

# SolidWorks

计算机技术及工程应用丛书

# SolidWorks

## 二次开发 实例解析

江洪 魏峰 王涛威 等编著



 机械工业出版社  
CHINA MACHINE PRESS

计算机技术及工程应用丛书

# SolidWorks 二次开发 实例解析

江 洪 魏 峥 王涛威 等编著



机械工业出版社

SolidWorks 是一种非常优秀的三维机械设计软件, 在我国的用户越来越多。本书首次用 VBA/VB/VC/Delphi 四种语言讲述 SolidWorks 的二次开发内容, 在一定程度上解决了广大初学者入门难的问题。本书有若干循序渐进、深入浅出的范例, 读者可以照着做, 也可以学习其思路。通过本书的学习, 广大读者能学会 SW API 中的各种对象、方法、属性和事件的使用方法, 开发出适合自己特色的系统, 从而提高产品的质量和工作效率。

全书通俗易懂, 切合实际, 适合不同领域的人员阅读, 也可作为广大工程技术人员的自学用书和参考书。

## 图书在版编目 (CIP) 数据

SolidWorks 二次开发实例解析/江洪等编著. —北京: 机械工业出版社, 2004.2  
(计算机技术及工程应用丛书)

ISBN 7-111-13959-3

I. S... II. 江... III. 计算机辅助设计—应用软件, SolidWorks  
IV. TP391.72

中国版本图书馆 CIP 数据核字 (2004) 第 007948 号

机械工业出版社 (北京市百万庄大街 22 号 邮政编码 100037)

策 划: 胡毓坚

责任编辑: 蔡 岩

责任印制: 李 妍

北京机工印刷厂印刷 · 新华书店北京发行所发行

2004 年 3 月第 1 版 · 第 1 次印刷

787mm×1092mm 1/16·17.5 印张·431 千字

0 001—5 000 册

定价: 32.00 元 (含 1CD)

凡购本图书, 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

SolidWorks 是一种非常优秀的三维机械软件，其易学易用、全中文界面等特点使广大工程技术人员爱不释手。目前 SolidWorks 是市场分额增长最快、技术发展最快、市场前景最好、性能价格比最优的软件。在全球的销量已经达 20 万套，排名处于 3D CAD 软件销售榜首，遥遥领先于其他同类产品。

随着 SolidWorks 的普及，学 SolidWorks 的人越来越多，需要 SolidWorks 二次开发的单位、企业也越来越多，但有关 SolidWorks 二次开发方面的学习资料很少，广大初学者只能自己去摸索，在开发的过程中肯定会遇到种种困难，肯定会希望能得到一些指点与帮助。特别是 Visual C++ 6.0，入门很难，用 Visual C++ 6.0 开发 SolidWorks 就更难了。本书正是为了让广大初学者少走弯路，节省学习时间，提高学习效率而编写的。本书是国内第一本用 VBA/VB/VC/Delphi 四种语言讲述开发 SolidWorks 的过程，并拥有大量实例的书籍。

本书遵循教育规律和学习者的心理规律，用实例来讲述，且其相关的知识点都贯穿在实例中。从而具有生动活泼、活学活用、理论联系实际的特点。

由于 SolidWorks 的对象、方法、属性和事件的内容很多，如 ModelDoc 对象就有 477 种方法，所以本书不可能全部介绍完，开发者也不可能用到所有的函数。还有二次开发的编程思想和理论是在学习程序语言和实际应用中逐步建立的。本书可能解决不了实际问题，但是只要掌握了本书所述的知识，就能举一反三、融会贯通，根据需要，查帮助中相关的函数，解决自己的问题，开发出能满足企业要求的系统。

本书的程序测试曾用到李家鹏的减速器和清华大学出版的李启炎等编写的《SW 装配图和二维工程图》一书中的模型，本书曾在网上下载过几个免费程序和模型，如 <http://www.scottjbaugh.com/>、<http://www.nhcad.com>、<http://www.mikejwilson.com/>。在此对提供免费程序和模型的网站和 ScottBaugh、Joseph Jones 等人表示衷心感谢！感谢 ghz001 曾回答过作者的两个问题。

除封面署名外，参加本书编写的人员还有李明宇、陈锋、李铮、李仲兴、丁家翔。由于编者水平有限，书中难免有疏漏之处，恳请广大读者批评指正。

编者信箱为：99998888@126.com。

# 目 录

前言

第 1 章 SolidWorks API 的基本知识 .....	1
1.1 SolidWorks API 中的术语及开发工具 .....	2
1.1.1 OLE .....	2
1.1.2 ActiveX .....	2
1.1.3 COM .....	3
1.1.4 MFC .....	3
1.1.5 ATL .....	3
1.1.6 宏 .....	4
1.1.7 SolidWorks 插件 .....	4
1.1.8 开发工具 .....	8
1.2 程序指导 .....	8
1.2.1 OLE、COM 和 Dispatch .....	9
1.2.2 访问 SolidWorks 对象及属性 .....	9
1.2.3 布尔值 .....	10
1.2.4 整数 .....	11
1.2.5 质量属性 .....	14
1.2.6 返回值 .....	14
1.2.7 安全数组 .....	17
1.3 SolidWorks API 的语法 .....	18
1.3.1 变量 .....	18
1.3.2 返回值 .....	19
1.3.3 接口 .....	19
1.3.4 对象接口查询 .....	20
1.3.5 DLL 中导出函数的管理 .....	21
1.4 SolidWorks API 对象概述 .....	21
1.4.1 SldWorks 对象 .....	22
1.4.2 Environment 对象 .....	23
1.4.3 Frame 对象 .....	23
1.4.4 AttributeDef 对象 .....	23
1.4.5 Modeler 对象 .....	23
1.4.6 SWPropertySheet 对象 .....	23
1.4.7 ModelDoc 2 对象 .....	24
1.4.8 PartDoc 对象 .....	25

