

autodesk
authorised developer

Autodesk授权出版社

Inventor R6

VB(A)程序设计

——技巧与范例

资深专家

深入浅出

覆盖面广

附加光盘

Autodesk Inventor

autodesk

6

陈伯雄 张朝阳 编著

 **机械工业出版社**
CHINA MACHINE PRESS



Inventor R6 VB(A)程序设计

—— 技巧与范例

陈伯雄 张朝阳 编著

机 械 工 业 出 版 社

这是一本关于在 Inventor R6 中, 使用 VBA 编写应用程序的参考书。本书共 13 章, 由浅入深, 全面地介绍了 Inventor-VBA 的程序设计基础、规则和具体的功能。

本书是作者编程解决工程实际应用问题经验和技巧的珍贵结晶。每章都列出了应用程序范例, 并作了详尽的程序结构分析。对于某些 Inventor 的功能缺口, 笔者也提供了可能的修补程序, 可以供读者直接使用。

本书从最基础的内容说起, 并延伸到相当深入的程序。因此, 对于想了解和掌握 Inventor-VBA 程序设计的读者, 这是一本很好的教材或参考书。

图书在版编目 (CIP) 数据

Inventor R6 VB(A)程序设计: 技巧与范例/陈伯雄, 张朝阳编著.

—北京: 机械工业出版社, 2004.1

ISBN 7-111-13230-0

I. I... II. ①陈... ②张... III. 机械设计: 计算机辅助设计—应用软件, Autodesk Inventor R6 VB (A) ~ IV. TH122

中国版本图书馆 CIP 数据核字 (2003) 第 094101 号

机械工业出版社 (北京市百万庄大街 22 号 邮政编码 100037)

责任编辑: 周国萍 黄丽梅

封面设计: 鞠 杨 责任印制: 路 琳

北京机工印刷厂印刷·新华书店北京发行所发行

2004 年 1 月第 1 版第 1 次印刷

1000mm×1400mm B5·12.375 印张·480 千字

0 001—4 000 册

定价: 39.00 元 (含 1CD)

凡购本书, 如有缺页、倒页、脱页, 由本社发行部调换

本社购书热线电话 (010) 68993821、88379646

封面无防伪标均为盗版

前 言

Inventor 是 Autodesk 公司软件产品家族中比较新的一员，也是比较出色的一员。它象 Autodesk 公司的其他产品一样也支持开放的应用程序接口 (API)。本书就从浅入深介绍 Inventor API 的结构，并以 VB(A)语言为例介绍一些实际应用和技巧。

一句话讲，Inventor API 是一个组件对象模型 (COM) 或者叫 ActiveX 接口。这种接口为大多数的流行语言所支持，例如 VB、VBA、VC++、Delphi、Java 等等。此接口一旦完成，就可以提供给所有这些语言一个一致的编程接口，减少错误的发生从而降低工作量。另外一个更重要的特点是它可以在不同版本之间保持代码的兼容性，从而降低维护费用。VBA 是一种最常用的支持 ActiveX 编程的语言。因此，我们往往用某一个产品所提供的 VBA 来代表它的 ActiveX 接口，并用 VBA 来说明和示范此 ActiveX 接口及其相应的对象模型。

实际上，Inventor API 支持两个 COM 对象模型，一个是学徒服务器对象模型，另一个是完整的 Inventor 对象模型。学徒服务器对象模型提供对于 Inventor 图形文件和对象的一些基本的访问，大部分是只读的，它独立存在，不依赖于 Inventor 产品的安装。从本质上来说学徒服务器是一个 Inventor 的子集，它没有用户界面，与学徒服务器交互的唯一方式是通过它的 API。Inventor API 暴露 Inventor 产品的各种对象的内在功能。Inventor 对象模型包含 Inventor 产品的大部分功能，而且是读写访问。本书对这两种对象模型都有介绍，并且都有具体的应用实例。

