

本书叙述了 Visual Basic 5 所支持的插件开发的所有主要能力和特征。全书共分四大部分，即对插件的介绍、如何使用插件对象和成员、插件程序专题研究和插件词典。

本书适合有经验的 VB 和 Windows 开发者使用。

Gene Swartzfager: Visual Developer Creating Visual Basic 5 Add-ins.

Authorized translation from the English language edition published by The Coriolis Group, Inc.

Copyright 1998 by the Coriolis Group, Inc.

All rights reserved.

本书中文简体字版由机械工业出版社出版，未经出版者书面许可，本书的任何部分不得以任何方式复制或抄袭。

版权所有，翻印必究

本书版权登记号：图字：01-98-0741

图书在版编目（CIP）数据

Visual Basic 插件开发人员指南/（美）斯沃德非格（Swartzfager, G.）著；健莲工作室译. - 北京：机械工业出版社，1998

（计算机软件开发与程序设计系列丛书）

书名原文：Visual Developer Creating Visual Basic 5 Add-ins

ISBN 7-111-06687-1

I . V… II ①斯… ②健… III . Basic 语言-程序设计 IV . TP312

中国版本图书馆 CIP 数据核字（98）第 24647 号

出 版 人：马九荣（北京市百万庄大街 22 号 邮政编码 100037）

责任编辑：傅豫波 李云静

北京忠信诚胶印厂印刷·新华书店北京发行所发行

1998 年 10 月第 1 版第 1 次印刷

787mm×1092mm1/16·22.25 印张

定价：58.00 元（含光盘）

凡购本书，如有倒页、脱页、缺页，由本社发行部调换

序

当 Gene Swartzfager 打电话来问我是否愿意为他的新书《Visual Basic 插件开发人员指南》写序时，我对他说：“我当然很乐意，但你知道我快要退休了，而且作为一个工作了 25 年的计算机业者，我已经厌倦了。”他说正是由于这个原因，我才是写这个序的合适人选。我的回答是：“我有些不明白。”

然后，他解释到，因为这些年我一直与 Ford Motor 公司的 MIS 部门一起工作，所以是介绍新的插件软件技术重要性的合适人选。他指出，当 IBM 的“big iron”计算机处于统治地位，而且公司开发者所需知道的语言只有 Cobol 和汇编时，我就已经处于计算机编程的史前纪元。近 20 年以后，在我有生之年幸运地看到了微软校园里 Windows 95 初次公开展出时的盛大场面。他使我想起了在我任 Ford 公司的 Windows 95 β 版程序测试主管的时候，我曾与 Bill 总裁及 Jay Leno 一起照过相。Gene 说，我已经目睹并经历了一切，所以我最有资格将 VB5 的插件置于合适的历史位置。

我决定把 Gene 所说的都作为一种称赞并暂停在网上的冲浪。在 Ford 公司中工作的最近几年里，我一直关注着 VB 的发展，因为在公司中我领导的许多“优秀人才”都使用它来编写 Windows 应用程序。我将他的书的 CD-ROM 文件安装在我的 PC 机上，开始攻读 VB5。我不得不承认这一段时间我并没编写多少 VB 代码，但我知道 Add-Ins 菜单在哪里，因此我花了更多的时间来熟悉他所编写的各种不同的插件。

我被微软 VB 开发组对插件技术所做的改进所吸引。如果在我大汗淋漓地编写 Cobol 代码时有了这种工具，编程将更有趣，每天我也会多编写一些代码，而且我的头发可能不会像现在这么花白。VB5 的插件是这种语言的神奇的补充。Gene 所写的如何使用扩展的对象库定制 VB 的开发环境以及如何自动完成编程中的单调乏味的工作都是一流的。

微软在计算机编程发展中的作用仍然是不甚清晰的。Windows 和 NT 是目前最具历史影响力的产品。但是我认为用长远的眼光来看，ActiveX/ COM/ DCOM 软件结构和它的基于组件开发的承诺将日益占上风。在下一个十年里，这种结构将对除 Windows 以外的操作系统（包括网络操作系统）产生重要的威胁。这种结构的动人之处在于它的灵活性、高雅性以及易再用性。你可以编写各种不同的应用程序，比如插件、服务器、控件和文档，并且用基本相同的方式再利用它们，这是一种非同一般的功能。

我积极地向任何想停留在“优秀程序员行列”的 VB 程序员推荐 Gene 的关于插件的书。在 ActiveX 和可再利用软件的大的影片中，插件只能扮演一个小角色。但是，对于一个想在他的同类人中出类拔萃的 VB 程序员来说，掌握插件开发技术是成功的必由之路。

—— Jim Schott

负责微软 Windows 95 β 程序测试的经理

前言

所有的演讲，写的或说的，在发现一个自愿的且已做好准备的听众之前，都是一种死的语言。

——Robert Louis Stevenson

这本书是写给那些想充分利用 VB5 专业版和企业版中所包含的新的插件扩展对象模型的有经验的 VB 和 Windows 开发者的。这种插件对象模型包含在文件 VB5EXT.OLB 中；VB5EXT.OLB 的参考名是微软 VB5 扩展，而在 VB 对象浏览器中，它的类库名为 VBIDE。VB5EXT.OLB 使你可以开发微软所指的 VB5 的插件。

一个 VB5 插件是一个你使用 VB5EXE.OLB 中的集合、对象和成员通过编程创建的 ActiveX 组件。与 Microsoft Excel, Access, 或 FoxPro 插件不同，一个 VB5 插件，是一种为在 VB5 集成开发环境中使用而设计的特殊的 ActiveX 组件。开发插件的最初的原因（从这以后，插件专指 VB5 插件）使你可以自动完成的 VB 的 IDE（集成开发环境）中一些比较困难、乏味或耗时的功能。

我不太清楚微软为何使用插件来描述你用扩展对象模型创建的 ActiveX 组件。然而，即使是巧合，注意到缩写 AI 既可用于代表插件，也可用来代表人工智能也是很有趣的。确切地说，插件是微软使 VB 开发者向 VB 的 IDE 中加入人工智能或智能代理的一种途径。