IV

1.4.9	DrawingDoc 对象 .....	25
1.4.10	装配体对象 .....	26
1.4.11	PropertyManagerPage 2 对象 .....	26
1.4.12	其他的对象 .....	26
<b>第 2 章</b>	<b>Visual Basic 6.0 开发 SolidWorks 的实例 .....</b>	<b>28</b>
2.1	Visual Basic 6.0 发展简史及特点 .....	28
2.2	用 Visual Basic 6.0 开发 SolidWorks 的一般步骤 .....	29
2.2.1	Visual Basic 6.0 开发 SolidWorks 的一般步骤 .....	29
2.2.2	执行应用程序的方式 .....	31
2.3	圆球 .....	35
2.3.1	激活\装载\新建文件 .....	35
2.3.2	圆和直线等 .....	38
2.3.3	修剪和旋转等 .....	39
2.3.4	程序的功能、运行结果、代码及说明 .....	43
2.4	添加自定义属性到文件 .....	46
2.4.1	自定义属性信息 .....	46
2.4.2	程序的功能、运行结果、代码及说明 .....	47
2.5	等距一系列面 .....	48
2.5.1	倒数第 N 个特征 .....	48
2.5.2	特征的边界范围 .....	49
2.5.3	插入等距面等 .....	49
2.5.4	程序的功能、运行结果、代码及说明 .....	50
2.6	修改零件序号 .....	51
2.6.1	SelectionManager 等 .....	51
2.6.2	零件序号等 .....	54
2.6.3	程序的功能、运行结果、代码及说明 .....	55
2.7	轴承标准件库 .....	57
2.7.1	分析结构特征, 建立轴承实体模型 .....	57
2.7.2	轴承库 .....	58
2.7.3	程序的功能、运行结果、代码及说明 .....	60
<b>第 3 章</b>	<b>用 Visual C++ 6.0 开发 SolidWorks 的一般步骤 .....</b>	<b>62</b>
3.1	Visual C++ 6.0 发展简史及特点 .....	62
3.2	早期工程向导 Swizard.swx .....	63
3.2.1	用 Swizard.swx 生成 DLL 的一般步骤 .....	64
3.2.2	分析 TryAddin 工程代码 .....	71
3.3	ATL(C/C++)对象向导 .....	76
3.3.1	用 ATL 生成 DLL 的一般步骤 .....	77
3.3.2	SwAtl 工程文件 .....	87
3.4	用户自定义向导和利用现有的实例 .....	88

3.4.1	生成用户自定义向导	88
3.4.2	分析现有实例的代码	89
<b>第 4 章</b>	<b>VC++开发 SolidWorks 的基础实例</b>	<b>95</b>
4.1	InitUserDLL3 和 SW 版本号	95
4.1.1	初始化 DLL	95
4.1.2	SolidWorks 的版本代码	96
4.1.3	消息框	98
4.1.4	程序的功能、运行结果、代码及说明	99
4.2	普通平键和圆锥	100
4.2.1	拉伸凸台	100
4.2.2	中心线和圆弧	101
4.2.3	清空草图和视区显示	102
4.2.4	程序的功能、运行结果、代码及说明	103
4.3	直齿圆锥齿轮	106
4.3.1	ModelDoc2::SelectByMark 方法	106
4.3.2	放样/旋转切除	107
4.3.3	样条曲线等	109
4.3.4	程序的功能、运行结果、代码及说明	111
4.4	选择点的坐标值和获得实体名	112
4.4.1	拾取点坐标和实体名	112
4.4.2	程序的功能、运行结果、代码及说明	113
4.5	有名称的实体	116
4.5.1	所命名实体的数目和数组等	116
4.5.2	程序的功能、运行结果、代码及说明	118
4.6	动态旋转模型视区	119
4.6.1	视区和设置单位等	119
4.6.2	程序的功能、运行结果、代码及说明	121
4.7	获得草图	123
4.7.1	特征等	123
4.7.2	程序的功能、运行结果、代码及说明	126
4.8	选择所有的草图线段	128
4.8.1	草图线段列表等	128
4.8.2	程序的功能、运行结果、代码及说明	130
4.9	隐藏 FeatureManager 里所选的特征	131
4.9.1	当前特征的用户接口状态	131
4.9.2	程序的功能、运行结果、代码及说明	132
4.10	打开所选的装配零部件文件和遍历装配体	134
4.10.1	装配对象等	134
4.10.2	程序的功能、运行结果、代码及说明	136

