

附赠1CD
含本书源代码



AutoCAD VBA&VB.NET 开发基础与实例教程

(第二版)

曾洪飞 卢择临 张帆 编著

C#版



中国电力出版社
CHINA ELECTRIC POWER PRESS

TP391.72
1195-2

013744067

AutoCAD VBA&VB.NET 开发基础与实例教程 (第二版)

曾洪飞 卢择临 张帆 编著



P
TP391.72
1195-2



中国电力出版社
CHINA ELECTRIC POWER PRESS

内 容 提 要

本书以 Visual Studio 2010 (编程语言为 C#) 和 AutoCAD 2008 中文版 (部分实例采用了 AutoCAD 2012) 为开发平台, 通过 90 个精彩的编程实例, 向读者展示了 AutoCAD .NET 二次开发的高级技术。

全书共 19 章, 内容涵盖基础开发环境配置、创建和编辑基本图形对象、AutoCAD 内置用户交互手段、图形数据库块表和符号表、几何类辅助计算、扩展 AutoCAD 用户界面、三维实体的创建编辑、读/写数据库和外部文件、规则重定义和参数化绘图约束、布局打印、程序发布, 以及与 VC++ 的混合编程等方面, 从不同的角度分析 AutoCAD .NET 编程的技术和技巧。

本书层次清晰、实例丰富, 具有较强指导性和实用性, 适用于土木、建筑、机械、电子等多个领域的工程设计人员和软件开发人员, 也可供对计算机辅助设计感兴趣的各类高等院校学生, 以及各工程单位相关技术人员等使用。

图书在版编目 (CIP) 数据

AutoCAD VBA&VB.NET开发基础与实例教程 / 曾洪飞, 卢择临, 张帆编著. —2版. 北京: 中国电力出版社, 2013.3

ISBN 978-7-5123-4103-6

I. ①A… II. ①曾… ②卢… ③张… III. ①AutoCAD软件-教材
②BASIC语言-程序设计-教材 IV. ①TP391.72②TP312

中国版本图书馆CIP数据核字 (2013) 第037597号

中国电力出版社出版、发行

(北京市东城区北京站西街 19 号 100005 <http://www.cepp.sgcc.com.cn>)

北京丰源印刷厂印刷

各地新华书店经售

*

2008 年 2 月第一版

2013 年 5 月第二版 2013 年 5 月北京第三次印刷

787 毫米 × 1092 毫米 16 开本 39.5 印张 925 千字

印数 5001—8000 册 定价 88.00 元 (含 1CD)

敬告读者

本书封底贴有防伪标签, 刮开涂层可查询真伪
本书如有印装质量问题, 我社发行部负责退换

版 权 专 有 翻 印 必 究



前 言

再版说明

本书第一版于 2008 年 2 月出版后,得到了广大 AutoCAD 二次开发人员的积极评价。但由于第一版使用的编程语言是 VBA 和 VB.NET,而大部分 .NET 开发人员使用的编程语言是 C#,因此在第一版的基础上使用 C#重新编写第二版。

读者对象

本书介绍利用 C#编程语言对 AutoCAD 进行二次开发,对这方面感兴趣的工程技术人员、研究人员和程序开发人员都可以阅读本书,并从中受益。

从技术的层次来说,本书适合初学者和中级用户,以在国内几个著名的 CAD 论坛经常讨论的问题为主线,希望能够引导读者进入 AutoCAD 二次开发的大门,成为合格的专业开发人员。

本书内容

从内容上来看,本书可以分为 19 章。

第 1 章是学习 .NET 开发 AutoCAD 的基础。首先介绍了使用 Visual Studio 2010 创建 AutoCAD .NET 程序的基本步骤及环境搭建,然后讨论了如何进行程序的优化工作、调试及异常处理,最后讲解了使用增强型向导 AutoCAD .NET Addin Wizard 来简化 AutoCAD .NET 程序开发环境的搭建。

第 2 章详细地分析了在 .NET 中创建图形对象的基本原理,介绍了直线、圆、圆弧、多段线、椭圆、样条曲线、文字、图案填充、面域、尺寸标注、(多重)引线、形位公差等多种图形对象的创建,并以自定义函数的形式对这些图形对象的构造函数进行了封装,在实用性方面进行了一定的扩展。