我这样说冒着让人感到自命不凡的危险，谁知道将来如果说 VB 编程语言开发环境具有人工智能将意味着什么呢？

通常，当介绍一种复杂的新的软件技术时，它可能的应用及细节都只能被模糊地理解。对一项新技术的普通反应是抓抓脑袋说：“噢，那很好！但它的优点在哪儿呢？”这也是本书将要回答的问题。我写这本书有以下几个目的：

- 告诉你如何编写你自己的常用插件，以及如何成为一个更高效的 VB 程序员。
- 作为关于插件和 VB5 EXT.OLB 中包含的对象模型的参考书。
- 在你学习如何使 VB 的 IDE 越过圈套，拖出野兔时，你会有神奇的感受并从中得到很多乐趣。

无论你是具有 500 或 1000 水平的 VB 开发者，Microsoft Solutions Provider 公司的 VB 开发队伍中的一员，或只是一个梦想编写一个新的 ActiveX 组件的 VB 发烧友，学习如何创建插件对你都是有益的。这篇简介说明了在写这本书时使用的假设和约定，相应的插件代码，并介绍了 VB5EXT.OLB 的 Windows 帮助文件。它还将告诉你如何安装随书的 CD-ROM 中的文件，并且提供了一些对你深入了解书中其他内容有帮助的信息。

使用这本书所需的预备知识

要成功地使用这本书，你必须是一个有经验的 VB 程序员。一个插件可能是你能用 VB5 创建的 ActiveX 组件中最复杂也最难调试的一种了。初学者无需使用插件开发工具；而且，虽然这本书有几百页之多，却没有浪费时间来解释通常的 VB 编程。本书认为在使用了 VB

的 4.0 版以后，你理所当然地具有了关于 VB 语法及语言元素的知识。

你需要一台安装了 VB5 专业版或企业版的计算机，因为在 VB5 的学习版或创建控件版中不包含 VB5EXT.OLB。你还需配置 Windows 95 或 Windows NT (3.51 或更高版本) 操作系统以运行随书 CD-ROM 上的安装程序 SETUP.EXE。这是因为与 VB4 不同，VB5 只有 32-bit 版本。

本书中关于插件的语言元素及语法的引用都是针对 VB5 版本的。虽然在 VB4 中包含插件能力和一个对象库 (VBEXT32.OLB)，但其对插件的利用是原始的，而且其文件是不充分的。

如果你想开发 VB5 插件，了解一些关于面向对象程序设计（简称 OOP）开发示例的术语及微软关于 VB5 的特殊 ActiveX 实现的知识是有益的。最理想的是，本书的读者已经阅读了我先前出版的名为《VB5 面向对象程序设计》的书，该书由 Coriolis Group 出版社出版。然而，这并不是真正必须的，在简介中我将介绍在插件开发中用到的基本的 OOP 术语及概念。

本书包含什么内容

本书解释、归档并演示了 VB5 所支持的插件开发的所有主要能力及特征。它包括一个前言，9 章内容，一个插件语素词典和一个包含插件例子及其源代码的 CD-ROM。该 CD-ROM 中还包括一个 Windows 帮助文件，以清晰易懂的形式阐述了关于目前已有的插件的参考信息。本书共分四部分。

第一部分

第一部分为关于插件的介绍，包括第 1 章和第 2 章。第一部分包括了在写本书时的一些假设和约定，相应的插件代码以及关于插件语素及语法的最好参考——Windows 帮助文件 (ADDINEFS.HLP)。它阐述了插件开发周期中涉及到的步骤并引导你创建一个简单的插件。它还提供了一个可以演示插件对象模型的 243 个独立语素插件。最后，它将指出你如何定制并增强一个插件的内部和外部接口 (Add-In Manager 对话框，VB 插件工具条，IDTExtensibility 对象等) 的不同部分的行为。

第二部分

介绍如何使用插件对象和成员，包括第 3 章到第 6 章。第二部分将 VB 的扩展对象模型分为四个范畴。

- 操纵 VB 的菜单、工具条和其他 IDE 级的插件对象。
- 处理高水平插件对象 (工程、参照、代码块、设计器和控件)。
- 操纵低级对象 (属性、成员和代码)。
- 处理并对与插件相关的事件做出反应。

在第二部分中，你将学习如何在句法上将插件对象模型的不同对象集合和成员组织起来，并创建真正有用的插件。

第三部分

学习插件的实例，包括第 7 章到第 9 章。第三部分学习集合了插件对象模型各部分而

成的复杂的、成熟的插件，包括下面这些有用的应用程序。

- Dialog Box Builder 插件。
- VB Programmer's Log 插件。
- ActiveX Control Creation Tutor 插件。
- Code Librarian 插件。

第三部分阐述了这些插件的设计开发过程，演示了它们是如何工作的，并建议了增强它们的方式。

第四部分

插件词典，按字母顺序列出了插件对象模型中所有的集合、对象和成员。你将发现这个词典将因三个原因而成为特别有用的参考工具。第一，VB5 上市时没有任何印刷文档。第二，关于 VB 中的插件和扩展对象模型的在线信息不充分、易混淆，甚至有时是错误的。第三，这个词典参照并由大量的示例代码（在 ADDINEFS.HIP 文件中）补充，这些代码是确保运行通过的，因为它们都在随书的 CD-ROM 的 DEMOSYNT.VBP 工程中。

这个词典有 VBAI 类库中所包含的对象和成员的条目，而 VBAI 类库是我为插件开发所写的方法的集合。VBAI 类库中的方法可以：

- 增强插件对象模型的能力。
- 简化插件开发中经常遇到的任务。
- 定制 VB 插件工具条和你自己的插件的行为。

从 CD-ROM 中安装文件

为了安装随书的 CD-ROM 上的文件，将 CD-ROM 插入到你的 PC 的光驱中并关上它。CD-ROM 上的 AUTORUN.INF 文件将自动显示图 1 所示的 Creating VB5 Add-Ins Setup 对话框。如果没有显示，从 Windows Explorer 中双击 CD-ROM 根目录下的 SETUP.EXE 文件也可以显示这个对话框。