4.11	获得文件的历史版本信息 .....	140
4.11.1	版本历史数据的数组和存盘文件的版本字符串 .....	140
4.11.2	程序的功能、运行结果、代码及说明 .....	141
4.12	遍历所有打开的文件 .....	142
4.12.1	打开文件列表和获得 ModelDoc 对象 .....	142
4.12.2	程序的功能、运行结果、代码及说明 .....	143
<b>第 5 章</b>	<b>Visual C++ 6.0 开发 SolidWorks 的应用实例 .....</b>	<b>144</b>
5.1	菜单 .....	144
5.1.1	与菜单有关的几个方法 .....	144
5.1.2	示例程序中的菜单 .....	148
5.1.3	添加菜单项的实例 .....	150
5.1.4	样条曲线等 .....	152
5.1.5	程序的功能、运行结果、代码及说明 .....	157
5.2	工具栏 .....	159
5.2.1	与工具栏有关的几个方法 .....	159
5.2.2	示例程序中的工具栏 .....	162
5.2.3	添加工具栏的实例 .....	163
5.2.4	配合参数等 .....	164
5.2.5	程序的功能、运行结果、代码及说明 .....	175
5.3	交互对话框 .....	180
5.3.1	生成交互对话框的一般步骤 .....	180
5.3.2	程序的功能、运行结果、代码及说明 .....	184
5.4	数据库与工程图 .....	185
5.4.1	数据库 .....	185
5.4.2	与工程图标注有关的若干方法 .....	189
5.4.3	程序的功能、运行结果、代码及说明 .....	200
<b>第 6 章</b>	<b>拖放与预览零件 .....</b>	<b>204</b>
6.1	拖放零件 .....	204
6.2	预览零件 .....	207
6.2.1	在 SolidWorks 中预览 SolidWorks 文件的方法 .....	207
6.2.2	不需打开 SolidWorks 就能预览 SolidWorks 文件的方法 .....	211
6.2.3	无需安装 SolidWorks 即可预览 SolidWorks 文件的方法 .....	215
<b>第 7 章</b>	<b>用 VBA 开发 SolidWorks 的实例 .....</b>	<b>218</b>
7.1	VBA 概述及开发的一般步骤 .....	218
7.1.1	概述 .....	218
7.1.2	VBA 开发 SolidWorks 的一般步骤 .....	219
7.1.3	执行宏的方式 .....	221
7.2	存零件为其他格式 .....	224
7.2.1	路径名与保存文件 .....	224



7.2.2	程序的功能、运行结果、代码及说明 .....	226
7.3	批量修改零件的属性 .....	227
7.3.1	材料属性等 .....	227
7.3.2	程序的功能、运行结果、代码及说明 .....	230
7.4	用 Excel 宏开发 SolidWorks .....	231
7.4.1	用 Excel 宏开发 SolidWorks 的一般步骤 .....	231
7.4.2	程序的功能、运行结果、代码及说明 .....	233
7.5	用 Excel 控制装配体 .....	235
7.5.1	尺寸的系统值 .....	235
7.5.2	程序的功能、运行结果、代码及说明 .....	235
7.6	螺旋线 .....	237
7.6.1	模拟键盘操作 .....	237
7.6.2	程序的功能、运行结果、代码及说明 .....	238
7.7	动画的实现 .....	240
7.7.1	视区比例和当前图形视区 .....	240
7.7.2	IIF 函数 .....	240
7.7.3	程序的功能、运行结果、代码及说明 .....	242
<b>第 8 章</b>	<b>Delphi 7.0 开发 SolidWorks 的实例 .....</b>	<b>244</b>
8.1	Delphi 7.0 发展简史及特点 .....	244
8.2	用 Delphi 7.0 开发 SolidWorks 的一般步骤 .....	245
8.2.1	Delphi 7.0 开发 SolidWorks 的一般步骤 .....	246
8.2.2	执行应用程序的方式 .....	251
8.3	运动仿真和预览 SolidWorks 模型 .....	251
8.3.1	运动仿真 .....	251
8.3.2	预览 SolidWorks 模型 .....	253
8.4	Delphi 7.0 与数据库结合创建量具辅助设计系统 .....	254
8.4.1	圆\圆角\镜向 .....	254
8.4.2	Delphi 7.0 数据库 .....	257
8.4.3	量具辅助设计系统 .....	262
<b>参考文献</b>	.....	<b>270</b>

# 第 1 章 SolidWorks API 的基本知识

目前,在我国 CAD 市场上比较流行的三维 CAD 软件有 SolidWorks、MDT、UG、Pro/E、SolidEdge 等。但就价位来讲 SolidWorks、MDT、SolidEdge 是中低档价位的软件产品,而 UG、Pro/E、IDEAS 都属高中档价位的產品。