第 3 章的主题是 AutoCAD .NET 二次开发应用程序和用户的交互,包括获取用户输入、构建选择集、自定义对象捕捉、鼠标停留提示及访问 AutoCAD 内部命令等内容,特别是引入 LINQ 功能使选择集的构建大大的得到简化。

第 4 章内容是符号表。通过多个实例介绍了层表、文字样式表、标注样式表、视图表、视口表、UCS 表和线型表的新建、修改、遍历、导出等多种常用的处理方法。

第 5 章介绍了块和表格。本章开头介绍创建、插入与访问简单块、属性块、可缩放块、动态块的方法,然后介绍使用属性块制作自动编号系统,最后介绍了从块派生的表格。

第 6 章讨论扩展数据、扩展字典和对象字典。扩展数据和扩展字典都是向实体附加数据的一种手段，扩展数据能够向实体附加的数据量较小，扩展字典是推荐的新方法；对象字典则是向图形本身追加数据的一种手段，本章介绍了 AutoCAD 中常用的对象字典：组及多线样式字典。

第 7 章介绍了文档与 AutoCAD 图形数据库。其中文档部分涉及的内容包括文档的管理（新建、打开、保存、关闭及锁定）、图形摘要信息、外部参照和光栅图像，图形数据库部分则主要介绍如何创建和访问图形数据库以及在数据库之间传递数据。

第 8 章首先讨论常用几何类的基本应用，然后在此基础上以绘制三维管道的实例来介绍矩阵的主要作用，最后通过判断点和多段线的位置关系介绍计算实体交点及射线法的知识。

第 9 章介绍 AutoCAD 中的各类事件，应用程序事件以禁用命令、按照实体类型分层和系统变量变化时通知用户为例，文档事件以阻止文档的关闭为例，对象事件以对象位置的关联为例。

第 10 章讲解用户界面的实现，包括自定义对话框、AutoCAD 对话框、标签式对话框、面板、菜单和工具栏、进度条、状态栏。

第 11 章首先介绍使用 LINQ 技术访问 SQL Server 及 Access 数据库的方法，然后介绍如何读/写外部文件（包括文本文件、二进制文件和 XML 文件），最后介绍如何与 Excel 进行交互。

第 12 章介绍了 AutoCAD 中的三维实体。本章首先介绍标准形状的三维实体的创建，然后在此基础上通过拉伸、旋转、扫掠、放样与布尔操作构建高级的三维实体，最后讨论如何对创建的三维实体进行修改。

第 13 章介绍了实现 Jig（拖拽效果）的基本原理。动态创建圆、等边三角形及其内切圆的实例展示了 EntityJig 类的使用技巧以及点、距离和角度的拖拽输入方法；动态创建椭圆及模拟镜像命令的实例揭示了 DrawJig 类的运用方法及几何算法分析。

第 14 章介绍程序的自动加载与发布。首先介绍如何通过修改注册表的方式让 AutoCAD 自动加载.NET 程序，然后介绍使用 Inno Setup 为.NET 程序创建专业的安装包。

第 15 章的内容是混合编程。首先介绍使用 P/Invoke 技术在.NET 程序中调用未封装的 C++函数，然后通过调用 COM 对象来访问 VBA 提供的函数及属性，最后介绍如何在.NET 程序中调用 ObjectARX 生成的 arx 文件、dll 文件及自定义实体。

第 16 章讲解高级用户界面的实现，包括 OPM（属性面板）、WPF 界面及 Ribbon 选项卡。

第 17 章的内容是讨论 AutoCAD 2010 中新出现的规则重定义（Overrule）。通过规则重定义，可以改变实体的外观和行为。本章实例主要介绍了显示、亮显、变形及对象重定义。

第 18 章介绍了参数化绘图与约束。AutoCAD 2010 引入了参数化绘图的概念，参数化绘图是一项用于具有约束设计的技术。AutoCAD 中的约束分为几何约束和标注约束，本章实例将说明如何通过编程的方式实现这两种约束。

第 19 章讨论打印和布局的问题。首先介绍布局和工作空间的关系，然后通过一个模拟 AutoCAD “页面设置对话框”的实例讲解打印的设置工作，最后介绍打印单页与多页文档的具体方法。

