HZ BOOKS

碁峰
www.gotop.com.tw

Microsoft Visual Basic 2005 Express Edition

微软公司资深顾问及讲师章立民作品

Visual Basic 2005

程序开发与界面设计秘诀

章立民研究室 著

机械工业出版社
China Machine Press

Visual Basic 2005

程序开发与界面设计秘诀

章立民研究室 著

机械工业出版社
China Machine Press

本书综合讲解了 Visual Basic 2005 程序开发与界面设计的相关知识。全书共分 9 章，包括应用程序的基础生成技巧、面向对象程序设计、程序开发技巧、泛型、用户界面的设计与开发、控件的设计技巧、列表类型控件的设计技巧、人机界面的设计技巧以及工具栏、菜单及状态列等内容。书中包含有大量范例，内容全面，结构合理，论述清晰，对 VB 编程技术及其实际应用都有独到见解，是一本专业性较强的计算机书籍。本书可作为专业编程人员的参考书籍，也适合于对 Visual Basic 2005 有一定了解且想深入研究的读者。

本书版权登记号：图字：01－2006－5435

图书在版编目(CIP)数据

Visual Basic 2005 程序开发与界面设计秘诀/章立民研究室著. －北京：机械工业出版社，2006.9

ISBN 7-111-19918-9

Ⅰ.V…　Ⅱ.章…　Ⅲ.BASIC 语言－程序设计　Ⅳ.TP312

中国版本图书馆 CIP 数据核字(2006)第 110549 号

机械工业出版社(北京市西城区百万庄大街 22 号　邮政编码　100037)

责任编辑：陈冀康　李南丰

北京牛山世兴印刷厂印刷 · 新华书店北京发行所发行

2006 年 9 月第 1 版第 1 次印刷

186mm×240mm · 37.75 印张

定价：79.00元（附光盘）

凡购本书，如有倒页、脱页、缺页，由本社发行部调换

本社购书热线：(010)68326294

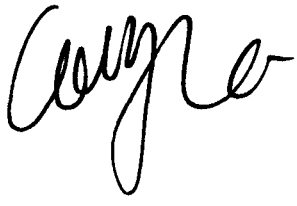
推荐序

Preface

各位亲爱的读者朋友们，大家好。我是微软亚太及大中华地区最有价值专家及技术社区总监——柯淑芬。非常高兴能够在这里与大家见面，并感谢大家购买本书。

本书的作者 章立民先生是一位知名的技术作家。他与中国台湾微软公司的合作时间长达 14 年以上，图书编写经验更有 17 年以上，他对微软开发工具与数据库管理系统等技术都有非常深入的研究。近两三年来，更积极参与微软技术社区的相关活动，不吝将其研究的心得与大家分享。正因为这股热忱和奉献的精神，他获选为微软最有价值专家(MVP)。

MVP 是名符其实的专业精英，他们不仅精通于某项微软产品及技术，更重要的是，他们非常乐于助人且不求回报。到目前为止，全球共有将近三千名来自八十多个国家的人士获选为 MVP。特别值得一提的是，中国台湾地区目前有将近 100 位人士获得 MVP 这一项殊荣。时至今日，MVP 对于微软公司产品的用户和技术社区已产生重大且不可磨灭的影响力，从操作的细节、乃至于全方位的策略性规划，MVP 都提供完善的协助和技术支持。微软技术社区是一个完全免费的技术咨询园地，微软产品任何使用上的问题，都可以在此询问，并与 MVP 们充分地讨论和交流。而在解惑与提升技术的同时，您也许就是下一位 MVP 的最佳候选者。期待在微软技术社区与您见面。我们下次再见。



微软技术社区及最有价值专家
亚太及大中华地区
区域总监
柯淑芬　Cally K

序

Preface

——心灵探索

Wise 介绍我的书还没看完，不过却已立即对我产生不少启发，开始自我探索起来。从昨天开始，我思考着，为什么我会选择作者这条路，待在家里工作。所有认识我的人，都觉得我不像。我爱讲话，善于演讲，“拍马屁”功力更是一流，男女老少不拘。记得以前陪微软的业务人员去各大企业 Pre-Sale，连业务人员都说我看起来根本就是个业务人员，怎么看也不像作者。就像我常说，我不喜欢跟工程师在一起，因为太没意思了。难怪了，连我都不禁要问自己，为什么会在家工作，做的还是技术钻研之极其枯燥的事物，甚至不与人接触，是什么因素潜在影响了我呢？我曾经开过公司当过总经理，不过痛苦到半夜作恶梦，半年不到，投资的钱都不要就跑了，为什么呢？回忆自己的成长过程，我恍然大悟。