SolidWorks 是一套基于 Windows 的 CAD/CAE/CAM/PDM 桌面集成系统,是由美国 SolidWorks 公司在总结和继承了大型机械 CAD 软件的基础上,在 Windows 环境下实现的第一个机械三维 CAD 软件,于 1995 年 11 月研制成功。它全面采用非全约束的特征建模技术,其设计过程全相关性,可以在设计过程的任何阶段修改设计,同时牵动相关部分的改变。它既提供自底向上的装配方法,同时还提供自顶向下的装配方法。自顶向下的装配方法使工程师能够在装配环境中参考装配体其他零件的位置及尺寸设计新零件,更加符合工程习惯。它具有独创性的“封套”功能,可用来分块处理复杂的装配体。具有“产品配置”功能,为用户设计不同“构型”的产品。它集成了设计、分析、加工和数据管理的整个过程,所获得的分析和加工模拟结果成为产品模型的属性,在 SolidWorks 的特征管理器中清晰地列出了详细的数据信息。它还可以动态模拟装配过程,进行静态干涉检查、计算质量特征,如质心、惯性矩等。它将 2D 绘图与 3D 造型技术融为一体,能自动地生成零部件尺寸、材料明细表(BOM)、具有指引线的零部件编号等技术资料,从而简化了工程图样的生成过程。同时有中英文两种界面可供选择,其先进的特征树结构使操作更加简便和直观。具有较好的开发性接口和功能扩充性。能轻松实现各种 CAD 软件之间的数据转换传送。

目前, SolidWorks 的全球用户超过 20 万,国外大多数制造公司招聘时都要求应聘人具有操作 SolidWorks 的能力。国际上很多一流名校都将 SolidWorks 定为大学本科学生的必修课目,如麻省理工学院、英国剑桥大学等院校。95%的西方企业要求员工会运用 SolidWorks 软件。SolidWorks 连续几年被美国一些杂志评为优秀的 CAD 软件。

目前 SolidWorks 是市场份额增长最快、技术发展最快、市场前景最好、性能价格比最优的软件。除了少数行业需要 UG、PRO/E、CATIA 等高档 CAD 软件外,中档的 SolidWorks 完全可以满足一般的应用,其装配分析基本满足一般企业的要求。

但是 SolidWorks 也并不是十全十美的,不可能完全满足特定企业的特殊要求。例如,由于该软件是外国人写的,不适合中国的国标(如标题栏、明细表、放大图的标注、剖视图的标注、装配图的序号、标题栏等),也没有适合我国国标的标准件库。因此,为了适合特定企业的特殊需求,形成企业自己的特色,使 SolidWorks 在我国的企业中有效地发挥作用,并使常用的或重复的任务自动化,提高效率,就必须对其进行本地化和专业化的二次开发工作。

为了方便用户进行二次开发, SolidWorks 提供了几百个 API(Application Program Interface, 应用程序接口)函数,这些 API 函数是 SolidWorks 的 OLE(Object Linking and Embedding, 对象的嵌入与链接)或 COM(Component Object Model, 组件对象模型)接口,用户可以使用 VB/VBA/VCDelphi 等高级语言对 SolidWorks 进行二次开发,建立适合用户需要的、专用的 SolidWorks 功能模块。

学习编程最好的方法之一就是阅读源代码。这是世界公认的，也是著名的权威专家，已逝的 Richard Stevens 博士平生一贯坚持的观点。本书是一本用四种语言叙述 SolidWorks 二次开发的内容和实践指南的书。因此，它不会用大量的无节制的篇幅写 API 的理论部分，除非这些理论直接与编程相关。一般的学习过程是：先消化吸收有关的内容，然后再编写自己的程序。遵照这一规律，本书提供了一个可以使读者能举一反三的尽可能利用 SolidWorks API 函数的方法。本书用大量的实例，讲述了用 VBA、VB、VC、Delphi 开发 SolidWorks 的一般步骤、方法及语法等，使读者用少量的时间就能快速入门，掌握 SolidWorks 二次开发的方法，并应用到实际工作中去。

## 1.1 SolidWorks API 中的术语及开发工具

### 1.1.1 OLE

OLE 就是对象的嵌入与链接，它使应用程序之间能够通过数据嵌入或链接的方式来共享数据。含有嵌入或链接文档数据的应用程序被称为容器程序，提供文档数据编辑的应用程序则被称为服务器程序。

在 20 世纪 80 年代晚期和 90 年代早期，微软公司遇到了用单个大模块模式开发受到限制的困难。对于他们来说，这个系统是微软的 Office。他们想使一种类型的文档，如 Microsoft Excel 电子表格，嵌入到第二种类型的文档中，如 Microsoft Word 文档。但是，他们不满足于仅在字处理文档中显示电子表格。他们还想从字处理文档内部提供电子表格的所有功能，好像电子表格是被创建到字处理器中的一样。针对这个问题，在 1999 年，微软开发并公开发布了一种叫做对象链接和嵌入（OLE）的技术。对象链接和嵌入这项技术的创建使一种类型的文档可以被链接到另一种类型的文档中。OLE 1.0 使用的基本技术是动态数据交换（DDE），虽然它能够工作，但它非常复杂。

针对 OLE 1.0 的种种问题，1993 年，微软发布了 OLE 2.0 的规范。OLE 2.0 获得了巨大的成功，它的意义已经远远超出了复合文档的范畴，事实上，OLE 2.0 已经成为基于组件对象的 Windows 编程的基础。另外微软的 MFC 2.0 使得 OLE 2.0 获得了更大的成功。