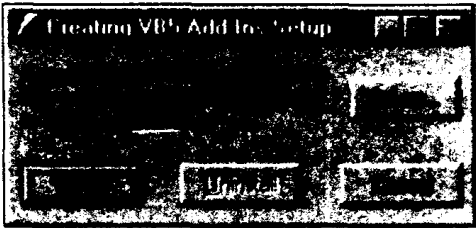


图 1 VBADDIN.INI 文件中的 Add-in entries

选择你想安装该文件的逻辑驱动器（缺省值为 C:），并单击 Install。然后安装程序将创建表 1 所列的子目录结构并将文件拷至合适的子目录下。安装程序还将自动为所有的插件登记注册，并将 ADDINEFS.HLP 文件拷至 Windows\HELP 子目录下，以便它可被书中所有的插件共享。安装需要约 6MB 的硬盘空间。在你将它们从 CD-ROM 上安装至你的硬盘之前，你无法运行这些插件。

表 1 随书附带的 CD-ROM 的子目录结构

子 目 录 名	文 件 说 明
\ ADDINEFS \ BRWSAPIS	16 位 Windows API 浏览器演示程序的插位版本
\ ADDINEFS \ CHAPTER1	第 1 章讨论的简单插件
\ ADDINEFS \ CHAPTER2	第 2 章讨论的，分别演示所有的 243 个插件语素与所有界面定制特征的插件
\ ADDINEFS \ CHAPTER3	演示如何控制其他插件，如何向 Windows 注册器写入注册项以及如何操作 VB 菜单与工作条的插件
\ ADDINEFS \ CHAPTER4	演示如何处理 VB 高层组件（工程、参考、代码模块、设计器和控制）的插件
\ ADDINEFS \ CHAPTER5	包括两个插件。一个用于演示如何操作低层对象（属性、成员和代码），另一个是 Add-In Interface Builder 插件
\ ADDINEFS \ CHAPTER6	包括两个插件。一个是 Events Log 插件，用于演示如何处理由 VB 激发的与插件相关的事件及如何对其进行反应。另一个是 Controls Monitor 插件
\ ADDINEFS \ CHAPTER7	包括两个插件：Dialog Box Builder 插件与 VB Programmer's Log 插件
\ ADDINEFS \ CHAPTER8	用于 ActiveX 控件创建的 Tutor 插件
\ ADDINEFS \ CHAPTER9	Code Librarian 插件
\ ADDINEFS \ KNOWBASE	包含来自微软知识库的关于插件文章的文件
\ ADDINEFS \ VBAI	关于 ActiveX 服务器组件的源程序，属于 VBAI 类库
\ ADDINEFS \ WINDINFO	Window Information Utility 插件

在你运行了书中的安装程序后，你将会看到 Setup 对话框上的 Uninstall 按钮成为有效的。如果你从书的 CD-ROM 上安装的文件出错了，而你需要重装它们，你需要首先单击 Uninstall。这样，程序将 Remove 损坏的文件和子目录结构，再单击 Install 重新运行安装程序。由于随书的 CD-ROM 的 Uninstall 的这种工作方式，你不能将你自己的任何文件存于 \ ADDINEFS 子目录下，以防你单击 Uninstall 键时将它们 Remove。

提示 如果你想 Windows 95 和 NT（4.0 版或更高）操作系统下安装并注册 CD-ROM 的插件，运行 SETUP.EXE 并单击 Install。对这两个操作系统你可以将文件安装在同一驱动器或同一部分上，以节省硬盘空间

ReadMe 变动及更新

有几点改变及更新没有包含在书的词典部分。三个插件语素（GetGuid 方法，InsertFile 方法和 VBNewProjects 集合）的内容在书中未提及，但你可以在 ADDINEFS.HLP 中找到它们。

通常，你可以认为 ADDINEFS.HLP 包括了 VB5 的插件对象模型的最新版的参考信息。如果你发现了书中词典部分所提供的信息与 ADDLNEFS.HLP 中关于特定语素的信息间的不同，你应该依赖于帮助文件的信息。这种差异一般是很小的。

印刷约定

本书中使用的印刷常规如表 2 所示。

表 2 印刷约定

例 子	说 明
Err.Raise 3	插件代码举例使用这种字体
....	这种字体及用空格分开的四个点表示代码存在于插件过程中但未列出
[HelpFile]	在语法上，方括号内的声明表示它是任选的
Public Private	在语法上，竖线（ ）表示其两边的项是互斥的
ADDINEFS.HLP	大写字母组成的单词表示它是目录或文件的名字

程序约定

本书中所列的示例代码中使用的程序约定如下。

- 撇号（'）后跟一个空格代表代码注释：

```
'Constants and variables:
```

- 所有的注释都以空行开始，并位于被注释代码的上面。不以注释行结束。
- 控制流代码块缩进几个空格。

```
Function GetDesc (VBE As VBE, ProgID As String) AS String
```

```
'Check for spectified failure case:
If Not IsProgID (VBE, ProgID) Then
    GetDesc = ""
    Exit Function
End If
```

```
....
```

```
End Function
```

- VB5 支持行连接符的使用，行连接符为一个空格加一个下划线（_）。当代码语句太长，在一行中不能显示完时可使用行连接符在下一行显示。每个待续行比它上面的行缩进一些。例子如下。

```
'Center form:
Move (Screen. Width-Width) * 0. 5, _
    (Screen. Height-Height) * 0. 5
```

- 每个变量都事先定义。假定在 VB 的 Options 对话框（Tools|Options）中关于 Editor 标签的 Require Variable Declaration 被设置。
- 所有的声明遵循以下格式：

```
Dim Item As String
```

其中，声明的数据类型通过使用 As 关键字明确指定。这条规则的唯一例外是 Windows API 函数的声明。在该种情况下，尽可能多的使用后缀以节省空间，从而减少行连接符的使用。例如：

Declare Function LockWindowUpdate & Lib"USER32" (ByVal hWndLock &)