因为这是在讨论一种很新的技术方法，本书从最基础的 VB(A)概念、常用语法、函数，到 Inventor 学徒服务器，到 Inventor API 的草图和图档操作，再到比较复杂的插件生成、用户界面设计、交互操作、各种事件的处理。最后一章还提供了一些颇具实用价值的典型程序。有些稍加改动，即可应用到实际的机械工程设计中去。因此，对于想了解和掌握 Inventor-VBA 程序设计的读者，这是一本很好的教材或参考书。

书中的每一章节都有生动的图片和代码说明相应的 API 部分。所有的代码都汇集起来，放在本书所附光盘之中，并且大部分复杂的代码和实例还提供相应的测试图形。

本书的两位作者，都是精通 Autodesk-API 和 Autodesk 软件产品的资深工程师。他们的观念和经验，对于读者应当具有指导性的作用。如果在

IV

Inventor 的具体使用中还有什么问题,请参考机械工业出版社出版的《Inventor R6 机械设计应用 —— 技巧与范例》,相信读者会找到答案的。

编者

2003 年 7 月

目 录

前言

第 1 章 概论	1	第 5 章 学徒服务器 (Apprentice Server) ...	148
1.1 Inventor API 总论	2	5.1 简介	148
1.2 Inventor API 种类	3	5.2 特点	148
1.3 Inventor VBA 编程基础 ...	6	5.3 与 Inventor API 的不同之处	149
1.4 几个 Inventor VB(A) 入门程序	21	5.4 与 Inventor API 的相同之处	157
第 2 章 通用 VB(A)基础知识	26	第 6 章 Inventor 文件的属性	159
2.1 工程	26	6.1 在 API 中访问文件属性	159
2.2 代码模块	27	6.2 创建属性	166
2.3 使用资源文件	31	6.3 修改属性	168
2.4 VB(A)常用功能小结	36	6.4 删除属性	170
第 3 章 关于草图	48	6.5 保存属性	171
3.1 在 API 中创建草图的特点 ...	48	第 7 章 插件	172
3.2 在 API 中创建简单草图线 ...	50	7.1 ActiveX 部件和 插件(Add-In)	172
3.3 在 API 中创建数学曲线	57	7.2 体验 SimpleAddIn 插件的使用	172
3.4 草图创建小结	59	7.3 创建 EXE 或 DLL 插件	175
3.5 草图在零件环境中的应用 ...	60	7.4 插件的注册	182
3.6 关于特征的轮廓	65	7.5 用工具进行插件的 注册和卸载	186
3.7 工程图中的草图	68	7.6 插件版本控制	189
3.8 草图应用小结	80	7.7 插件调试	190
3.9 Inventor VBA 初显身手	81	第 8 章 用 API 控制 Inventor 界面	194
第 4 章 图档	92	8.1 用户界面定制 API	197
4.1 零件和特征	92		
4.2 边界映像和几何对象	121		
4.3 装配件	126		
4.4 工程图明细表	144		

8.2 基对象（基命令条和基环境）	208	12.3 在窗口中显示模型	301
8.3 浏览器集成	214	12.4 文档子类型	303
第9章 用户交互操作	223	12.5 打印到文件	304
9.1 选择集	223	12.6 客户图形文本	304
9.2 交互选择	225	12.7 装配件延时更新	306
9.3 鼠标和键盘事件	236	12.8 缩略图查看	307
9.4 状态栏文本	237	12.9 文件工具函数	308
9.5 文件对话框	237	12.10 自定义函数	309
9.6 综合实例分析	237	第13章 典型程序实例分析 ...	314
第10章 事件	260	13.1 显示文件属性(T001.IVB)	314
10.1 应用程序事件	260	13.2 显示命令条成员名(T001a.IVB)	317
10.2 文件访问事件	268	13.3 创建渐开线齿沟草图轮廓(T002.IVB)	319
10.3 交易事件	272	13.4 关于简单弹簧的创建(T004a.IVB)	332
10.4 文件界面事件	276	13.5 弹簧造型程序(T004.IVB 详解)	338
10.5 文档事件	280	13.6 圆柱凸轮造型(T005.IVB)	350
第11章 材料和渲染样式	286	13.7 关于质量属性的对策和补丁	359
11.1 材料	286	附录 A VB(A)常用语法和函数速查	365
11.2 渲染样式	287	A.1 基本语句	365
11.3 生成和复制渲染样式	289	A.2 基本函数	370
11.4 设置渲染样式的参数	291	A.3 导出的数学函数	386
11.5 使用渲染样式	292		
11.6 综合示例	295		
第12章 杂项	297		
12.1 交易	297		
12.2 生成位图	300		