OLE 2.0 推出以后，OLE 已经不再是对象的链接和嵌入的缩略语。

不妨认为 ActiveX 是 OLE 3.0，事实上 ActiveX 是 OLE 在网络上的扩展，它使用了 OLE 技术并且使它超过了本地机的范围，进入了一般的企业网和 Internet。

### 1.1.2 ActiveX

ActiveX 是 Microsoft 新近提出的一种技术，它以 COM 为基础，其中包括了 OLE 技术以及应用于 Internet 的多种技术。它使得不同的进程（甚至是网络上的进程）之间可相互通信，并且朝着多媒体方向发展。ActiveX 是一种面向对象的组件系统，其中包括了组件软件的各个方面，有复合文档、自定义控制、OLE 自动化、交互应用程序脚本、数据传送及其他分布式软件的相互作用。这些组件对象为用户提供了各种不同的功能，使得不同软件供应商提供的组件可在二进制间进行相互连接和通信。对象通过接口实现组件之间及组件与系统之间的相互作用。在桌面系统进行软件交互时，除了系统提供的标准控制外，OLE 控件是一个

重要的组成部分。

ActiveX 是一种标准，利用这个标准可以使通过不同语言开发的软件构件在单机或网络环境中相互操作。

同时，ActiveX 也是以组件对象模型为基础的开放技术的集合，它代表了应用程序与 Internet 的一种集成策略。

### 1.1.3 COM

COM 定义了标准的构建组件的方法，是软件组件互相通信的一种方式。它是一种二进制的网络标准，允许任意两个组件互相通信，而不管它们是在什么计算机上运行（只要计算机是相连的），不管各计算机运行的是什么操作系统（只要该操作系统支持 COM），也不管该组件是用什么语言编写的。COM 还提供了位置透明性：当编写组件时，其他组件是进程内 DLL、本地 EXE 还是位于其他计算机上的组件都无所谓。

COM 不是一个特殊类型的应用程序，而是一个可以用来为任何类型的应用程序构建组件的普遍模型。例如，ActiveX 和 OLE 都使用 COM，但两者都不是 COM。

传统应用程序的组成部分是分立的文件、模块或类，这些组成部分经过编译并链接之后才形成应用程序。要想推出应用程序的新版本，就需要将这些组成部分重新编译，既费时又费力。有了组件的概念，就可以将改进后的新组件插入到应用程序中，并替换掉原有的旧组件，从而赋予应用程序新的活力。

把许多已经做好的组件放到一起来形成一个组件库，好比一个类库。当制作应用程序时，如果要用到不同的组件，只需从刚建好的组件库中调出所需要的组件，然后将它们插入到适当的位置，来获得所向往的功能即可。

### 1.1.4 MFC

MFC (Microsoft Foundation Class Library, Microsoft 基础类库) 是一个庞大的类库，是 C++ 类结构的扩展。使用 MFC 将显著简化 Windows 程序的开发。MFC 封装了大多数 Windows API 函数，使程序员在开发 Windows 程序时能充分发挥 C++ 语言的优越性。每当新的 Windows 版本推出时，MFC 都作相应的改变以使旧代码能在新操作系统上编译并运行。

使用 MFC 最大的优点是 MFC 为程序员做了所有繁杂艰苦的工作，它的类库包括了成千上万行正确、完善而且强大的 Windows 代码。其中许多的成员函数，如果让程序员编写的话会花费数周的时间。从这点上说，使用 MFC 将显著缩短项目的开发周期。

### 1.1.5 ATL

使用 MFC 虽不能进行自由线程处理，但却可以轻松地编写 ActiveX 控件，其方便性的代价是，可执行部分过于庞大。对于 ActiveX 控件来说，它适于小文件，特别是在 WWW 上应用时，这点非常重要。一个非常流行的 MFC 替代工具，是微软的 ATL (Active Template Library, 活动模板库)。ATL 提供了一套范围很广的 C++ 类模板库。它提供了对几个接口内在的支持，包括 IUnknown 和 IClassFactory。它也支持创建 ActiveX 控件。ATL 被 Visual C++ 中的 ATL 应用程序向导所支持，可以自动地创建 COM 组件。

ATL 是 Visual Studio 所包括的工具之一，提供了创建部件的最大灵活性。它支持所有 COM

线程模型，并允许创建部件，如 DLL、EXE 或者服务。与 MFC 不同，ATL 支持双重接口。自从 ATL 成为一个 C++ 模板库以来，所有代码均包含在部件中。ATL 使用模板与多继承来实现对象。如果不适应这些 C++ 特性，可能会发现使用 ATL 会很顺利。但是，其所提供的向导在为部件创建基本框架，并将对象类加入框架的过程中，提供了巨大的帮助。