- 如果存在一个 VB 固有的或枚举常量，可以使用它。例如：

```
'Warn user there will be a delay.
Screen.MousePointer = vbHourglass

'Reset default cursor.
Screen.MousePointer = vbDefault
```

- 如果对于 Windows API 函数的自变量或返回变量存在常量，本书也使用它们。例如：

```
' Constants for Windows API functions:
Const SWP_NOSIZE = &H1
Const SWP_NOMOVE = &H2
```

• 如果没有 VB 枚举常量或 Windows API 函数常量存在，本书的代码声明自己的常量，并为这些常量分配数字的或字符串的值以及控件数组索引值。书中代码声明的所有枚举常量都以小写字母 ai 为前缀。

代码的命名约定

命名约定与宗教信仰类似。每个 VB 程序员都能给出几种原因来解释为什么他或她的命名规则比其他人的更好。多数程序员都不会浪费时间来试图说服别人。然而，如果你不遵守所描述的命名约定，在阅读关于 VB 或 Microsoft Access 编程的书时，你会感觉到难以理解 VB 代码。微软为 VB 和 VB For Application (VBA) 建议的一些规定在 Visual Basic Books Online Help|Books Online 中做了大致阐述，其专题名字为 Constant And Variable Naming Conventions。

本书中关于 VB 内建对象的命名约定基于微软对 VB 和 VBA 所建议的规定。然而，书中其他的命名规则是假定你将使用 VB5 编写封装的、面向对象的代码，并编译为插件 ActiveX 组件。当你这么做时，你是想编写与 VB 的和 VBA 的成员一样的过程。为此，本书这样命名。

- 用动词为方法过程命名。如果需要，动词后跟一个指明相关对象的名词。
- 用名词为属性过程命名。
- 用名词，偶尔也使用动词为变量命名。变量的名字并不以代表其数据类型的三个前缀字母打头，因为 VB 的和 VBA 的成员变量不使用三个字母前缀。

为变量命名

当变量被明确声明时，使用描述性名字，而且不带任何前缀来指明它们的数据类型。听起来，这与经常使用的带一定前缀的 VB 命名方法相比是最坏的一种异端。不使用前缀的理由是。1) 所有的过程都是封装和面向对象设计的一部分；2) 99% 的插件代码变量都声明为过程级的；3) 代码要尽可能地读起来更像自然语言。

微软提倡的 Windows 编程语言和 ActiveX 组件软件协议使得编写高聚合、封装的过程/模块/类变得简单。如果你遵守面向对象的编程范例，包含类库或组件的成员的过程应该相对短些，而且不能与其他过程的代码结合，除非使用消息传送。一般情况下，不能使用

Public/全局变量，而且除了那些与 VB Get, Let 和 Set 属性的过程联系所必须的，模块级变量也应该尽可能少的声明（然后详细注释它们的用途）。

对于那些遵循 OOP 范例的 VB 开发者来说，继续用详细阐述和秘密方式为变量命名并采用大作用域变量使过程变得乱七八糟的程序员，与那些继续遵守已废弃的宗教道义的人没有两样。

变量的作用范围用下面的方式指定。

- 公用的和全局的变量只用于特定情况，而且不使用前缀来标明它的作用域。第一，使用 WithEvents 关键字来声明的对象变量需要是 Public 型的。第二，为了提高可阅读性，VBE 对象变量通常与 VB5EXT.OLB 对象库中的 VBE Object 具有相同的名字。第三，参照书中 VBAI 类库（AITBar, Gmd, Reg 和 Util）中对象之一的对象变量与它实例化的类取相同的名字。

- 模块级的变量声明以 m 为前缀。对这个规则有两个例外情况。第一，VBE 对象变量的命名经常与它所参照的 VBE 对象相同。第二，一个参照 VBAI 类库的对象之一的对象变量与它所实例化的类具有相同的名字。

- 除了前面所提到的使用 Public 变量的情况以外，所有模块级变量使用 Private 语句声明。使用 Dim 或 Static 语句声明的过程级变量当然是私有变量。

窗体和控件对象的名字

VB 固有的所有的窗体和控件对象都用小写字母定义，用三个字母的前缀指明这个对象的类。这些前缀列于 Visual Basic Books Online 中 Object Naming Conventions 主题下。

提示 对于控件数组中的控件我的确违背了上述约定。在这种情况下，我只使用控件的三个字母前缀作为它的 Name 属性。控件的 Index 属性值由枚举常量确定。例如，作为控件数组一部分的两个 CommandButton 对象，一个表示确定（OK），一个表示取消（Cancel），在代码中分别用 cmd(aiCBOK) 和 cmd(aiCBCancel) 来标识，其中 aiCBOK 和 aiCBCancel 为枚举型常量。

其他约定

本书中使用的其他规定有：

- 除非特别说明，字母 VB 代表专业版的 Visual Basic 5.0。
- 代词你指 VB 开发者或阅读此书的人。
- 连续的菜单选择操作作用字符（|）分隔。
- 贯穿整书，参考都指向存有不同文件的路径（例如，C: \ ADDINEFS \ CHAPTER2）。根目录假定为 C:，但你必须记住实际安装时你所选的驱动器。
- 开头字母 EFS 用于书中 Help 文件的名称及安装子路径中。它们是我的名字的前三个字母（Eugene Francis Swartzfager），使用它们没有什么特殊意义（除了让我母亲特别高兴以外）。

面向对象的术语及概念

在这篇前言的前面我已提及，为了更好地理解本书中的材料及 ADDINEFS.HLP，了解

一些关于面向对象程序开发范例的术语及微软的特殊 ActiveX 实现的知识是有益处的。在本书中我无法用一个完整的部分来写关于 OOP 的概念，但有近半打的术语是经常用到的类，对象实例化，成员方法公用接口消息发布。这些术语在下面列出并进行了简洁的解释。

- 类：类是定义一个对象属性可能设置的模板或正式类型，方法则规定对象的行为，事件是对象的反应。对象的属性、方法和事件（通常指成员）的正式定义只在类的创建之初进行一次。对类经常使用的一个比喻是它像一个甜点切割机。类的例子是人类，男性和女性。类可被分层组织，男性和女性可看作人类的子类，而女孩和妇女又可看作是女性的子类。