配套光盘

本书的配套光盘中给出了本书所有源程序代码，以及在编程中可能用到的资源文件。使用配套光盘，你可以直接将代码复制到计算机中学习，而不必输入所有的代码。但是我们还是建议自己手工输入代码，将程序运行结果与光盘中程序执行结果相对照，不要害怕自己输入的代码运行时出现问题，在解决这些问题的过程中积累的经验会大大提升你的水平。

本书使用的编程环境

本书使用如下的编程环境。

- 操作系统：Windows XP。
- AutoCAD 版本：主要是 AutoCAD 2008，部分新功能（第 10 章第 9 节 WPF 界面、第 10 节 Ribbon 选项卡、第 14 章规则重定义）使用 AutoCAD 2012。
- Visual Studio 版本：Visual Studio 2010 SP1。
- 开发语言：C#。

对于 AutoCAD 2007 及其以下版本，如果不使用 AutoCAD 2008 及以上版本新增加的功能，仍然可以在这些低版本的 AutoCAD 中运行本书的程序。但本书的程序未在低版本的软件上经过完整测试，本书作者不保证所有的程序都能在 AutoCAD 2007 及其以下版本中使用。

关于ObjectARX编程站

ObjectARX 编程站 (<http://www.objectarx.net>) 以托管 ObjectARX 编程为主要发展方向，是国内该方面技术水平最高、讨论最活跃的论坛，站长才鸟欢迎每一位对托管 ObjectARX 编程有兴趣的朋友加入。

本书的技术支持

读者可以通过以下方式获得关于书中问题的技术支持：

- 在 ObjectARX 编程站的论坛 (<http://www.objectarx.net/forum.php>) 的相关版块发帖。
- 给作者发送电子邮件，曾洪飞 (zhf7878@126.com)、卢择临 (ahlzl@126.com) 和张帆 (zf9568@263.net)。

关于本书作者

本书是集体劳动的结晶！下面是每一位作者所撰写的内容：

- 曾洪飞（才鸟）（“ObjectARX 编程站”站长，.NET 和托管 ObjectARX 开发方面的专家）：第 1~7 章、9~11 章、14 章、15 章第 1~2 节、16~18 章、19 章第 2~4 节。
- 卢择临（ahlzl）（安徽省淮北工业学校教师）：第 12 章、13 章和 15 章第 3、4 节。
- 张帆（zfbj）（北京智帆高科科技有限公司创始人）：第 8 章、15 章第 5 节、19 章第 1 节。

本书的勘误和更新

尽管在写书和校对过程中我们花费了巨大的精力，但是还不能保证书中没有任何的错误，更不能保证某些技术随着时间的推移还是最好的解决方案，我们对书中可能存在的缺陷表示歉意，同时欢迎每一位读者提出宝贵的意见和建议。您可以通过论坛或者电子邮件表达意见和建议。

致谢

本书的再版经历了很多的环节，作者感谢购买本书第一版的朋友们，正是在你们的鼓励和支持下，我们才有机会和信心使用 C#来编写本书。

曾洪飞首先要感谢我的母校同济大学，七年的大学生涯是我人生中最为重要的阶段；然后我要感谢我的家人，尤其是我的妻子加满，在精神上、生活上无微不至的关心和照顾；最后我要感谢广大关心“ObjectARX 编程站”的朋友们，本书的很多内容就来自于你们在论坛上发表的帖子，希望你们能一如既往地支持本站的发展。

卢择临在此要感谢两位合作者对我的信任，感谢家人对我的关心，是你们的支持才使我完成了这次编写工作。在这里，我还要感谢国外的 AutoCAD 二次开发高手丹尼尔，我技术水平的提高与你多年来对我的帮助是密不可分的。愿上帝永远保佑你！

张帆要感谢曾洪飞在统稿过程中帮我做的格式调整和代码整理工作，以及卢择临对我撰写程序的细致验证。由于工作比较忙导致自己撰写内容一再缩水（顺表歉意），所幸曾洪飞和卢择临两位作者在本书中奉献了近三年来在.NET 开发方面积累的所有技能，确保本书包含足够丰富的实例和技巧！