原因是我的父亲吧！我的记忆力其实很好，虽然父亲已经过世 30 年，但是从他平日身体不适、发病住院、转院到台北荣总、尔后过世，以致于盖棺入殓前躺在棺材里穿着寿衣的样子，这一切我至今都清晰记得。一直以来，我都不认为没有父亲对我有什么影响。小时候我们常说，因为爸爸太忙，平常根本不在家，所以即使他过世了，我们也不觉得少了一个人。话虽如此，其实成长的过程是辛苦的，只是心里觉得没什么。不过真的没什么吗？到了大学的时候，我猛然发现其实有什么。

记得大二时搬出宿舍与同学在外租房子，我室友的父母来探望他，看看居住环境如何，当时我突然一阵心酸，觉得我好像少了什么。确实，没有老爸不说，我老妈也才车祸不久。或许吧！其实丧父的阴影一直藏在我的潜意识里，让我觉得不能再像他一样老是不在家，因此虽然我是个叛逆分子，却选择了一个要成天待在家的工作，原因很简单，我希望能够一直陪伴我的家人，守护着他们。唉！我对小孩关心的程度，甚至已经造成别人的压力。

话说回来，在家工作，做个作者，书卖得再好，也不可能成为真正的有钱人。我现在认为，我应该抛弃潜意识的束缚，大步走出去，先成立一个团队，开个公司，建立一个横跨语言与国界的出版与数字教学企业帝国。我想，我也不应该只在 IT 出版这条路上发展。我跟 Danny 提过一个数字学习的点子，曾经建议过 Wise 应该开个咨询公司，这些想法，不应该只是说说而已。

你给自己多久的时间成为一位富翁呢？你希望年收入多少呢？我已经写下这两个问题的答案。接下来，就是行动。

章立民

2006 年 2 月 27 日

作者简介：

章立民

微软公司资深顾问讲师。从1992年开始于台湾微软主讲研讨会

台湾微软最有价值专家MVP，连续四度当选MVP

资深计算机图书作家，潜心技术创作17年，拥有60本以上著作

专长：

关系型数据库管理系统

Visual Studio 2005开发工具

SQL Server、Access 2003等

著作：

迄今为止，章立民先生已有60余部计算机著作问世，内容涵盖SQL Sever 2000、Visual Basic.Net、ASP.NET、Crystal Report for Visual Studio.NET、Access 2002-2003、Visual FoxPro、Word VBA、Windows等

章立民老师博客：

www.cnblogs.com/liminzhang

章立民研究室简介

章立民先生是台湾非常知名和资深的技术作家。他在台湾微软公司担任顾问与讲师的时间长达15年，对微软开发工具与数据库管理系统等技术都有非常深入的研究。章立民先生不仅拥有丰富的实务经验，还经常与第一线的开发人员接触并交流，深知从业人员的真正需求。因此，不仅他的图书著作受到大家的肯定与喜爱，他所主讲的研讨会更是深获好评。

为了能够质与量并重，撰写出更多的优质书籍，并为两岸信息文化事业贡献一份心力，章立民先生在2006年初正式成立了“章立民研究室”并担任技术总监，亲自审校所有书籍，期许通过团队的力量，提供更多的好书给广大的读者；也希望借此促进良性竞争，提升信息图书的整体质量。

章立民研究室的所有成员有几项特色：第一，在相关信息领域必须有六年以上的实务经验；第二，必须参与或主持数项具代表性的项目；第三，必须同时专精Visual Basic与Visual C# 程序设计语言；第四，对数据库技术必须有相当程度的认识；最后一项，就是必须拥有高度的热诚以及无私分享的精神。