- 对象：一个对象是一个特定类的例子或实例。当一个类被实例化时，就创建了一个对象或实例，而它的属性被初始化为一些特定的缺省值。这些属性的设置可以在对象的生存期内改变，但只能在对象所属的类所允许的范围内。一个对象就好比甜点切割机生产的一块甜点。对象的例子是 Tom 和 Dick（属男性类）；或 Susie 和 Mary（属女性类）。Tom 的属性有 HairColor 和 Age 等，它们的值随时间的推移而变化。

- 实例化：实例化是在类中创建一个对象的过程。在 VB 中，你可以使用 Create Object 函数或 Dim As New 的声明方式来实例化一个对象。

- 成员：一个成员是一个对象的元素。在 VB 的 OOP 实现中，它是一个与属性、方法或事件等同的一般术语。在抽象的代码等级上，一个 VB 成员由一个过程组成。在作用域上，成员可以是 Public 型或 Private 型的，而只有 Public 型的成员才显示在对象浏览器中。

- 方法：方法是作用于一个对象上的 Sub 或 Function 过程，以使其按照一定的方式动作。

- 公用接口：一个类的公用接口是可被公用的信息。类或类的对象实例有时称为“黑盒子”，将它的实现中的多数细节隐藏起来。在 VB 的 OOP 实现中，你能了解的关于类的信息（即类的公用接口）通过对象浏览器而公布。VB 类库或 ActiveX 组件的公用接口中的主要元素是它的 Public 型类和这些类的 Public 型过程/变量/返回值。如果一个类库或 ActiveX 组件有相应的 Windows 帮助文件，帮助文件中的信息可以认为是公用接口的一个扩展。程序员通过向对象的公用接口的元素发布消息而重用它。

- 消息发布：消息发布是对象间相互联系的过程。发送给对象的消息包括方法执行、设置/返回属性或激发事件。为了发送信息，你要指定对象的名字，成员的名字，和成员的变量（必选或可选）所指定的参数。一个对象的许多成员也向对象的使用者返回消息，其中包括返回值。

提示 要了解更多的关于 VB 面向对象程序设计的信息，可参阅我写的名为《VB5 面向对象程序设计》一书，由 Coriolis Group 出版社出版。用 VB 编程关于 ActiveX 组件的最好的书可能是 Don Appleman 的《用 VB5.0 开发 ActiveX 组件》，由 Ziff-Davis 出版社出版。

第 2 章的插件

这本书的词典部分将是它的最常使用的部分。我是先写的词典，在我写书的其他部分及书中的插件代码时，我时常参阅它。然而，我必须声明这本词典是不完整的。出版商和我都认为，如果我们在词典中将所有 243 个独立语素的示例代码都印刷下来，是不明智的。打印

示例代码需要再加 300 页，也就意味着读者要多花 10 美元到 15 美元来买这本书。因此，我决定创建一个插件（在第 2 章中讨论），其唯一的目的是演示 243 个插件语素并运行示例代码。

提示 此外，你还可以在 ADDINEFS.HLP 的帮助主题下找到每一语素的示例代码。

运行 Demo Syntax 插件

在你运行书上的 CD-ROM 中的 SETUP.EXE 时，这个 Demo Syntax 插件也被安装并注册。要运行并使用它，采用下面的步骤。

- 假定你已从书上的 CD-ROM 中安装了文件，启动 VB5。
- 选择 Add-Ins|Add-In Manager，显示 Add-In Manager 对话框。
- 选定 Creating VB5 Add-Ins: Chapter2 一项，并选择 OK，关闭对话框。
- 通过单击 VB 插件工具条上的按钮（#2 标志）来运行 Demo Syntax 插件，或通过选择 Add-Ins 并单击 Creating VB5 Add-Ins: Chapter2 菜单项。Demo VB5 Add-In Syntax 对话框如图 2 所示。

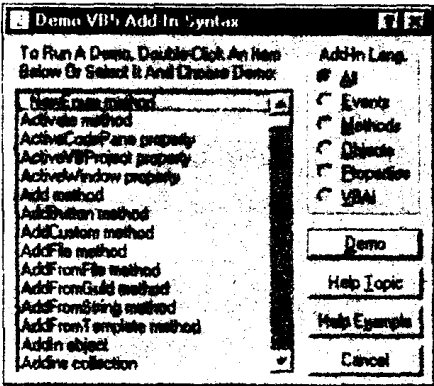


图 2 Demo VB5 Add-In Syntax 对话框

从现在开始，运行一个特定语素的示例代码是不明智的。双击选项（例如，Activate 方法）或选择后单击 Demo，VB 将运行示例代码所指示的动作，并且会显示一系列的消息框，用以解释程序正在干什么。例如，Activate 方法 demo 中显示的第一个消息框如图 3 所示。

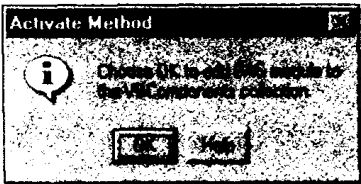


图 3 Demo Syntax Add-In 显示的消息框

如果你单击消息框上的 Help 按钮，ADDINEFS.HLP 中包含示例代码的帮助主题如图 4 所示。

如果你所演示的插件语素不止有一种用途，单击消息框的帮助按钮以显示示例代码，并置于目前正执行的语句行。

提示 要得到关于 Demo VB5 Add-In Syntax 对话框自身如何工作的帮助，单击标

题条右上角的? 按钮，使其成为上下文敏感的 What's This Help 弹出式方式。然后单击你想获得信息的对话框上的控件（例如，ListBox）；则会显示如图 5 所示的弹出式帮助菜单。

