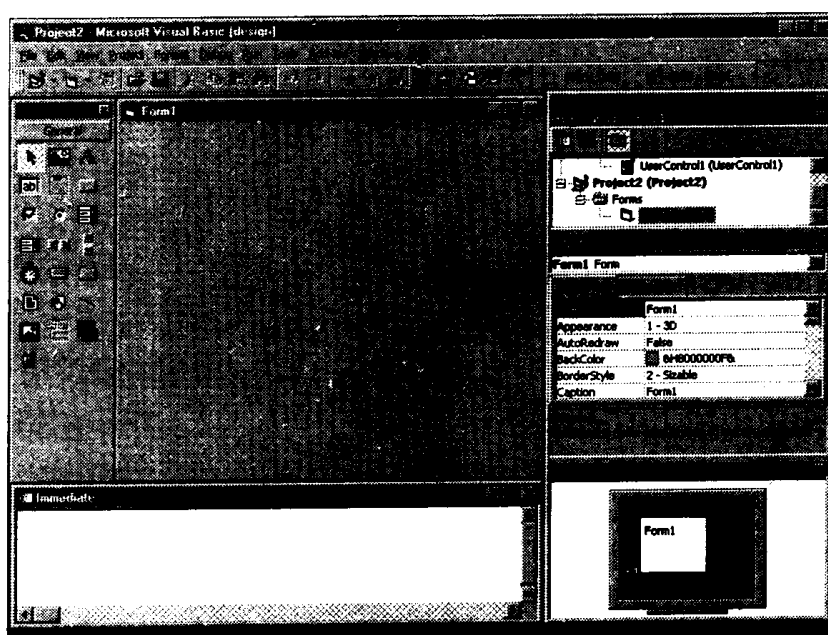


图1.2 Form Layout窗口

通过右击Form Layout窗口并选定Startup Position，你可以指定在你的应用程序启动时底表应该定位在何处。（要实际地看到这种效果，你需要在你的项目前景中设置一个底表）。

你还可以把底表的Startup Position属性设置成2-Center Screen（这是针对Visual Basic 5.0表格的新属性），从而使底表在屏幕上居中。

当你用Visual Basic 5.0建立控件时，你可能不想太多地使用Form Layout窗口，除非你的控件需要显示充当定制对话框的底表。在这种情况下，你应该使用Form Layout窗口以确保你的定制对话框正确显示。

缩进/伸出（Indent/Outdent）

Visual Basic 5.0给你提供了自动缩进或伸出一行或多行程序代码的能力。这将使你的程序代码更便于阅读。这项功能意味着当你要把一大段程序代码放入一个If...Then语句中

时，你不必去改变每行代码的缩进量。要了解自动缩进/伸出特性是如何工作的，请按下列步骤操作：

- 1 启动Visual Basic。当Startup屏幕出现时，创建一个新的标准EXE项目。
- 2 当该项目创建后，通过双击Form1的代码窗口来打开它。
- 3 敲入新的代码行。敲入注释行（以单引号或Rem打头的语句）就可以了，只要其长度为多行即可。
- 4 选定两行或三行代码。
- 5 按一、两次Tab键。这段代码就会缩进。
- 6 按Shift+Tab键。这段代码就会伸出来。

可定制的工具栏

你拥有移动、缩放和定制Visual Basic 5.0工具栏的能力。这个特性与Microsoft Office应用软件中多年来一直拥有的可剪裁和可定制的工具栏不同。

Visual Basic中有四个工具栏：Standard（标准）、Edit（编辑）、Form（底表）和Debug（调试）。Visual Basic通常显示适合于你当前操作的工具栏。但有时你可能想把一个经常要用的命令添加到现有的VB工具栏中。例如，假设你频繁地使用Toggle Bookmark命令，并且想让这个命令出现在工具栏而不是Edit菜单中。那么，你可以按以下步骤在Visual Basic工具栏上增加一个Bookmark按钮：

- 1 使用鼠标，右击一个现有的工具栏（或者使用菜单命令View，Toolbars，Customize）。
- 2 Customize对话框将出现。
- 3 单击Commands标签。
- 4 在Categories下，单击Edit。
- 5 此时，在Commands之下的列表框（list box）中将出现一份编辑命令列表。向下滚动并找到Toggle Bookmark。图1.3显示了这个对话框，其中的Toggle Bookmark命令是高亮度的。
- 6 单击并拖动Commands列表中的Toggle Bookmarks到该工具栏。Toggle Bookmarks命令现在驻留在该工具栏上。

如果你愿意，你现在就可以修改你刚刚创建的Toggle Bookmarks工具按钮。为此：

- 1 在Customize对话框中，单击Modify Selection按钮。
- 2 Modify Selection菜单将出现，如图1.4所示。
- 3 选定Change Button Image，然后单击你选中的图像。

- 4 返回Modify Selection菜单并选中Image and Text。当你不熟悉这条命令或者当图标无法使你想起按钮的作用时，这种选择是最合适的。
- 5 如果你想从定制的工具栏中移走新创建的按钮，你只需简单地单击它并把它从工具栏上拖走。（这仅在Customize对话框打开时才行。）
- 6 当你不需要Customize对话框时，请单击Close。

图1.3 定制工具栏

向导 (Wizards)

Visual Basic 5.0带有两个能使开发控件过程自动进行的新向导。

ActiveX控件接口向导（ActiveX Control Interface Wizard）是一个工具，它自动在你编写的控件中插入属性甚至代码。你可以用它把现有控件的属性添加到你的控件中（这个过程叫做“委派”）或者创建全新的属性和方法。虽然控件有许多属性，但实现它们的代码是完全一样的，所以这个向导能使创建可编程接口的过程不那么单调乏味。