目前，“章立民研究室”的成员中，有人甚至已在美国从事项目开发10年以上，并且已经具备项目经理人的职位。他们在章立民先生的号召之下，投入研究室的写作行列。如此高水平的成员，让广大读者对“章立民研究室”拥有极高的期待。

封面设计：王冰

目 录

Contents

第1章

应用程序的生成技巧

条款 1　如何生成可执行文件 . EXE

您可以采用下列两种方式来生成可执行文件 . EXE：

- 方法一：在 Visual Studio 2005 的集成式开发环境中，从“生成”菜单中选择“生成”，即会在项目的 \ bin 文件夹中生成 . EXE 文件。
- 方法二：在 SDK 命令提示符窗口下，运行 vbc 命令来编译 . EXE 文件。

条款 2　如何指定 . EXE 的输出位置

虽然可执行文件 . EXE 默认会保存到项目的 \ bin 文件夹中，但是您可以依下列步骤来指定其输出位置：

1. 在 Visual Studio 2005 的集成开发环境中，启动您的 Visual Basic 2005 项目。

2. 在“解决方案资源管理器”中执行下列操作之一：

- 双击 My Project 选项。
- 在资源管理器窗口中单击鼠标右键，选择快捷菜单中的“属性”选项。

3. 单击左侧的“编译”索引标签。

4. 如图 1-1 所示，在“生成文件输出路径”文本框中输入您所希望的输出位置，或是单击“浏览”按钮来选择其他输出位置。值得一提的是，如果您希望将 . EXE 输出至项目的根目录下，可以将此框清除成空白。

5. 单击右上角的“关闭”按钮。

条款 3　如何设定启动对象

启动对象就是当加载应用程序时所要调用的进入点(Entry Point)。一般来说，我们会将启动对象设定成应用程序的主窗体，或是当应用程序启动时所会运行的 Sub Main 程序。值得注意的是，类库项目与 ASP. NET Web 应用程序项目都没有进入点，因此没有启动对象。

欲给一个 Visual Basic 2005 的 Windows 应用程序项目设定启动对象，请依下列步骤进行：

1. 在 Visual Studio 2005 的集成开发环境中，将您的 Visual Basic 2005 项目

图 1-1 设定生成文件输出路径

打开。

2. 在“解决方案资源管理器”中执行下列操作之一：

➢ 双击 My Project 选项。

➢ 在资源管理器窗口中单击鼠标右键，选择快捷菜单中的“属性”选项。

3. 单击左侧的“应用程序”索引标签。

4. 如图 1-2 所示，在“启动窗体”下拉列表中选取您希望作为启动窗体的对象。如果您希望以 Sub Main 程序作为启动对象，请如图 1-3 所示，取消勾选复选框“□启用应用程序框架”，并从“启动对象”下拉列表中选取 Sub Main 窗体或 Sub Main 程序。

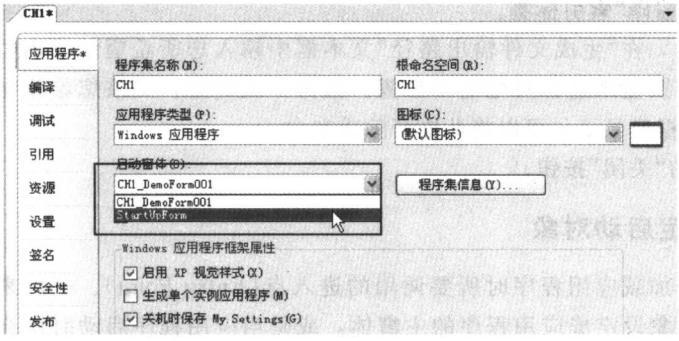


图 1-2 选取启动窗体对象

5. 单击右上角的“关闭”按钮关闭窗体。

本书各章的应用程序项目都是以主窗体 StartUpForm. vb 作为进入点，因此都是从“初始窗体”下拉列表中选取 StartUpForm 窗体。不过为了示范如何以 Sub Main 程序作为进入点，第 1 章的应