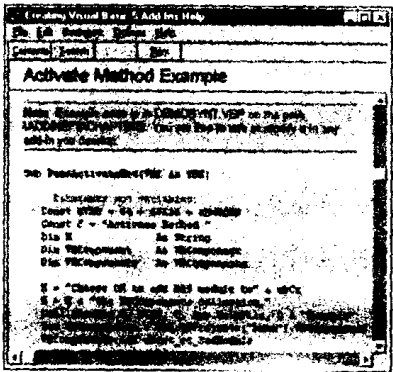


图 4 插件语素的 Example 代码



图 5 Demo Syntax Add-In 的弹出式帮助示例

关于 Demo Syntax Add-In 的其他信息

你还应该了解关于 Demo VB5 Add-In Syntax 插件的一些其他知识点。

- 当你运行它和为插件语素编写的示例代码时，每个示例都从为 VB 的 IDE 增加一个名为“Demo”的标准 EXE 工程开始。这个“Demo”工程的增加是在演示语素的过程执行之前进行的（你将可以看到这个工程被加到 Project Explorer 窗口中）。
- 如果在 VB 的 IDE 中有另外的工程被打开时你运行它，它绝不会干扰或损坏其他的工程。
- 尽管它的主要功能是示范每个插件语素的作用，Demo Syntax 插件也可以用来演示一个插件可以支持的所有可能的界面定制特征。这项功能是通过在执行各种不同的 IDTExtensibility 对象接口方法时显示消息框来完成的。如果你不想受这些消息框的干扰，你可以从插件工具条中运行插件。要了解该怎么做，参见第 2 章中“VB Add-In Toolbar 插件”一节。
- 你可以以 debug 方式运行 Demo Syntax 插件工程 (DEMOSYNT.VBP)，并观察它的源代码的单步执行过程。要学会该怎么做，参阅第 1 章中“测试并调试一个插件”一节。

插件的帮助文件 ADDINEFS.HLP

这本书所带的 CD-ROM 的安装程序将文件 ADDINEFS.HLP 放在你的 PC 上 Windows 或 Windows NT 路径下的 \HELP 子目录下，因此它可供书上所有的插件共享。这个 Windows 帮助文件，是我写本书的词典部分时写的，可以作为 VB5 插件的准确参考。要进入 ADDINEFS.HLP，可以运行第 2 章的 Demo VB5 Add-In Syntax 插件，或在 Windows 管理器中双击该文件。它的内容主题如图 6 所示。

243 个插件语素中的每一个都在此帮助文件中存有文档。例如，Activate 方法的主题如

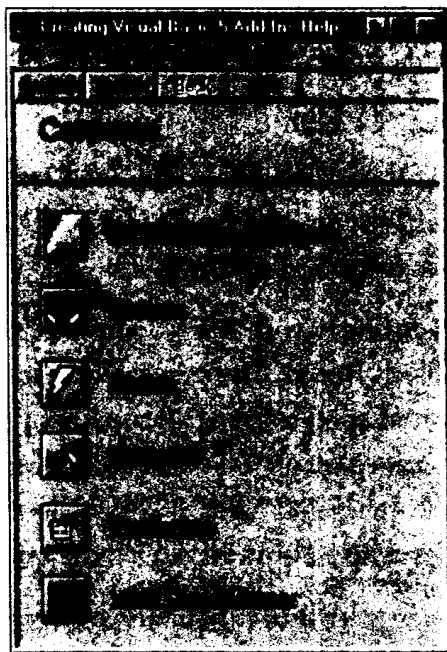


图 6 ADDINEFS.HLP 的 Contents 主题

图 7 所示。ADDINEFS.HLP 使用标准的微软方式为一个 ActiveX 组件提供文档，带有 See Also, Example, 和 Applies To 跳转功能。此外，在 ADDINEFS.HLP 中还提供了关于我的 VBAI 类库的 32 种方法和扩展对象模型的隐藏型成员的文档。

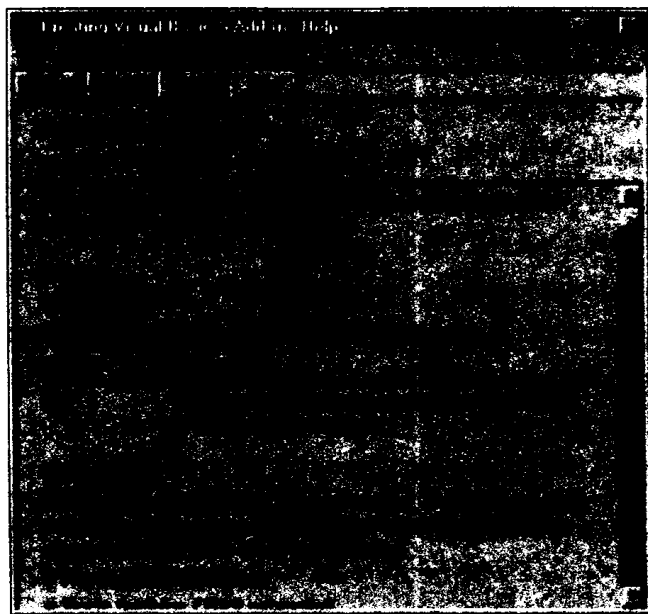


图 7 ADDINEFS.HLP 中 Activate 方法的帮助主题

关于插件的其他信息源

除了本书，它的 ADDINEFS.HLP 文件，它的 Demo VB5 Add-In Syntax 插件以及书上 CD-ROM 中的其他插件以外，我所能想到的关于插件的其他信息源如下：

- Visual Basic Books Online。从 VB 的菜单中，选择 Help|Books Online，然后，从 Visual Basic Books Online 中选择 View|Index，并在 Index 框中敲入“add-ins”以显示与插件有关的关键字和主题的列表。在 Visual Basic Books Online 中能得到的关于插件的信息比 VB4 中所包含的要好，然而，它主要是针对初学者的，不包括更深的例程代码。

- VB 的帮助文件。在这里可以找到关于插件语素的主题，但它所提供的信息经常是不充分，易混淆的，有时甚至是错误的。例如，在 VBIDE 插件类库中有一个 Description 属性。如果你将 VB5.HLP 中不完整的帮助主题与 ADDINEFS.HLP 中全面的帮助主题进行对比，你会发现我为什么说 ADDINEFS.HLP 是关于插件的最清楚详细的参考。