## 1.1.6 宏

宏是一系列命令的集合，相当于 Dos 下的批处理文件，及 AutoCAD 中的脚本文件。可以录制使用 SolidWorks 用户界面执行的操作，然后使用 SolidWorks 宏重新执行这些操作。宏所包含的调用相当于使用用户界面执行操作时，对 API 的调用。通过记录宏和交互式的执行任务，可以在所需的代码上获得命令和语法上的飞跃。在写任何代码前，都是记录宏来用作程序的基础。即当向程序添加功能时，返回到 SolidWorks，记录添加的宏。然后剪贴和粘贴记录的宏到代码中，这样做，即使对最先进的程序也是有益的。

## 1.1.7 SolidWorks 插件

### 1. 插件的定义及装卸

SolidWorks 插件是直接同 SolidWorks 软件一起工作的程序，它能提供给用户一些额外的功能，例如 PhotoWorks（渲染）程序。SolidWorks 可以自动地装载和卸载插件程序。

从 SolidWorks 97Plus 版开始引进了插件作为第三方软件。插件管理器看起来和 Excel 插件管理器一样。

在 SolidWorks 菜单中，选择【工具】→【插件(D)...】将显示插件对话框，如图 1-1 所示。

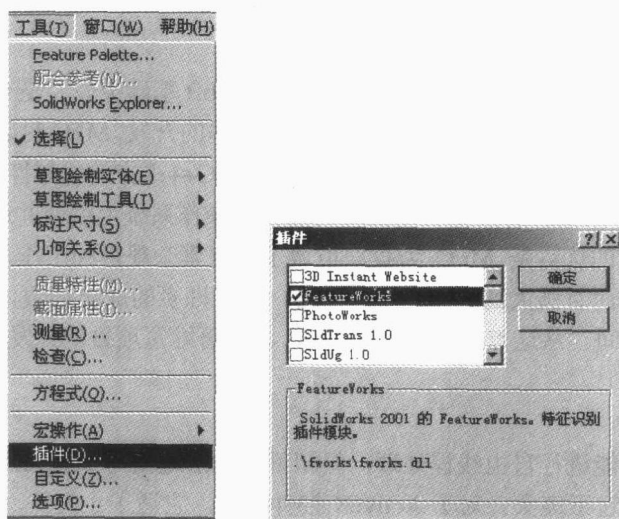


图 1-1 插件下拉菜单和对话框

对每个可用的插件程序，对话显示其插件名，插件名左边具有复选框。当用户高亮选中一个指定的插件名时，对话框的底部将显示一些提示文本，它简要地描述了插件。同时也显示了运行插件时的 DLL 文件名和路径。

如果用户选择或清除复选框之一，则表示 SolidWorks 将装载或卸载所选中的插件。在 SolidWorks 运行时，用户可以随时选择或清除任何一个插件。一次可以装卸多个插件，并在 SolidWorks 运行时保持设置。

当用户通过单击【确定】按钮退出对话框时，插件便开始根据复选框的状态起作用，所选的插件名将出现在菜单里，如图 1-2 所示。SolidWorks 在运行时保持所选的插件状态，SolidWorks 结束时运行的任何插件将自动在下一个 SolidWorks 开始时自动开始运行。

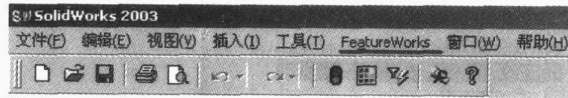


图 1-2 所选插件在菜单中的位置

为了通过【插件】对话框由 SolidWorks 装载插件，程序必须被作为一个 Microsoft Windows 的 DLL (Dynamic Link Library, 动态连接库) 文件而运行。

注意: 新建的 DLL 不能起中文名，否则不能通过编译。

## 2. 注册 SolidWorks 插件

Windows 注册是配置信息的集中贮存。可以像文件系统一样来组织它。它由键体系结构组成 (类似于文件夹)。每个键有一个名称，可以包含其他的键和值 (类似于文件)。每个值有一个名称和数据。

SolidWorks 插件管理器从一系列注册项目中获得所有的可用插件的信息。

插件使用的可用的键名可以在【HKEY\_LOCAL\_MACHINE\SOFTWARE\SolidWorks\Applications】下立即找到，如图 1-3 所示。

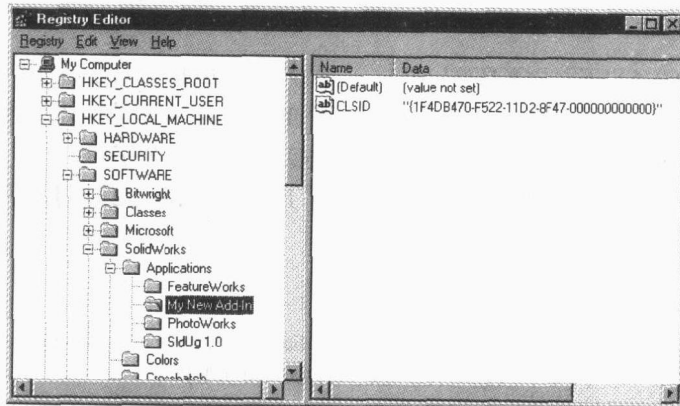


图 1-3 插件注册名的位置

每个键都有一个被称为 CLSID 的值。该值包含了被称为 GUID (Globally Unique Identifier, 全局唯一标识) 的 32 位数。GUID 使用由 OSF (Open Systems Foundation, 开放式系统基金会) 的 DCE (Distributed Computing Environment, 分布式计算环境) 创建的算法生成。GUID 是按 16 字节 (128 位) 结构存储的，这样就可以有  $2^{128}$  个可能的 GUID，不必担心会用完所有的 GUID。用作生成 UUID 的算法使用 48 位网络适配器的 ID，当前日