最后要感谢的是本书的读者，感谢您选购这本图书，这是对我们辛苦工作最好的认可，希望这本书能对你有所帮助，祝读书愉快！

编著者

目 录

前言

第 1 章 .NET 开发 AutoCAD 编程基础	1
1.1 第一个 AutoCAD 程序	1
1.2 程序的优化、调试及异常处理	7
1.3 用向导创建 AutoCAD 程序	16
第 2 章 创建和编辑基本图形对象	23
2.1 直线	23
2.2 编辑图形对象	34
2.3 圆和圆弧	45
2.4 多段线	49
2.5 椭圆和样条曲线	54
2.6 文字	57
2.7 填充	64
2.8 面域	72
2.9 尺寸标注	77
2.10 引线与形位公差	84
第 3 章 用户交互	94
3.1 获取用户的输入信息	94
3.2 选择集	101
3.3 选择集的简单化: LINQ	114
3.4 自定义对象捕捉	120
3.5 鼠标停留提示	124
3.6 访问 AutoCAD 内部命令	128
第 4 章 符号表	135
4.1 操作图层	135
4.2 文字样式	142

4.3	标注样式	148
4.4	视图	153
4.5	视口	157
4.6	UCS	162
4.7	线型	168
第 5 章	块和表格	177
5.1	简单块	177
5.2	属性块	182
5.3	动态块	189
5.4	可缩放块	194
5.5	导入外部文件中的块并输出预览图形	200
5.6	利用块制作自动编号系统	204
5.7	表格	210
第 6 章	扩展数据、扩展记录和对象字典	218
6.1	扩展数据	219
6.2	使用扩展字典和有名对象字典创建明细表	227
6.3	组字典	234
6.4	多线与多线样式字典	240
第 7 章	文档和图形数据库	245
7.1	创建和访问图形数据库	245
7.2	在图形数据库之间传递数据	252
7.3	图形摘要信息	258
7.4	文档管理	263
7.5	外部参照与光栅图像	269
第 8 章	几何类和几何计算	278
8.1	常用几何类	278
8.2	坐标系变换	289
8.3	判断点是否在多段线内部	294
第 9 章	事件	302
9.1	应用程序及文档事件	302
9.2	对象事件	308

第 10 章	用户界面	313
10.1	自定义对话框.....	313
10.2	AutoCAD 内置对话框.....	323
10.3	标签式对话框.....	333
10.4	面板.....	342
10.5	菜单和工具栏.....	353
10.6	进度条.....	370
10.7	状态栏.....	375
第 11 章	外部数据库与外部文件	380
11.1	创建基于 SQL Server 数据库的应用程序.....	380
11.2	创建基于 Access 数据库的应用程序.....	394
11.3	访问外部文件.....	407
11.4	与 Excel 交互.....	414
第 12 章	三维实体	419
12.1	创建标准形状的三维实体.....	419
12.2	三维实体的拉伸、旋转及布尔操作.....	423
12.3	扫掠与放样.....	430
12.4	三维实体的边与面.....	434
12.5	三维实体的倒角与圆角.....	442
12.6	Brep:边界表示.....	449
第 13 章	Jig:拖拽效果的实现	455
13.1	使用 EntityJig 类以拖动圆心的方式创建圆.....	455
13.2	使用 EntityJig 类创建等边三角形及其内切圆.....	461
13.3	使用 DrawJig 类创建有拖拽效果的椭圆.....	466
13.4	使用 DrawJig 类模拟镜像命令.....	471
第 14 章	程序的自动加载与发布	476
14.1	注册表自动加载.NET 程序集.....	476
14.2	创建安装程序.....	484
第 15 章	混合编程入门	490
15.1	使用 P/Invoke 技术调用未封装的 C++函数.....	490
15.2	.NET 与 COM.....	499

15.3	调用 ObjectARX 生成的 arx 文件	503
15.4	调用 ObjectARX 生成的 dll 文件	507
15.5	自定义实体	510
第 16 章	高级用户界面	527
16.1	.NET 版本的 OPM (属性面板)	527
16.2	WPF 界面	533
16.3	Ribbon 选项卡	545
第 17 章	规则重定义 (Override)	554
17.1	显示重定义: 改变实体的外观	554
17.2	重定义有扩展数据的实体	558
17.3	自定义过滤: 重定义符合条件的实体	563
17.4	亮显重定义: 亮显改变实体的颜色	567
17.5	变形重定义: 让点沿曲线移动	570
17.6	对象重定义: 控制对象的复制与删除	573
第 18 章	参数化绘图与约束	578
18.1	几何约束	578
18.2	标注约束	582
第 19 章	布局与打印	587
19.1	布局	587
19.2	打印设置	593
19.3	打印单页文档	608
19.4	打印多页文档	615
参考文献	621