第 1 章 概 论

众所周知，Autodesk 公司的软件以其开放性著称于世，作为其产品家族中的一员，Inventor 也同样秉承了此一贯的优良传统。理解和掌握 Inventor API，对于充分地发挥软件功能，更好地完成专业设计等，将起到十分重要的作用。

在 Inventor 中，嵌入了完整的 VBA 程序设计模式，这是一个功能强大的开发工具。本书将以 Inventor R6 中文版为基础，展开关于 Inventor VBA 以及相关技术的讨论。

严格上来说，VBA 并不是某一个产品所独有的开发工具。实际上它是微软的一种编程语言，它的全称是 Visual Basic for Applications，意思是为应用程序而量身定做的 Visual Basic。因此，可以把 VBA 理解为 VB 的一个子集，它集成了大部分 VB 的常用功能。它们两者之间的一个重要的区别在于某一个产品（例如 Inventor）的 VBA 程序只能运行在该产品内部，而 VB 则可以生成独立的可直接运行的 EXE 文件。VBA 作为一种易学易用的程序设计工具，存在于 100 多种软件中。典型的有微软公司的 Office 套件（包括 Word, Excel 和 PowerPoint）和 Outlook 等，和 Autodesk 公司的 Inventor、AutoCAD、ADT、MDT 等。就像在其他软件中使用 VBA 一样，在 Inventor 中使用 VBA，除了要了解 VB 程序设计的一般性知识之外，还要学习相关的 Inventor API 的规则和知识。由于 Inventor R6 的 VBA 文档尚未本地化，本书也可以作为 Inventor VBA 程序设计的参考书。好在 VB 的支持文档已经被微软完整地本地化，许多基础功能的参考可以在那里找到。

从更底层来说，不管是 VBA 或 VB，还是任何其他编程语言，如果我们想用它们来定制 Inventor，那么必须通过 Inventor 所提供的 ActiveX 接口来访问和控制 Inventor 中的各种对象、属性、方法和事件。ActiveX 或者后来的 COM 是微软的一项通用的编程技术，它用于在不同程序之间通信和信息共享。简单来说，Inventor 的 ActiveX 开发接口就是把产品的对象和功能根据一定的规则暴露给支持 ActiveX 编程的语言。VBA 是一种最常用的也可以说是只支持 ActiveX 编程的语言。因此，我们往往用某一个产品所提供的 VBA 来代表它的 ActiveX 接口，而且也用 VBA 来说明和示范此 ActiveX 接口及其相应的对象模型。由于以上的原因，加上本书中也涉及到了很多 VB 程序，所以我们在书名中使用了 VB(A)这个称呼。在后面的章节中，我们

将使用 VB 或者 VBA 来特指当时使用的编程语言，而不使用这个笼统的称呼。

Visual Basic 是一个在世界上应用最广的应用程序快速开发工具。它是一个可以创建包括可执行程序, ActiveX 控件和 COM 部件的独立软件。Visual Basic 必须单独购买。它具有三种版本: 学习版、专业版和企业版。专业版和企业版都适合用来定制 Inventor。学习版拿掉了一些功能, 因此也就不能充分利用 Inventor API 的所有功能。VBA 不能创建独立运行的程序包, 也不能创建插件程序, 它的代码只能在 Inventor 内部运行。因此, 为了充分利用 Inventor API 的完全功能, 应当使用 VB 来作为主要的开发工具。当然也可以同时使用 VBA 来作为辅助工具, 因为它更容易被工程师学习和使用。更重要的是它随 Inventor 产品提供, 而不需要单独购买和安装。

在阅读本书并参看所提供的程序前, 请安装好下面的软件:

- Inventor R6
- Microsoft Visual Basic 6
- MSDN Library Visual Studio 6.0

以上是必要的软件环境。另外, 本书的程序是在 Win2000 中文专业版+SP3 的系统环境下创建和调试的。本书的附加光盘中, 分章列出了所有的相关程序。

1.1 Inventor API 总论

1.1.1 什么是 API

API (Application Programming Interface) 是一种功能, 它的全称是应用程序编程接口。广义地讲, 这种功能是把一个应用程序中的功能暴露出来, 供其他的应用程序使用。例如: 我们可以用 Inventor API 写一个程序来完成一些交互操作的自动化。在 Inventor 中使用 API 是相当有意义的, 在于能够添加一些满足特定专业设计需求的自定义功能, 这样 Inventor 就成为了“我的” Inventor。

另外, API 还允许第三方应用程序与 Inventor 集成在一起。例如: PDM 产品就可以通过 API 和 Inventor 通信。

注意: 根据 Autodesk 的声明, 在使用 Inventor API 时, 尽量不要使用任何被标志为隐藏、不支持或者仅供内部使用的对象、方法、属性、事件和常量。这些对象是由 Inventor 创建或维护的, 并且仅供内部使用。它们在新的版本中可能改变甚至被取消。

1.1.2 如何利用 API

Inventor API 的技术基础是微软的自动化（Automation）技术。这类自动化接口，在为 Windows 写的应用程序中是通用的。

这种 API 技术模式有许多显著的优点：

- 首先，几乎可以用任何当前流行的编程语言来写自己的程序。例如：VBA、Visual Basic、Visual C++、Delphi、Perl、Java 等等。

- 其次，因为使用了标准的模式和概念来暴露 Inventor 的功能，以至于能实现以下的结果：如果已经有了对 Word 或者 Excel 自动化接口的编程经验，那么实际上就已经理解了许多 Inventor API 的概念，它们在许多方面都很相似。

- 最后，因为是以面向对象的方式暴露应用程序的功能，所以，一旦理解和掌握了某些一般的概念（例如一个面向对象的 API 是如何工作的），这种 API 就比面向过程的 API 更容易学会和使用。

在 Inventor 中，可以有很多种不同的方式访问 API，因此有必要对这些不同的方式有一个基本的了解，然后才能决定如何编写你自己的程序。

在图 1-1 中，白色框代表 API 部件，包括 Inventor 和“学徒服务器”(Apprentice Server)。

图底部的灰色圆柱代表正在处理的 Inventor 数据（零件、装配件等等）。所有的灰色框代表用户自己写的程序。

如果一个框包含另一个框，就表示被包含的框所代表的程序与外面的框所代表的程序运行在相同的数据处理空间内。例如：VBA 就运行在与 Inventor 相同的处理空间内。

这种“内处理”的程序将比运行在外部空间的程序快得多，因为“外处理”程序需要消耗更多时间用于彼此之间的相互通信。

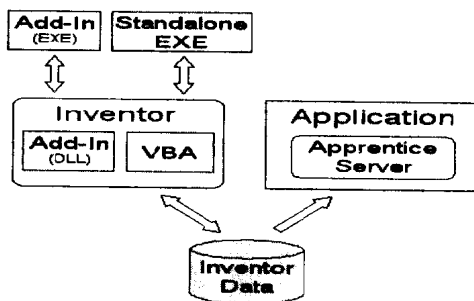


图 1-1 Inventor API 的逻辑关系

1.2 Inventor API 种类

1.2.1 VB(A)家族

Inventor 集成了微软的 Visual Basic for Applications (VBA)。Inventor VBA

是定制 Inventor 和集成 Inventor 与其他应用程序和数据的一个强大工具。用 VBA 写的程序经常被称为“宏”。Inventor VBA 可以说是访问和使用 Inventor API 的最便捷的开发工具。