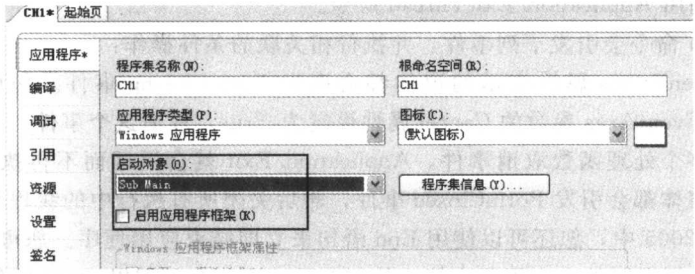


图 1-3 取消勾选“启动应用程序框架”

用程序项目 CH1.vbproj 特别将启动对象设定成 Sub Main 程序，并特别在一个模块文件 MainModule.vb 中编写如下的 Sub Main 程序。

```
Module MainModule
    Sub Main()
        Application.Run(StartUpForm)
    End Sub
End Module
```

上述写法是一种常规的写法，那就是把启动窗体传递给 Application.Run 方法。

值得注意的是，当您使用一个自定义的 Sub Main 程序作为应用程序的启动对象时，您在应用程序事件(Startup、Shutdown、StartupNextInstance 与 UnhandledException)中编写的程序代码将不会被运行。

附注 关于如何编写应用程序事件，请参阅第 3 章的说明。

Main 程序是应用程序的起点，并负责应用程序的整体控制。基本上，Main 程序共有下列四种形式：

- Sub Main()
- Sub Main(ByVal cmdArgs()As String)
- Function Main()As Integer
- Function Main(ByVal cmdArgs()As String)As Integer

最常见的 Main 程序形式就是 Sub Main()。如需更详细的信息，请查阅在线帮助，在此不再赘述。

条款 4 关闭窗体与结束应用程序

如果您只是要关闭窗体，请调用该窗体的 Close 命令。因此我们经常在窗体中的“关闭”按钮的 Click 事件处理函数中编写下列程序代码：

```
Me.Close()
```

如果您将应用程序项目的启动窗体设定成某一个窗体，则调用该启动窗体的 Close 命令时，也就会结束应用程序。

一般来说，要在任何时候结束应用程序，请调用 Application.Exit 命令。Application.Exit 命令会停止所有运行中的线程，并关闭应用程序的所有窗口。Application.Exit 命令并不一定会强制结束应用程序。Application.Exit 命令一般是在线程内调用，并强制 Application.Run 返回。若只是要

结束当前线程，则调用 Application. ExitThread 命令。

Application. Exit 命令会引发下列事件，并执行相关联的条件操作：

- 每一个以 OpenForms 属性表示的窗体都会引发 FormClosing 事件。您可以通过将事件的 FormClosingEventArgs 参数的 Cancel 属性设定为 True，取消这个事件。
- 如果一个或多个处理函数取消事件，Application. Exit 就会返回而不再执行操作。否则，每一个打开的窗体都会引发 FormClosed 事件，然后关闭所有执行中的线程和窗体。

在 Visual Basic 2005 中，您还可以使用 End 语句来立即结束应用程序。您可以将 End 语句摆在程序的任何位置，以便强制整个应用程序停止执行。End 语句会关闭以 Open 语句打开的任何文件，并删除应用程序的所有变量。一旦没有其他程序使用这些数据时，而且也没有任何程序代码在执行时，应用程序就会立即关闭。

End 语句会突然停止程序代码的执行，而且不会调用 Dispose 或 Finalize 命令，亦不会调用任何其他的 Visual Basic 程序代码。其他程序所使用的对象引用将会失效。如果 End 语句是位于一个 Try 或 Catch 语句块中，控制权将不会传递至相对应的 Finally 语句块。

Stop 语句会暂停程序的执行，但它与 End 语句不同的地方是，它并不会关闭任何文件或删除任何变量(除非是在编译过的可执行文件 . exe 中使用过)。

因为 End 语句在终止应用程序时并不理会任何可能已打开的资源，因此在使用它之前，您应该尝试彻底地关闭所有资源。例如，如果应用程序已打开了一些窗体，则您应该在控制权移交给 End 语句之前先将它关闭。

显然，在使用 End 语句时应该非常谨慎，而且只有在需要立即停止时才使用它。终止程序的正常方法(Return 语句与 Exit 语句)不只会彻底地关闭程序，而且还会提供调用相关的函数进行彻底清除操作。