第 1 章

.NET开发AutoCAD编程基础

当 AutoCAD 被首次用 Visual Studio .NET 编译生成的时候(开始于 AutoCAD 2004), .NET 开发 AutoCAD 的大门就被打开了。实际上 Autodesk 公司在内部开发 AutoCAD 的时候, 不像以前一样完全使用非托管 C++, 部分开发工作使用了 .NET, 这样就可以利用 .NET 框架提供的许多优点如创建用户界面、访问数据等。

众所周知, 对 AutoCAD 进行二次开发用到的主要工具有 ObjectARX、VBA 和 Lisp, 但它们的优缺点是显而易见的: ObjectARX 功能强大, 编程效率高, 但它的缺点是编程者必须掌握 VC++, 而这门语言非常难学; VBA 和 Lisp 虽然简单易上手, 但它们对于开发大型的程序好像无能为力。而 .NET 则结合了 VC++ 功能强大与 VBA 易用的特点, 可以非常快速地开发出功能强大的 AutoCAD 程序。通过本书的学习, 就可以掌握如何使用 .NET 来进行 AutoCAD 的二次开发。

通过本章非常简单的实例学习, 你将大致了解 .NET 开发 AutoCAD 的基本流程。

1.1 第一个AutoCAD程序

1.1.1 说明

让我们以在 AutoCAD 最下端的命令行窗口内显示一行欢迎文本来开始 .NET 开发 AutoCAD 的旅程。在这一节中, 我们将使用 Visual Studio 2010 来创建一个新的类库项目。通过这个项目, 你可以生成一个能被 AutoCAD 装载的 .NET dll 文件。这个 dll 文件会向 AutoCAD 加入一个名为 Hello 的新命令。当用户执行这个命令后, 在 AutoCAD 命令行上将显示“欢迎进入 .NET 开发 AutoCAD 的世界!”的字样。

1.1.2 思路

在开始正式的编写程序之前, 让我们首先来了解一下 AutoCAD 与 Visual Studio 版本之间的对应关系。表 1.1 列出了 AutoCAD、.NET 及 Visual Studio 版本之间的对应关系。

如果使用 C++ 来开发 AutoCAD 程序, 那么你就必须严格遵守表 1.1 对 Visual Studio 版本的规定。比如, 你要开发 AutoCAD 2008 的程序, 那你就必须使用 Visual Studio 2005, 高版本的 Visual Studio 2008 或低版本的 Visual Studio 2002 都不行。但 .NET 对 Visual Studio 的版本就没有任何限制了, 你可以使用最新版本的 Visual Studio 2010 对任何版本的 AutoCAD 进

行二次开发。

表 1.1 AutoCAD、.NET 与 Visual Studio 版本对应关系

AutoCAD 版本 (版本号)	.NET 版本	Visual Studio 版本
2004~2006(R16.0~R16.2)	1.0	Visual Studio.NET 2002
2007~2008(R17.0~R17.1)	2.0	Visual Studio 2005
2009(R17.2)	2.0	Visual Studio 2005 SP1
2010~2011(R18.0~R18.1)	3.5	Visual Studio 2008 SP1
2012(R18.2)	4.0	Visual Studio 2010
2013(R19.0)	4.0	Visual Studio 2010 SP1

本书使用如下编程环境。

- 操作系统: Windows XP。
- AutoCAD 版本: 主要是 AutoCAD 2008, 部分新功能使用 AutoCAD 2012。
- Visual Studio 版本: Visual Studio 2010 SP1。
- 开发语言: C#。



提示

由于 AutoCAD 2008 存在一定的 bug, 因此请大家到下面的网址下载 AutoCAD 2008 的补丁程序 Service Pack 1: <http://images.autodesk.com/adsk/files/autocad2008lockedsp1.exe>。本书所用的 AutoCAD 2008 都是打了这个补丁程序的。