1. VBA

VBA 具有 Visual Basic 编程系统的大部分功能，并且具有相同的集成开发环境。

VBA 随相应产品一起发送，不需要单独购买。

VBA 不能创建独立的应用程序，它必须运行在产品内部。

VBA 语言和编程环境在不同产品中都一样。如果已有其他产品的 VBA 的编程经验，那么只需要再学习 Inventor API 的对象模型即可。

2. Visual Basic Scripting (VBS)

VB 脚本语言用来为低内存需求的环境提供轻量的定制能力，例如网络浏览器。它大部分应用在创建 HTML 网页方面。VB 脚本语言不是很适合来为 Inventor 编写应用程序。

3. Inventor VBA

这是由微软提供并集成到 Inventor 中的一个特定于 Inventor 的开发工具。虽然 VBA 是通用的，但是它可以被定制，以满足集成它的特定应用程序的需求。以后将讨论 Inventor 集成的 VBA 的一些特有规则。关于 VBA 的通用功能的信息可以从微软提供的 VBA 帮助中得到。我们在第 2 章中也会介绍一些 VB(A) 的通用功能。

首选 VBA 来定制 Inventor 有很多好处：

- (1) VBA 随 Inventor 发行，不需要去购买额外的编程语言工具。
- (2) 可以把程序嵌入到 Inventor 文件中，从而使程序和数据绑定在一起。
- (3) 也可以把程序存成独立的.IVB 文件，供其他的用户和文件共享。
- (4) 它运行在 Inventor 的相同的处理空间，运行效率较高。

4. Visual Basic

VB 是一种通用的编程语言，它可以对任何支持自动化接口的产品进行编程。Inventor 支持自动化接口，于是我们当然可以用 VB 来定制 Inventor。实际上，由于 VBA 有一定的局限性，例如不能创建插件，我们在很多时候不得不借助于 VB 的强大功能。

1.2.2 插件

插件是一种特殊类型的 Inventor 程序，有几个特点：

- (1) 插件在 Inventor 启动时可以自动加载。
- (2) 插件可以创建新的命令。

(3) 插件与其他的方法一样可以访问和使用 Inventor API。

插件可以自动加载对于很多程序来说是一个非常有用的功能。大多与 Inventor 无缝集成的应用程序都需要被做成插件，这样它们的功能才能够始终被用户获得。几乎任何应用程序都需要创建新的菜单或者按钮，能够创建新命令对于专业应用程序来说也非常重要。

在图 1-1 中，Add-In 插件被列出了两次：一次作为 DLL，另一次作为 EXE。作为插件，可以选择生成 DLL，它将运行在 Inventor 相同的处理空间内；也可以选择生成 EXE，它将运行在单独的处理空间中。实际上几乎所有的插件都是 DLL，因为这样可以提高运行效率。至于 EXE 类型的插件，主要的好处是调试方便。插件可以用任何支持生成 ActiveX EXE 或者 DLL 的语言来生成，譬如 Visual C++ 和 Visual Basic，但是 VBA 不具备此种能力。

1.2.3 独立的 EXE

独立的 EXE 是和 Inventor 联系在一起的，运行在它自己的处理空间的一种应用程序。

这种程序典型的使用情况是：需要使用 Inventor，但是有自己的界面并且不需要用户在 Inventor 中做交互操作。例如，一个批处理出图工具，可以是一个运行在 Inventor 之外的 EXE。它监控数据库记录的添加，记录和描述出图文件。当一个新记录在数据库中生成时，如果 Inventor 尚未启动，这个程序就启动它，打开需要的图档并且开始打印。所有这些不需要与 Inventor 做任何的用户交互。

独立的 EXE 运行在 Inventor 之外的处理空间，因而会有一些效率上的损失。此外，当对另外一个软件写 VBA 程序时，它也运行在独立的运行空间。例如，如果在 Excel 的 VBA 内写一个程序来连结和控制 Inventor，这个 VBA 程序就运行在 Excel 的处理空间中，而不是在 Inventor 的处理空间中。

1.2.4 学徒服务器 (Apprentice Server)

它给运行在 Inventor 之外的应用程序提供了访问 Inventor 文件的能力。

从本质上来说学徒服务器是一个 Inventor 的子集，运行在使用它的应用程序的处理空间内。学徒服务器没有用户界面，与学徒服务器交互的唯一方式是通过它的 API。