过程视图与全模块视图

许多刚开始使用Visual Basic的程序员会因为不能同时看见他们所编写的代码而烦恼，因为他们害怕代码以后会“消失”在接口中。全模块视图非常方便，因为它允许你比较代码中相邻的过程。当你为控件定义一个新属性时，这种方式特别有用，这是因为实现一个属性至少需要两个过程（通常需要更多的过程）。要了解过程视图与全模块视图之间的差别，请按以下步骤操作：

- 1 双击一个标准的EXE项目表格以打开其代码窗口。
- 2 敲入一些代码。可以敲入几行注释。
- 3 利用代码窗口右上边的Procedure列表框切换到不同的过程。
- 4 你应该能在代码窗口中同时看到两个过程。单击代码窗口左下角的Procedure View按钮。现在，你仅能在代码窗口中看到当前的过程。

当你使用Visual Basic 5.0编程时，根据你正在编写的代码类型，你可能会频繁地在Procedure View和Full Module View之间切换。如果你正在编写一组短小的属性过程，你可能想采用Full Module View方式。但如果你正在废寝忘食地编写一个很长的全局过程，那你恐怕应该切换到Procedure View以减少屏幕上显示的组件数。

如果你总是希望在Procedure View方式下开始编写代码（也就是说，你希望Visual Basic 5.0以Visual Basic旧版本的方式工作），那么你可以按以下步骤进行设置：

- 1 选择菜单命令Tools及Options。
- 2 确认Editor标签已被选中。
- 3 清除标有Default To Full Module View的框。
- 4 单击OK

书签 (Bookmarks)

在Visual Basic 5.0中，你可以在代码中设置书签以便能方便地回到你正在编写的代码处或者快速地在两段代码之间切换。你可以移到上一个或下一个书签处，也可以清除项目中的所有书签。在这方面，书签的作用很像断点，只不过它们不会使你的程序暂停。

要了解书签如何起作用，请按以下步骤操作：

- 1 打开任何一个代码窗口。
- 2 单击该代码窗口左下角的Procedure View按钮以切换到过程视图方式。
- 3 敲入一些代码。如果愿意，你可以敲入注释。
- 4 把插入点移到你所敲入代码的中间位置。
- 5 选中菜单命令Edit、Bookmarks及Toggle Bookmarks。
- 6 在你指定行的左边将出现一个明亮的兰色菱形书签。
- 7 利用右上角的Procedure列表框，切换到该模块中的另一个过程。
- 8 选中菜单命令Edit、Bookmarks和Next Bookmarks。你就能迅速返回到你的原始书签的位置。
- 9 选定菜单命令Edit、Bookmarks和Clear All Bookmarks，清除你建立的书签。

自动补全代码

当你在Visual Basic 5.0中编写代码时，你会注意到当你敲入语句时，屏幕上出现了一些有用的提示。这些提示由Visual Basic 5.0编辑器的几个特性构成，这些特性统称为Automatic Code Completion（自动代码补全）。表1.1列出了Visual Basic 5.0中的代码补全特性。

表1.1 Visual Basic 5.0中的代码补全特性

特性	显示
Auto List Members	对象类的成员
Auto Quick Info	Visual Basic函数句型
Auto Data Type	显示光标所指变量的值

这三种自动代码补全元素都是缺省设置的。如果愿意，你也可以关闭这些特性。为了演示Auto List Members是如何实现的，请打开任何一个代码窗口。敲入Debug.P。在你敲入的地方将弹出一个Auto List Members窗口，显示出这个Debug对象的两个成员，Assert和Print。（Assert是Visual Basic 5.0新增加的，我们将在本章的后面讨论它。）图1.5显示了Auto List Members窗口的外观。敲入Tab键以接受Print作为已完成的语句。

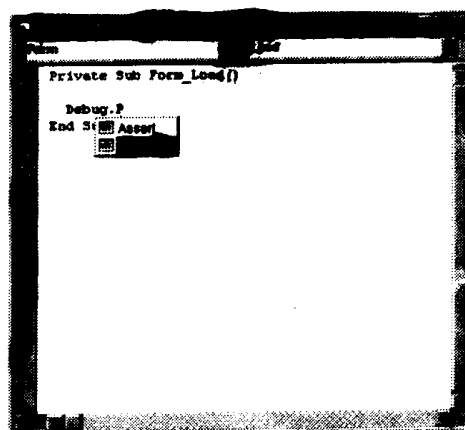


图1.5 Auto List Members窗口

为了演示Auto Quick Info是如何工作的，请按以下步骤操作：

- 1 在任何一个代码窗口中，敲入以下代码：
MsgBox
- 2 敲入一个空格，就好像你打算给MsgBox语句敲入一个参数以完成这个语句一样。
- 3 此时将出现一个Auto Quick Info窗口，窗口中给出了MsgBox语句的句型。图1.6显示了Auto Quick Info窗口的外观。
- 4 结束MsgBox语句的输入，或者按Esc退出Auto Quick Info窗口。

Auto Data Tips很像ToolTips，如果你是任何一种Microsoft Office产品的用户，你可能熟悉ToolTips。例如，在Microsoft Word中，当你把鼠标光标定位在一个工具栏按钮上并且稍等一会儿，就会出现一个ToolTips告诉你这个按钮是干什么的。差别在于Visual Basic的Auto Data Tips仅对正在运行的代码有效。因此，为了演示它们如何工作，你必须编写一些可运行的代码。