End 语句会调用 System 命名空间的 Environment 类型的 Exit 命令。您必须拥有 UnmanagedCode 权限才能调用 Environment. Exit 命令，如果您并未拥有该权限，将会引发 SecurityException 异常。

在 End 语句之后加上其他的关键字时，它会描述适当程序或语句块定义的结束。例如，End Function 是 Function 函数体的结束。

第2章

浅谈面向对象程序设计

条款5 概论

面向对象程序设计(Object Oriented Programming，简称OOP)绝对是Visual Basic 2005非常重要的一项变革，本节我们将简介此特性，以便让您对Visual Basic 2005的面向对象程序设计技术有一个基本的认识，并为后续的学习奠定良好的基础。

一种程序语言要成为真正的面向对象程序设计语言，它必须符合下列条件：

- 抽象(Abstraction)：抽象能够有效管理问题的复杂性，其作法是划分出与该问题相关的一组对象。
- 封装(Encapsulation)：封装是指将一个抽象的内部实现隐藏在特定的对象之内。
- 多态(Polymorphism)：多态会提供相同方法的多种操作。例如，不同的对象都会拥有一个Save函数，但是每一个Save函数会执行不同的功能。
- 继承(Inheritance)：Visual Basic 2005最令人兴奋之处就是其“继承”特性。Visual Basic 5.0首次引入了接口继承的概念，它允许您重复使用类的接口，但是不能重复使用其方法。Visual Basic 2005则提供了真正的方法继承，因此您可以重复使用类的方法。

首先我们要特别声明的是，Visual Basic 2005已经全部重新改写使其成为真正的面向对象语言。事实上，Visual Basic 2005中的每一个项目都可以被视为一个对象，即使是字符串与整数也可以在Visual Basic 2005中当作对象来访问。例如，您可以在Visual Basic 2005中编写下列程序代码：

```
Dim i As Integer
MessageBox.Show(i.MinValue)
```

整数i会被认为是一个对象的参数，就是当您输入i之后的点号(.)时，就会显示出属性与方法的菜单(如图2-1所示)。请从中选取一个属性，例如，如本例一样选取MinValue属性。运行应用程序时，您将会见到一个对话框并在其中显示出所选取的整数属性值。

.NET已经给自带的数据类型事先定义了相对应的类，但是如何自定义一个类呢？现在我们将以逐步解说的方式来示范如何创建类，然后加以继承，以

便充分利用 Visual Basic 2005 全新的 OOP 功能。

定义一个类

类的基本用途在 Visual Basic 2005 中并没有改变。您仍然可以为对象或是应用程序所需的任何支持性对象创建类。从 Visual Basic 6.0 到 Visual Basic 2005 的主要改变只是涉及语法以及一些新增的功能。

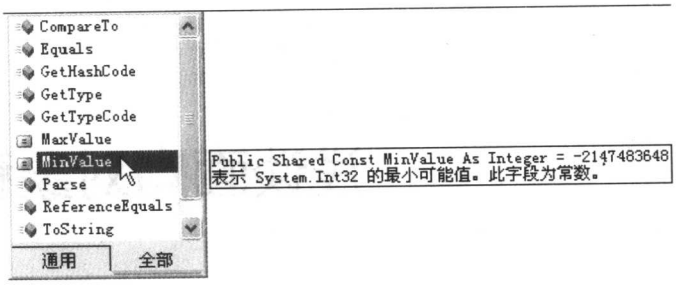


图 2-1 整数也会被视为是一个对象

在 Visual Basic 6.0 中，您会通过创建一个类模块来创建一个类：也就是每个类都需要一个类模块。此项限制在 Visual Basic 2005 中已不复存在，现在，您可以在单一程序源代码文件中创建任意数目的类。您甚至还可以在类的内部创建类(也就是嵌套类)。不过呢，现在先让我们从一个简单的范例开始。

将一个类加入到 Visual Basic 2005 项目中的方式与 Visual Basic 6.0 非常类似，然而，与过去出现的是一个空程序源代码文件不同的是，您的类一开始将会出现下列程序源代码：

```
Public Class Customer
End Class
```

如果您想要在同一个文件中加入第二个类，只需要如下所示加入另外一个类语句：

```
Public Class Customer
End Class
Public Class Contact
End Class
```