学徒服务器提供对装配件结构、边界映像、几何特征以及文件属性的访问。大部分通过学徒服务器访问到的信息是只读的，文档属性和文件参考例外。

访问 Inventor 文档所包含的信息，这是许多应用程序常见的需求。在这

方面，学徒服务器比较好用，因为学徒服务器能够运行在与这个应用程序相同的处理空间内，比起在 Inventor 内部操作具有高得多的效率。由于它没有用户界面，能更快地处理很多的操作。学徒服务器包含在 Design Tracking 中，而 Design Tracking 可以从 Autodesk 网站免费下载。

1.2.5 其他

我们当然也可以用其他的支持 ActiveX 编程的语言如 Visual C++、Java、Delphi 等等来对 Inventor 编程，但是这不是本书谈论的内容，我们将不涉及这方面的内容。

1.3 Inventor VBA 编程基础

1.3.1 知识准备

自动化 API 是面向对象的，可以用对象的方法和属性来查询和编辑它们。如果有使用 Visual Basic 编程的经验，您就已经具备了面向对象的编程经验。

在一个对话框中放置一个控件时，此控件在你的程序中就代表一个对象。例如，放置一个编辑框，就创建了一个具有多种属性的对象，象文本、对齐、宽度和字型等。这些属性决定了这个对象的当前状态，我们可以控制和修改它们。

图 1-2 是简化的 Inventor API 对象继承关系图。

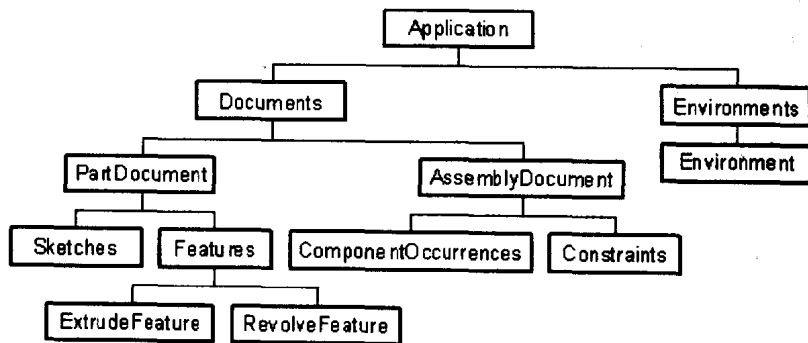


图 1-2 Inventor API 对象继承关系图

Inventor API 暴露 Inventor 产品的各种对象的内在功能，例如：可以获得 SketchArc 对象（草图中的一条弧线），它也具有象 Radius、StartAngle 和

SweepAngle 等属性。要想使用 Inventor API，首先需要熟悉那些被暴露出来的对象，并知道怎样获得想要的对象。可以从对象继承关系中了解 API 的结构。

这个继承关系图显示了各个不同对象之间的关联，并描述了怎样遍历它才能找到一个特定的对象。继承关系的关键是应用程序 (Application) 对象，它总位于对象继承的顶端。通过获取应用程序对象的访问权，就可以得到 API 中的任何其他的对象。

另一个重要的概念是集合对象。集合对象是一种特殊类型的对象，它是一些类似对象的一个集合，并提供遍历它们的方法。集合对象最少应该支持 Count 和 Item 属性。其中：

Count 属性返回此集合对象包含的对象个数；

Item 属性返回此集合中的一个特定对象。

除了 Count 和 Item 属性之外，集合对象也常常支持生成新对象的方法。例如：Documents 对象是一个集合对象，Documents 对象的 Count 属性返回当前打开的文档数目，可以用 Item 属性访问任何一个文档，Documents 对象也支持 Add 方法，可以使用它生成一个新文档，还支持一个 Open 方法，此方法可以打开一个已有的文件。

类型库 RxInventor.tlb 描述所有 Inventor API 中的对象。在 VB(A)的工程中引用这个文件，就可以访问所有的 API 对象了。学徒服务器有自己的一个专门的类型库文件，如果应用程序想使用学徒服务器就应该引用它。我们在“学徒服务器”一章中有详细的相关论述和实例程序。