1.1.3 步骤

(1) 新建项目。启动 Visual Studio 2010, 选择【文件/新建/项目】菜单, 在项目类型中选择 Visual C#语言, 然后在模板列表中选择【类库】项目。在下面的【名称】文本框中输入项目名称 (本例中为“Hello”), 在【位置】文本框中输入项目保存的位置, 你可以通过右面的【浏览】按钮选择你想要的位置, 在【解决方案名称】对话框中输入 Chap01。将【为解决方案创建目录】的复选框设置为选中状态, 完成上述设置后, 单击【确定】按钮来创建项目, 如图 1.1 所示。

(2) 添加对 acdbmgd.dll 和 acmgd.dll 程序集的引用。程序集有时也称为组件, 是.NET 的一个专有名词。acdbmgd.dll 包含 ObjectDBX 托管类, 用于 AutoCAD 数据库服务和操作 DWG 文件, 而 acmgd.dll 包含 ObjectARX 托管类, 它与 AutoCAD 应用程序有关。要使用.NET 进行 AutoCAD 的开发, 必须使用 AutoCAD .NET 托管封装类, 而这些托管封装类就位于这两个程序集中。所以你必须在新创建的项目中添加对这两个程序集的引用, 下面介绍具体的操作步骤。

在项目解决方案浏览器中用鼠标右键单击项目名“Hello”下的【引用】结点, 然后选择【添加引用】菜单, 如图 1.2 (a) 所示。



图 1.1 新建一个类库项目

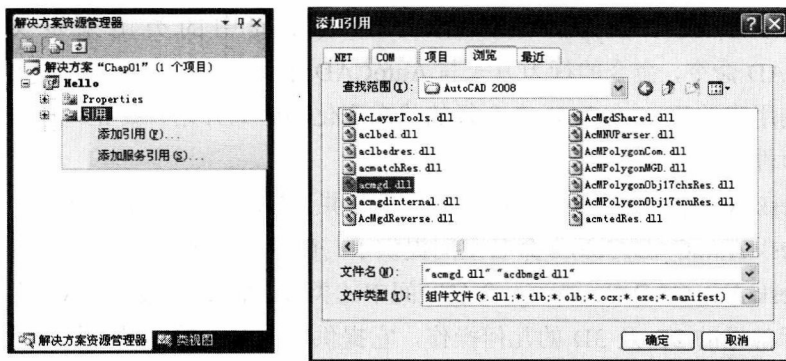


图 1.2 添加对 acdbmgd.dll 和 acmgd.dll 的引用

在弹出的【添加引用】对话框中选择【浏览】选项卡，然后选择 AutoCAD 2008 的安装目录（这里假定为 E:\Program Files\AutoCAD 2008）。

在 AutoCAD 2008 的安装目录下选中 acdbmgd.dll 和 acmgd.dll 这两个文件（使用 Ctrl 键实现多重选择），单击【确定】按钮，如图 1.2 (b) 所示。

通过上面的步骤，acdbmgd.dll 和 acmgd.dll 程序集就被加入进来了，在解决方案资源管理器窗口你可以进行查看，如图 1.3 所示。

(3) 使用对象浏览器（可以通过 Visual Studio 2010 的【视图/对象浏览器】菜单项访问）来浏览上面加入的程序集所提供的类，这些类都位于以 Autodesk.AutoCAD 开头的命名空间内，如图 1.4 所示。

下面来简要说明图 1.4 中常用的命名空间：

- Autodesk.AutoCAD.DatabaseServices 命名空间中的类表示组成 AutoCAD 图形数据库的元素，包括有图形界面的对象（也就是实体如直线、圆等）和非图形界面对象（如层、线型和文字样式等）。