期是从 1582 年 10 月 15 日子夜 12 时开始的在最近 100ns（纳秒）间隔内的时间。这种方法使两个 GUID 是同一个值的可能性非常小。

这些相同的 GUIDs 也出现在注册表中以【HKEY\_CLASSES\_ROOT\CLSID】为键名的别处，如图 1-4 所示。

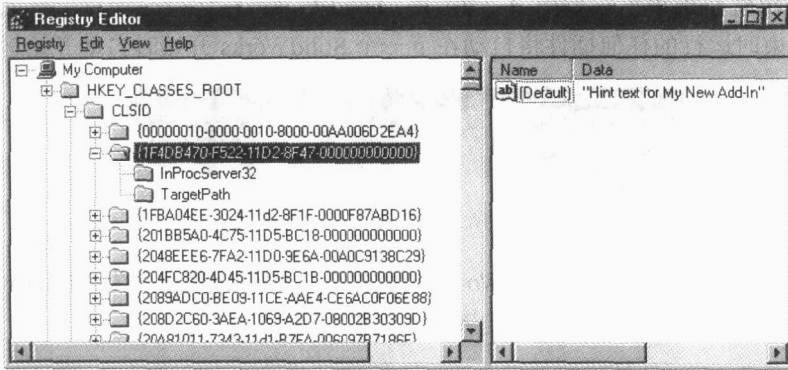


图 1-4 注册表中 CLSID 的另一处相同的 GUIDs

插件管理器在这些以 GUIDs 命名的键中可找到可用插件的额外信息：任何插件的提示文本都存储在以 GUID 命名的键的默认值中，运行插件时的 DLL 全路径和文件名存储在以 InProcServer32 命名的子键的默认值中。

为了使用 SolidWorks 插件管理器，有两个区域要求编址：

(1) 注册要求

1) 在注册表中的 HKEY\_LOCAL\_MACHINE/software/SolidWorks/Applications 目录下，为应用程序添加子键。该子键应该具有名称为“CLSID”的值和包含应用程序的唯一的 CLSID 字符串。

2) 在 HKEY\_CLASSES\_ROOT/CLSID 下，应用程序应该有与步骤 1) 中的 CLSID 匹配的唯一的 CLSID。CLSID 子键项应该具有无名称的值，该名称是描述应用程序的。该描述将被用于插件管理器对话框，一定不能超过行文本或大约 132 个字符。

3) HKEY\_CLASSES\_ROOT/CLSID 项应该有 InProcServer 32 子键，它应该包含完整的路径和主 DLL 名称。

下面是从一个\*.reg 文件中取出的示例：

```
1 [HKEY_LOCAL_MACHINE\SOFTWARE\SolidWorks\Applications\Sample Add-In]
2 "CLSID"="{42CF4A11-CC8F-11d0-9EC3-00609728F9FE}"
3 [HKEY_CLASSES_ROOT\CLSID\{42CF4A11-CC8F-11d0-9EC3-00609728F9FE}]
4 @="Sample C++ Add-In for SolidWorks. Illustrates general format required of a SolidWorks Add-In."
5 [HKEY_CLASSES_ROOT\CLSID\{42CF4A11-CC8F-11d0-9EC3-00609728F9FE}\InProc Server 32]
6 @="c:\sldworks\samples\comuserdll\i386\windebug\userdll.dll"
```

这个\*.reg 文件提供了 ComUserdll 应用程序示范。用 Notepad 即可编辑\*.reg。\*.reg 文件可以导入注册编辑器或从文件管理器中用鼠标双击也可以执行。

注意：不必用 CLSID 字符串提供 SolidWorks 示范程序或如上所示的内容。应用程序可

以在向导模板中通过选项产生一个惟一的 CLSID 字符串。

如果你正用 regedit 编辑注册，那么如上所述的注册文件中的“@”被注册编辑器解释为“默认”项目。因此应该生成子键，输入值到默认项中。

#### (2) 必要的应用程序条件

应用程序应该有自己的安装目录。不应该对任何文件再使用 APPS 子目录。因为 DLL 可能会仍然驻留在 APPS 目录，运行 SolidWorks 插件管理器时，SolidWorks 可能会加载 DLL 两次。这对应用程序的结构可能会引起问题。

应用程序必须适当地卸载。这包括释放所有的对象指针，删除所有的事件和所有的菜单。除了删除菜单外，应用程序应该已经考虑到适当地卸载了。这是关键的一步，因为不卸载 DLL 后留下的任何对象或事件也许会引起内存泄漏或程序异常。

应用程序必须考虑插件也许会在 SolidWorks 运行过程中加载。例如，应用程序也许在整个 SolidWorks 运行期间被加载和运行。可以在运行期间追踪被打开、生成、保存和关闭的每个文件。如果这类信息对应用程序是重要的，那么程序代码应该为其说明。