如图 1-3 所示，可以在 VBA 或 VB 内用“对象浏览器(Object Browser)”查看类型库的内容。

这对于用其他语言编程的情况也是非常有用的。对象浏览器左上角的组合框提供所有已被引用的类型库的一个列表。

其中“类(Classes)”列表显示所有在类型库中定义的对象列表。从类列表选择一个对象后，浏览器的右边部分就显示此对象所支持的方法、属性和事件。选择一个方法、属性或事件，相关简短的描述就显示在浏览器的底部。

例如：在图 1-3 中，Inventor 类型库被选中，Documents 对象被选中，之后选中 Add 方法，浏览器底部就显示了此方法的每个参数的名字和类型：

参数 DocumentType，需要一个 DocumentTypeEnum 类型的枚举值。DocumentTypeEnum 枚举值包含各种 Inventor 文件类型常量，诸如零件、装配件、工程图等。点击 DocumentTypeEnum 会把你带到这个常量表，参见图 1-4。

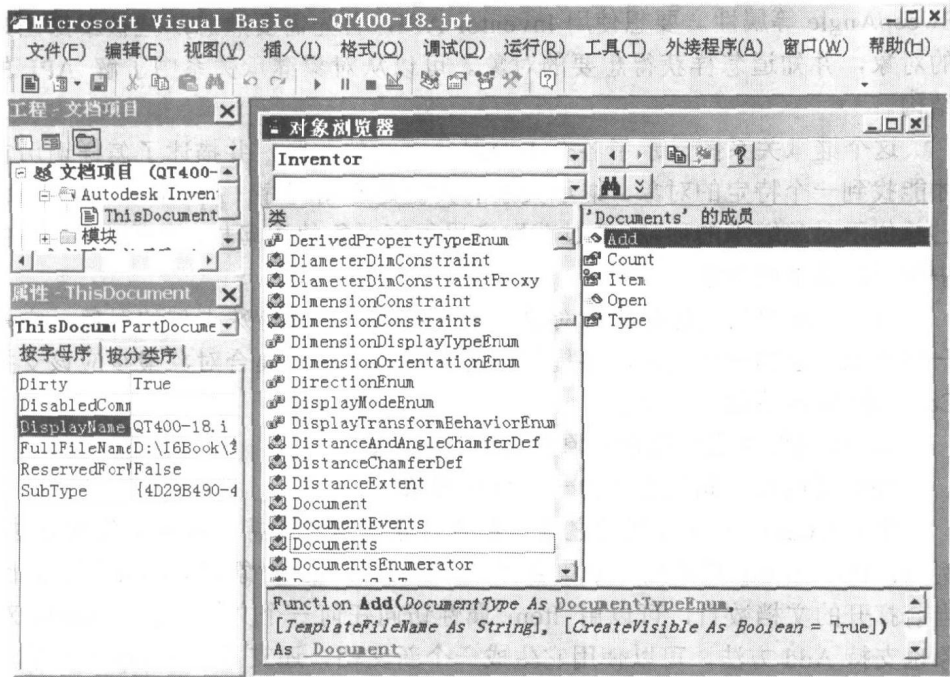


图 1-3 对象浏览器实例

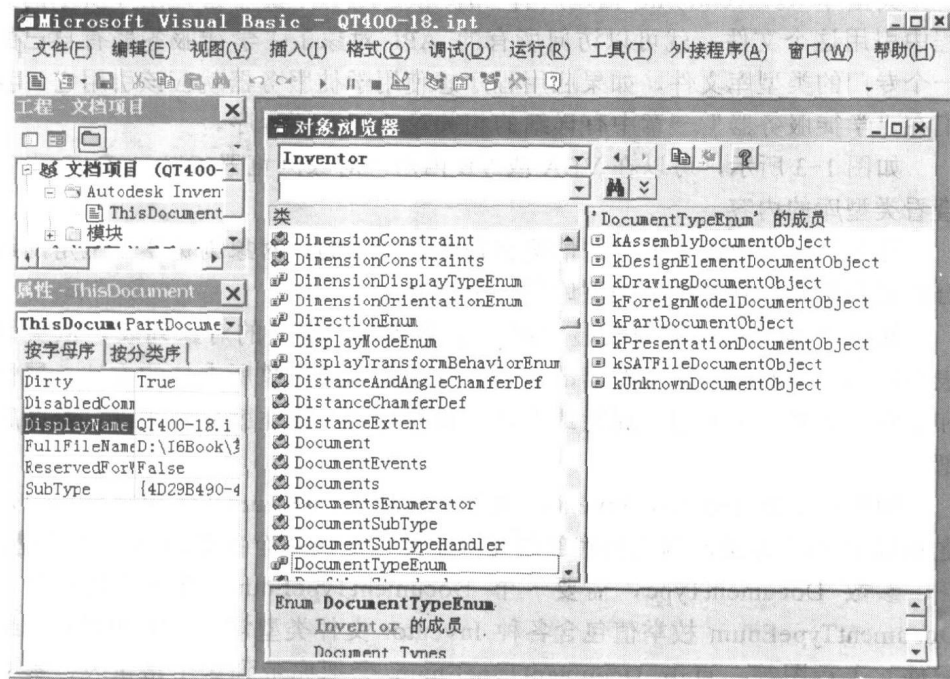


图 1-4 DocumentTypeEnum 可能的内容

第二个参数 `TemplateFileName`，需要一个字符串。

最后一个参数 `CreateVisible`，需要一个布尔值。

选定 `Add` 方法，按下 `F1` 键，就会出现 `Add` 方法的在线帮助，它包含这些参数的更详细的信息，并且链接到相关的示例程序。