图 1.3 查看添加的引用



图 1.4 Visual Studio 对象浏览器

- Autodesk.AutoCAD.Runtime 命名空间中的类提供了系统级别的功能如 DLL 初始化和运行时类的注册与确认。
- Autodesk.AutoCAD.ApplicationServices 命名空间中的类用来定义和注册新的 AutoCAD 命令，命令的行为方式与 AutoCAD 本身的命令一样。该命名空间还包含了一系列用来监视 AutoCAD 命令行的状态变化和诸如开始、终止或取消命令时通知程序的事件。
- Autodesk.AutoCAD.EditorInput 命名空间提供了与用户交互有关的类。
- Autodesk.AutoCAD.Colors 命名空间提供了与颜色有关的类。
- Autodesk.AutoCAD.Geometry 命名空间中的类被 DatabaseServices 命名空间中的类用来执行常见的 2D 及 3D 的几何操作，它提供了一系列的工具类如向量、矩阵、基本的几何对象（如点、曲线和面）。
- Autodesk.AutoCAD.GraphicsInterface 命名空间中的类表示绘制 AutoCAD 实体所使用的图形接口。这些类用于实体对象(Entity 类)的成员函数 WorldDraw、ViewportDraw 和 SaveAs，它们都是标准实体接口的组成部分。
- Autodesk.AutoCAD.PlottingServices 命名空间中的类用于打印。
- Autodesk.AutoCAD.Windows 命名空间中的类可以用来访问 AutoCAD 的对话框(如线型和颜色对话框)，它还提供了一些接口用于 AutoCAD 可扩展的用户界面对象如面板、托盘项和状态栏。你还可以通过 Autodesk.AutoCAD.Windows.Visuals 类来获取 AutoCAD 中表示“拾取点”、“拾取对象”及 AutoCAD Logo 标志的图片。

在本节中，我们将使用 Autodesk.AutoCAD.EditorInput 命名空间中的 Editor 类的一个实例在 AutoCAD 命令行中显示文本。Editor 类拥有访问 AutoCAD 命令行的相关方法，它还包括选择对象和其他一些重要的功能。

(4) 在所生成项目的 Class1.cs 文件中，一个公有类 Class1 已经被系统自动创建了，接下来向这个类加入命令。要加入命令，你必须使用上面导入的 AutoCAD .NET 托管封装类。在 Class1 类的声明语句（位于 Class1.cs 文件的顶部）之前，导入 ApplicationServices, EditorInput 和 Runtime 这三个命名空间，代码如下：

```
using Autodesk.AutoCAD.ApplicationServices;  
using Autodesk.AutoCAD.EditorInput;  
using Autodesk.AutoCAD.Runtime;
```

(5) 接下来在 Class1 类中加入命令 Hello。要加入能在 AutoCAD 中调用的命令，你必须使用 CommandMethod 属性，这个属性由 Runtime 命名空间提供。在类 Class1 中加入下列属性和函数：

```
public class Class1  
{  
    [CommandMethod("Hello")]  
    public void Hello()  
    {  
    }  
}
```

(6) 当 Hello 命令在 AutoCAD 中运行的时候，上面定义的 Hello 函数就会被调用。在 Hello 函数中加入以下代码：

```
// 获取当前活动文档的 Editor 对象，也就是命令行  
Editor ed=Application.DocumentManager.MdiActiveDocument.Editor;  
// 调用 Editor 对象的 WriteMessage 函数在命令行上显示文本  
ed.WriteMessage("欢迎进入 .NET 开发 AutoCAD 的世界！");
```

在这个函数中，一个 Editor 类的实例将被创建，AutoCAD 当前活动文档的 Editor 对象可以使用 Application 类来访问。当 Editor 对象被创建后，你可以使用它的 WriteMessage 函数在命令行中显示文本。

1.1.4 效果

(1) 选择 Visual Studio 的【生成/生成解决方案】菜单项或按 F6 快捷键编译程序，如果在 Visual Studio 左下角的状态栏显示“生成成功”，那就表示程序已经被成功编译。

(2) 启动 AutoCAD 2008，使用 NetLoad 命令装载编译好的托管程序。在 AutoCAD 命令行中执行 NetLoad 命令，这时会出现【选择.NET 程序集】的对话框，选择上一步生成的 Hello.dll 文件（位于..\Hello\bin\debug 文件夹），然后打开它，如图 1.5 所示。