### 3. SolidWorks 插件不出现在插件对话框里的原因

如果 DLL 不出现在【工具】→【插件】菜单的【插件】对话框里，有可能是下面的两个常见问题之一：

在 InProcServer 32 指定的位置找不到 DLL。

注册项之一不正确。这是非常敏感的区域。阅读上述的描述，确认每步操作正确。另外，通过【文件】→【打开】也可装载 DLL。

### 4. 用向导 (SwAddin) 生成 SolidWorks 插件时，插件和应用程序的作用

使用向导 (SwAddin) 来生成 SolidWorks 插件时，插件必须包含一些特殊的功能和代码。理解插件和应用程序的作用是必要的。

#### (1) 插件必须做什么

生成的 DLL 插件必须作为一个 COM 服务器。插件必须：

1) 运行支持向导 (SwAddin) 的共同生成的对象 (co-creatable object)。

2) 从安装处调用 regsvr32。在 HKEY\_LOCAL\_MACHINE\SOFTWARE\SolidWorks\AddIns 处注册插件的 CLSID，设置下列注册键：

默认：1 或 0，1 可使插件在插件管理器中显示，当用户启动 SolidWorks 时，即装载插件。

描述：在插件管理器中显示的插件说明文本。

标题：在插件管理器中显示的插件标题。

3) 作为必需的默认事件处理程序。

运行 SwAddin::ConnectToSW 时，插件可以：

1) 调用 SldWorks::SetAddinCallbackInfo，传递插件的实例句柄和对象，该对象支持回调方法。SolidWorks 保存该对象，用于回调。

2) 调用 SldWorks::AddMenuItem2，传递与菜单项有关的回调。

3) 调用 SldWorks::AddToolbar3，传递与工具按钮有关的回调方法。

#### (2) SolidWorks 应用程序做什么

当终端用户启动 SolidWorks 应用程序时，SolidWorks 应用程序应该做以下的事：



- 1) 检查注册的插件。
- 2) 生成与插件的 CLSID 有关的对象。
- 3) 运行插件上的 QueryInterface 来寻找 SwAddin 对象。
- 4) 调用 SwAddin::ConnectToSW, 传递 SolidWorks 和插件 ID 的指针。

当用户关闭 SolidWorks 或插件管理器中的插件时, SolidWorks 将:

- 1) 调用 SwAddin::DisconnectFromSW, 准备清除插件。
- 2) 清除用插件 CLSID 生成的对象。

如果终端用户取消了插件管理器中的插件, 在下次重新启动 SolidWorks 时, SolidWorks 将不会重新装载插件。如果终端用户在插件可用时, 关闭 SolidWorks, 那么在下次重新启动 SolidWorks 时, SolidWorks 会重新装载插件。

当终端用户把插件 DLL 拖进 SolidWorks 时或用【文件】→【打开】打开一个新插件时, SolidWorks 将运行 regsvr 32, 自动地注册插件。

### 1.1.8 开发工具

任何支持 OLE 和 COM 的编程语言都可以作为 SolidWorks 的开发工具。SolidWorks 二次开发分两种, 一种是基于自动化技术的, 另一种开发方式是基于 COM 的。COM 技术可以使用最多的 SolidWorks API, 并可控制 SolidWorks 运行的方式, 生成\*.dll 格式的文件, 也就是 SolidWorks 的插件。

最简单的二次开发工具是 VBA, 在涉及 Excel 时用 VBA 最为方便。VBA 常用于录制宏, 得到基本程序框架。易学易用的高级开发语言是 VB, 但功能有限。功能强大的语言是 VC, 但是入门难。Delphi 集中了 VB 和 VC 两者的优点, 学习过程中很容易上手, 而且功能也非常强大, 尤其在界面设计、数据库编程和网络编程方面更具有独到之处。遗憾的是 SolidWorks 常常不支持 Delphi+OLE。

Visual C++ (通常是 DLL 格式) 并不一定比 Visual Basic (通常是 EXE 格式) 快, 这取决于瓶颈在那里。SolidWorks 中的一些操作是非常昂贵的, 如重建装配体。此时, 瓶颈是 SolidWorks, 模块内的 DLL 并不比模块外的 EXE 快。相比之下, 其他的操作相当便宜, 如获得一个面的颜色。若想获得一个零件每个面的颜色, 就要从 SolidWorks 多次获得单个面信息, 此时, 跨越边界的费用也许是明显的, 模块内的 DLL 将比模块外的 EXE 快许多。

总之, 各种软件都有各自的特点与不足。开发者可以根据自身的条件、工具的特点, 选择一种合适的开发工具。当然也可以同时选用二三种工具, 但这对开发者的要求极高。本书将介绍常用的 VBA、VB、VC、Delphi 这四种语言开发 SolidWorks API 的一般步骤和实例。

## 1.2 程序指导

本节将介绍与 API 编程有关的知识, 如访问 SolidWorks 对象或属性、事件、整数、布尔值、接口指针、质量属性、对象返回值、变量返回值、SafeArray 及其返回值、返回单位、注意事项等内容。