对象浏览器和在线帮助是理解 `Inventor` 对象和对象继承的最主要和最有效的工具。

1.3.2 查看对象继承关系

无论用什么语言对 `Inventor` 编程，都可以用 `VBA` 调试器来更好地理解对象继承关系。下面的子例程就是做这件事情用的 (`T04.BAS`)。

```
Public Sub DebugTest()  
    Debug.Print ThisApplication.Caption  
    Stop  
End Sub
```

把鼠标指针移到子例程内的任何位置然后按 `F5` 运行，`Stop` 语句将象断点一样会使程序执行到该行然后暂停。把鼠标指针移到“`ThisApplication`”内的任何位置，然后在菜单中“**调试(D)**”→“**快速监视(Q)...**”，参见图 1-5。



图 1-5 监视操作实例

接着将弹出图 1-6 的对话框。

按下“**添加(A)**”按钮，`ThisApplication` 变量就显示在“监视窗口”中了。点击 `ThisApplication` 旁边的 `+` 符号，会在窗口中展开这个对象，并显示所有的属性值，参见图 1-7。

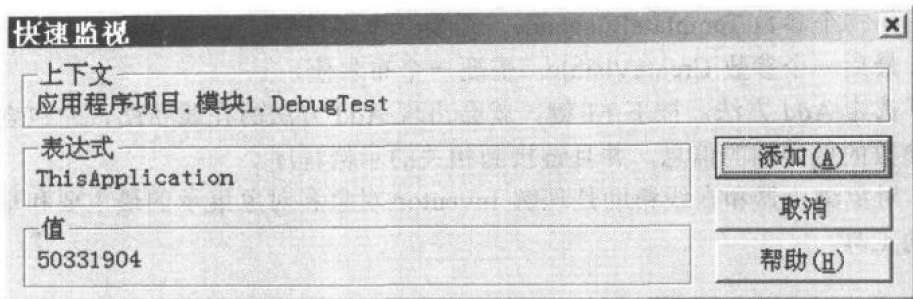


图 1-6 监视参数设置界面



图 1-7 对象继承关系展开实例

一些属性也会返回其他的对象，可以继续展开对象并遍历对象继承关系。

象这样使用 VBA 调试器就可以极大地帮助我们理解 Inventor API 的对象继承关系。