- VB 的插件例子。与 VB 一块发行的有一个插件例子，是用 VB 编写的（VB TabOrder Window 插件），其代码在 VB 路径 \ SAMPLES \ COMPTOOL \ ADDINS \ TABORDER 子目录下。在安装 VB 时并不对其进行注册，因此它并不出现在 Add-In Manager 对话框中。要编译并注册它，按照书中第 1 章名为“编译、注册并运行一个插件”（插件例子的工程文件为 TABORDER.VBP）的部分中所提示的去做。这个插件的源代码值得你学习研究，虽然它的可读性比较差。

警告 不幸地是，尽管与 VB 一起发行了许多 Wizard 型的插件，你却找不到它们的源代码。微软只选择性地发行了这些插件的已编译的 DLL ActiveX 组件，可以在 VB 路径 \ WINZARDS 下找到。

- 其他来源：本书的 CD-ROM 的安装程序将微软知识库或 www.microsoft.com 中关于 VB5 插件的文章放在子路径 \ ADDINEFS \ KNOWBASE 下。实际上并没有太多的信息。

如果有的话，我本打算包含一个关于 VB5 插件的文章和书籍的目录，然而，我查阅了 VB5 发行以来出版的所有关于 VB 的书籍，没有一本包括关于 VB5 插件的详细信息。

警告 在 1997 年 7 月，Wiley Computer 出版社出版了 Rod Stephens 所著的名为《Visual Basic 高级技术》一书。尽管此书包含了许多有用信息，不要被它所列出的目录表中有 4 章内容关于插件的现象所愚弄，那些插件是用原来的 VB4 扩展对象模型编写的。

使用 VB5 的 IDE 的注意事项

VB5 的新 IDE（Integrated Development Environment）的出现是一个好消息，也是一个坏消息。好的一面是新的菜单结构和工具条（现在有多于一个的工具条）比过去 VBA 在 Microsoft Office 97 中使用的要简单。坏的一面是这种 IDE 与 VB3 或 VB4 的有很大不同。因此，如果你刚开始使用 VB5，你要花些精力去熟悉它；如果你已经使用了 VB5，你可以跳过这一部分中剩下的内容。

我要强调的 VB5 的新 IDE 的主要特征是：

- 你第一次启动 VB5 时，它将显示自身缺省的 MDI（多文档界面）方式。当你需要一次打开多于一个工程时，这种新特征迟早有用，现在的 VB5 支持 ActiveX 组件开发。然而，如果你每次只调用一个工程，你可能更喜欢原来的 SDI（单文档界面）方式所提供的更大的自由度。要由 MDI 方式转至 SDI 方式，选择 Tools|Options 选项，并单击选项对话框上的 Advanced 标签，然后你可以选择 SDI Development Environment 选项。

- 为了支持它的新 MDI 方式，VB5 还使其 IDE 的不同窗口（Project Explorer, Proper-

ties, ToolBar 等) 可以移动和合并。要使它们自由移动, 你可以自由地拖动它们, 或者选择 Tools|Options, 单击 Docking 项, 并把你想要自由拖动的窗口的名字除去。

- 第一次启动 VB5 时, 它将显示自身的缺省的工具条配置, 即只显示标准工具条。你可能会疑惑灵活方便的调试工具条哪儿去了。现在的 VB5 可以显示 5 个不同的工具条 (Standard, Edit, Debug, Form Editor 和 Add-In), 而且你可以创建自己的标准工具条。你还可以连接它们或自由地移动它们。为了改变工具条的配置, 选择 View|Toolbars 选项, 并在那里进行。

- VB5 不再有 VB4 的 Insert 菜单。为了给 VB5 工程增加新的窗口, 标准模块、类模块 (或者只是任何类型的文件), 需要使用新的 Project 菜单。在那里你会发现所有的增加选项。

- 与过去的 VB3 相比, 现在的 VB5 让你配置的工程级或 IDE 级的选项特别多。VB5 将这些选项分成两个菜单。要设置工程级选项 (现在指工程的属性), 选择 Project|Properties 菜单, 并在 Project Properties 对话框上的四个标签之一上找到合适的设置。要设置 IDE 级的选项, 选择 Tools|Options, 并在 Option 对话框的六个标签之一上找到设置。

- 与以前的版本一样, VB5 也带有帮助文件, 名字为 VB5.HLP (而不是以往的 VB.HLP)。此外, VB5 有两个与帮助有关的新特征, 它们是 IDE 的补充。一个是 Visual Basic Books Online, 要启动它, 选择 Help|Books Online。另一个与帮助有关的新特征是 VB5 可以显示与你正编写代码的语法元素、对象或成员有关的信息。在你安装 VB5 时, 这些自动的下拉式菜单被缺省地打开。要关闭三个自动特征 (List Members, Quick Info, 或 Data Tip) 中的任一个, 选择 Tools|Options 并在 Option 对话框的 Editor 标签上改变它的设置。

以上是关于本书及 VB5 插件的简介, 下面是该浸湿你的双脚, 在插件开发的游泳池中徜徉的时候了。你不必担心, 因为第 1 章的水是温暖且诱人的。我将在游泳池的浅滩上指引你, 给你显示如何创建、调试、测试、编译并注册一个简单的插件。然而, 在你创建简单的插件之前, 你应该了解插件开发者所能使用的设计选项, 而这正是我们将要开始的第 1 章的主题。

目 录

序
前言

第一部分 对插件的介绍

第 1 章 创建一个简单的 VB 插件	1
1.1 设计插件的公用接口	2
1.1.1 ActiveX 组件的种类	2
1.1.2 为插件启动选项	3
1.1.3 与其他 ActiveX 组件的接口	4
1.2 集成一个插件的模块和成员	5
1.2.1 连接类模块	6
1.2.2 VBEEvents 类模块	9
1.2.3 GenlPrcls 标准模块	9
1.2.4 Form 模块	11
1.3 为插件的类模块过程编写代码	12
1.3.1 Connect 类过程	13
1.3.2 VBEEvents 类过程	15
1.4 Gene 的通用插件代码编写的十条重要提示	16
1.4.1 提示 1— 在 GenlPrcls 中将 VBE 声明为 Public 型的	16
1.4.2 提示 2— 在可能情况下使用 Accessor 属性	17
1.4.3 提示 3— 在可能情况下将变量设为局部的	17
1.4.4 提示 4— 将局部对象变量命名为与它们的类型相同的名字	17
1.4.5 提示 5— 使用 For Each... Next 而不是 For...Next	18
1.4.6 提示 6— 缩短长的对象表达式	18
1.4.7 提示 7— 尽可能使用 With 语句	18
1.4.8 提示 8— 尽可能地使用字符索引访问集合中的工程	19
1.4.9 提示 9— 使用固有的枚举常量	19
1.4.10 提示 10— 使用 Collection 和 Parent 属性	19