在命令行中输入 Hello 并按 Enter 键，在命令行中并没有出现程序中定义的欢迎字样，而是显示了如图 1.6 所示的错误信息。

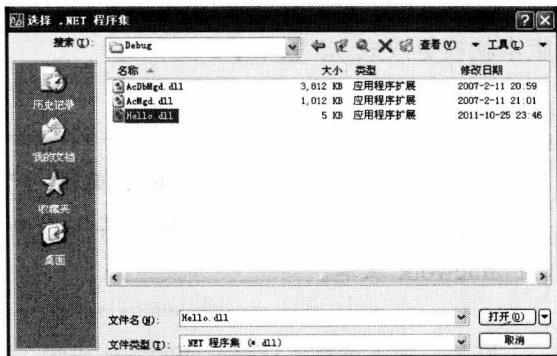


图 1.5 选择编译好的.NET 程序集


```
命令: netload
无法加载程序集。错误详细信息: System.BadImageFormatException:
未能加载文件或程序集“file:///F:\Hello\Hello\bin\Debug\Hello.dll”或它的某一个依赖项。生成此程序集的运行时代比当前加载的
运行时新, 无法加载此程序集。
文件名:“file:///F:\Hello\Hello\bin\Debug\Hello.dll”
```

图 1.6 显示无法加载程序集错误

这个问题是由于 Visual Studio 2010 的 .NET 版本是 4.0, 而 AutoCAD 2008 的 .NET 版本是 2.0, 因此会出现不兼容的情况。但是不要过分担心, 可以通过设置 .NET Framework 版本的方法来解决这个问题。

用鼠标右键单击项目名 Hello, 选择最下面的【属性】菜单, 在出现的【属性】面板中选择【应用程序】选项卡(一般为默认), 在“目标框架”下拉框中将 .NET Framework 版本改为 3.5, 如图 1.7 (a) 所示。

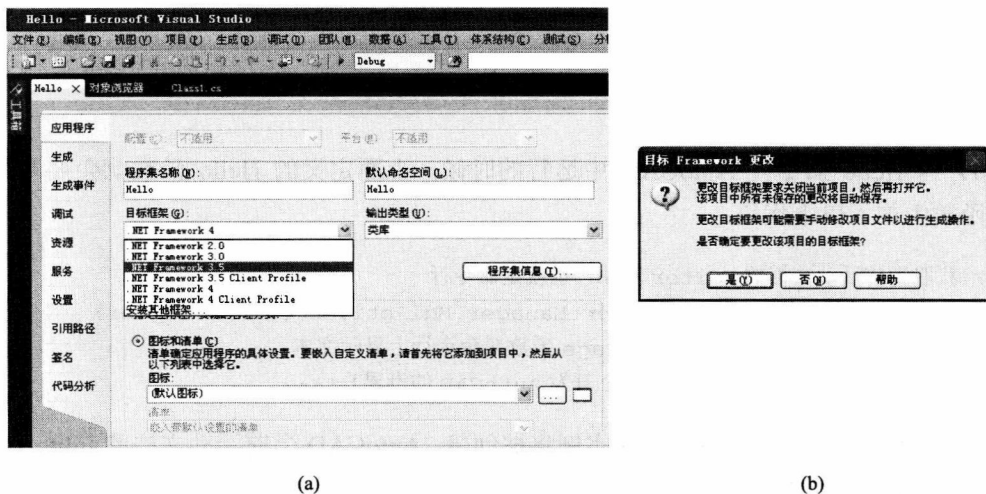


图 1.7 更改 .NET Framework 版本

这时会出现如图 1.7 (b) 所示的警告对话框, 单击【是】按钮确定要更改项目的目标框架。但这里工作还没有做完, 你会发现“Microsoft.CSharp”引用前会出现一个黄色的感叹号, 同时 Visual Studio 底部会出现一个“错误列表”面板, 提示移除“Microsoft.CSharp”引用。在“Microsoft.CSharp”引用上用鼠标右键单击, 选择【移除】菜单项, 这样 .NET Framework 版本就被更改成 3.5 版本了。



提示

细心的读者可能已经发现前面图 1.7 “新建项目”对话框的顶部有一个“.NET Framework 4”字样的下拉框, 其实在新建项目的时候就可以选择你要编译的 .NET Framework 版本。

好了, 请再重复执行最前面的操作步骤, 在 Visual Studio 2010 中通过 F6 键编译程序, 启动 AutoCAD 2008 并载入 Hello.dll 文件, 在命令行中输入 Hello, 这次应该会在命令行中显示程序中定义的欢迎字样了, 如图 1.8 所示。