在此要提醒大家，虽然同一个程序源代码文件中可以含有多个类，但是一般来说，每一个类还是会单独定义于各自的程序源代码文件中。我们只有在各个类彼此是紧密关联时才会将它们摆在同一个程序源代码文件中。例如，发票与发票细目这两个类就非常适合摆在同一个程序源代码文件中，因为一般来说您不可能只使用发票明细目而没使用到发票。如果您想要将客户联系人(Contact)与客户(Customer)分开使用，则 Contact 类与 Customer 类就应该分开存放在各自的程序源代码文件中。

接下来，您就可以开始为类添加属性(Property)与方法(Method)。在 Visual Basic 6.0 中，您通常会通过由声明一个私有变量与公共属性程序来定义一个属性。然而在 Visual Basic 2005 中，您必须如下所示来定义一个属性(此属性的名称为 Name)：

```
Private m_sName As String
Property Name()As String
    Get
        Return m_sName
    End Get
    Set(ByVal Value As String)
        m_sName = Value
    End Set
End Property
```

在 Visual Basic 2005 中只有两种类型的属性程序，它们分别是 Get 与 Set。Get 程序会从类中提取出属性值，Set 程序则会将值赋给类的属性。Visual Basic 6.0 提供一个属性 Let 语句来处理自带

的数据类型，至于 Set 语句则用来处理对象，不过在 Visual Basic 2005 中，由于所有的项目基本上都是一个对象，因此不再需要 Let 语句。

请注意到属性程序的语法也已经改变。Get 与 Set 会包含在单一属性语句中。在属性 Get 与 Set 之间不会再发生数据类型的冲突，此举使得这些语句更容易去维护。

一个简单方法(Method)的语法几乎与先前版本的 Visual Basic 相同，您所必须注意到的差别就是 Return 语句。您现在可以使用 Return 语句从一个函数中将值返回，而不需要使用函数名。以下的程序代码示范如何建立一个简单的方法：

```
Public Function SayHello()As String
    If Name <> "" Then
        Return "嗨!" & Name
    Else
        Return "嗨!大家好! "
    End If
End Function
```

到目前为止，您已经创建了一个类，它拥有一个属性与一个方法，而且它看起来与 Visual Basic 6.0 的类还挺相似。现在我们来看看 Visual Basic 2005 在 OOP 方面有哪些新功能。

创建构造函数与析构函数

在 Visual Basic 6.0 中，当您创建一个类的对象时，Initialize 事件就会被引发，因此您可以在 Initialize 事件中编写用来初始化对象的程序代码。例如，您可以在此处定义默认的对象数据、启动数据库连接、建立相关联的对象 ... 等等。然而，您不能将任何数据传递给 Initialize 事件，此项限制让我们很难以特定的参数去初始化组件。

Visual Basic 2005 引入了真正的构造函数(Constructor)，每当类的对象被创建时，其构造函数就会被执行。这些构造函数是使用一个名称为 New 的子过程所定义的，如下所示：

```
Public Sub New()
    '在此进行初始化操作
    Debug.WriteLine("我已经被创建 ...")
End Sub
```

重要的是，您可以将数据传递给构造函数，使得在初始化对象时享有更高的弹性与更强的功能。带有参数的构造函数称为参数型构造函数，如下所示者即是一例：

```
Public Sub New(ByVal sName As String)
    '将传递进来的参数赋给 Name 属性
    Name = sName
    '进行其他的初始化操作
    Debug.WriteLine(Name & " 对象已经被创建.")
End Sub
```

在本范例中，客户名会被传递至构造函数中，然后使用该名称来初始化使用 Property 程序所定义的 Name 属性。

在此之前所列的两个构造函数可以同时定义在一个类中。事实上，您可以为同一个类定义任意数目的构造函数，只要这些构造函数带有不同的参数即可，此项特性也就是所谓的“重载”(overloading)。当类的对象被创建时，会根据传递给构造函数的数据来调用合适的重载构造函数。

构造函数不是必须的。如果您并未创建任何构造函数，系统会自动使用一个默认构造函数。

至于 Terminate 事件也已经被取代，Visual Basic 2005 现在提供了一个 Finalize 析构函数