1.4.11 测试并调试一个插件	20
1.5 编译、注册并运行一个插件	23
1.5.1 General 标签项	23
1.5.2 Make 标签项	24
1.5.3 Compile 标签项	25
1.5.4 Component 标签项	26
1.5.5 Windows 注册项	27
1.5.6 运行一个插件	28
1.6 解决插件中的问题	29
第 2 章 定制一个插件的界面	30
2.1 Add-In Manager 对话框	30
2.2 VBAI 类库	34
2.3 VB Add-In Toolbar 插件	36
2.3.1 插件工具条的注册项	38
2.3.2 通过编程来显示或隐藏插件工具条	41
2.3.3 控制插件工具条按钮的图标	43
2.3.4 编写插件工具条的行为代码	46
2.4 定制 IDT 扩展对象接口	48
2.4.1 On Connection 方法的三个方面	48
2.4.2 定制 OnStartupComplete 方法	53
2.4.3 Tweaking OnDisconnection 方法	55
2.5 插件的窗体只显示一次	58
2.6 从插件中显示常用消息框	60

第二部分 如何使用插件对象和成员

第 3 章 VBIDE 级插件对象	65
3.1 VBE 对象	65
3.1.1 活动对象的访问属性	67
3.1.2 集合的访问属性	68
3.1.3 其他 IDE 级成员	68
3.2 Addins 集合对象	69
3.2.1 AddIn 对象的成员	69
3.2.2 Addins 集合的 Update 方法	70
3.3 CodePanels 集合对象	71
3.4 CommandBars 集合对象	72
3.4.1 使用 Office 成员的缺点	72

3.4.2 使用 Cmd 对象的方法	73	5.4 Add-In Interface Builder 插件	110
3.5 VBProject 集合对象	76	5.4.1 Add-In Interface Builder 插件的窗体	110
3.5.1 向 VBIDE 中增加一个工程	76	5.4.2 Add-In Interface Builder 的代码	111
3.5.2 保存一个工程组及 Remove 一个工程	78	第 6 章 与事件相关的插件对象	116
3.5.3 其他的 VBProjects 集合成员	79	6.1 Add-In Events Log 插件	117
3.6 Windows 集合对象	80	6.1.1 声明插件 Event 对象	118
3.6.1 Windows 对象访问属性	80	6.1.2 创建对象的事件过程	119
3.6.2 其他 Windows 对象成员	81	6.1.3 使事件过程做出响应	120
3.6.3 创建一个 VB 工具窗口	81	6.1.4 刷新事件句柄	121
第 4 章 工程级插件对象和成员	83	6.1.5 编写插件事件过程程序	122
4.1 工程的 References 集合	83	6.2 与事件相关的插件的使用	123
4.1.1 Reference 对象的属性	83	6.2.1 有关 Reference 对象自动化的想法	123
4.1.2 增加、删除一个 Reference 对象	84	6.2.2 有关 VBProject 对象自动化的想法	124
4.2 其他工程级成员	85	6.2.3 有关 VBComponent 对象自动化的想法	124
4.2.1 保存一个工程	86	6.3 Controls Monitor 插件	125
4.2.2 编译一个工程	87	6.3.1 设置控件的 Name 属性	126
4.3 增加、RemoveVB 组件	87	6.3.2 对 Name 属性更改的反应	129
4.3.1 增加由 VB 创建的模块	88	6.3.3 剪切、拷贝并粘贴控件/代码	131
4.3.2 增加一个 ActiveX 设计器模块	88	6.3.4 更改 TabIndex 和 Index 设置	134
4.3.3 指定启动模块	88		
4.4 操纵 VB 组件	89	第三部分 插件程序专题研究	
4.4.1 非设计器访问属性	89	第 7 章 插件程序和对话框	135
4.4.2 与设计器有关的成员	90	7.1 Dialog Builder 插件程序	135
4.4.3 文件操纵成员	90	7.1.1 Dialog Builder 如何工作	136
4.4.4 VBComponent 对象的其他成员	92	7.1.2 由 Dialog Builder 创建的 MsgBox 代码	137
4.5 用 VBForm 对象操作设计器	92	7.1.3 Dialog Builder 中 MsgBox 的代码	141
4.6 控制 VBControls	94	7.1.4 Dialog Builder 的其他功能	145
4.6.1 容器控件和 ZOrder 方法	94	7.2 Programmer's Log 插件程序	147
4.6.2 其他 VBControl 对象成员	95	7.2.1 建立一个 ActiveX 文档	148
第 5 章 与属性和代码相关的插件对象	98	7.2.2 显示一个 ActiveX 文档	148
5.1 处理属性	99	7.2.3 关闭/打开一个 ActiveX 文档	150
5.1.1 Property 对象的属性	100	第 8 章 ActiveX 控件的 Tutor 插件程序	152
5.1.2 设置返回对象的属性	101	8.1 设计 Tutor 插件	153
5.1.3 设置包含数组的属性	102	8.1.1 Tutor 插件的 FRM 模块	153
5.2 处理代码	103	8.1.2 Tutor 程序的 RES 模块	156
5.2.1 增加和格式化代码	103	8.1.3 其他设计考虑点	157
5.2.2 Code Module 对象的方法	104	8.2 Tutor 插件程序如何工作	159
5.2.3 Code Module 对象的属性	106		
5.3 处理成员	108		
5.3.1 Member 对象的种类	108		
5.3.2 Member 对象的使用	109		