

ASP.NET+ADO.NET

项目开发实例

樊建 编著

全部源代码下载

掌握面向对象开发方法

以网站新闻管理系统为开发实例



清华大学出版社

ASP.NET+ADO.NET 项目开发实例

樊 建 编著

清华大学出版社

北 京

内 容 简 介

在当今的软件开发中,面向对象的软件开发是一种相当先进的软件开发方法,在一个项目的生命周期里,要经历许多阶段,只有这些阶段完整地结合才能更好地实施项目。ASP.NET 的推出更是在 IT 业掀起了一股强劲的 .NET 旋风。本书旨在引导读者深入了解面向对象方法,使读者掌握用 ASP.NET+ADO.NET 技术构建新闻管理系统的技能。

本书采用面向对象方法构建一个完整的新闻系统。全书共分为 12 章,第 1 章~第 3 章主要介绍面向对象设计所经历各个阶段、具体分析和数据库设计过程。第 4 章主要介绍软件开发规范文档的制作。第 5 章~第 10 章介绍新闻系统实例的具体制作。第 11 章介绍相关控件的使用和数据库操作。第 12 章介绍新闻系统的系统配置。

本书适用于具有一定 ASP 和 ASP.NET 基础又想深入了解项目开发和 ASP.NET 的读者,也可以作为初学者的参考资料和培训教材。

版权所有,翻印必究。

本书封面贴有清华大学出版社激光防伪标签,无标签者不得销售。

图书在版编目(CIP)数据

ASP.NET+ADO.NET 项目开发实例/樊建编著.—北京:清华大学出版社,2004
ISBN 7-302-07949-8

I. A… II. 樊… III. ①主页制作—程序设计 ②数据库—接口—程序设计
IV. ①TP393.092②TP311.11

中国版本图书馆 CIP 数据核字(2004)第 001613 号

出 版 者:清华大学出版社 地 址:北京清华大学学研大厦

<http://www.tup.com.cn> 邮 编:100084

社 总 机:010-62770175 客 户 服 务:010-62776969

责任编辑:杨作梅

封面设计:陈刘源

印 刷 者:北京四季青印刷厂

装 订 者:三河市化甲屯小学装订二厂

发 行 者:新华书店总店北京发行所

开 本:185×260 印张:25.75 字数:650 千字

版 次:2004 年 3 月第 1 版 2004 年 3 月第 1 次印刷

书 号:ISBN 7-302-07949-8/TP·5772

印 数:1~5000

定 价:32.00 元

本书如存在文字不清、漏印以及缺页、倒页、脱页等印装质量问题,请与清华大学出版社出版部联系调换。联系电话:(010)62770175-3103 或(010)62795704

前 言

ASP.NET 是一种统一的 Web 平台，它提供了生成企业级应用程序所必需的全部服务。同时 ASP.NET 也是基于 .NET 框架生成的，因此用户可以使用任何与公共语言运行库兼容的语言(包括 Microsoft Visual Basic、Visual C# 和 JScript .NET)来创作 ASP.NET 应用程序。

ADO.NET 提供了对 Microsoft SQL Server 等数据源，以及通过 OLE DB 和 XML 公开的数据源的一致性访问。数据共享用户应用程序可以使用 ADO.NET 来连接到这些数据源，并检索、操作和更新数据。ADO.NET 借用 XML 的力量来提供对数据的断开式访问。ADO.NET 的设计与 .NET 框架中 XML 类的设计是相互关联的——它们都是同一个结构的组件。

在当今的软件开发中，面向对象的软件开发是一种相当先进的软件开发方法，在一个项目的生命周期里，要经历许多个阶段，只有这些阶段完整地结合起来才能很好地完成项目。

面向对象分析(Object-Oriented Analysis, OOA)的主要任务是分析问题领域，找出问题解决方案，发现对象、分析对象的内部构成和外部关系，建立软件系统的对象模型。

面向对象设计(Object-Oriented Design, OOD)的主要任务是根据已确立的系统对象模型，运用面向对象技术，进行系统软件设计。

在软件开发过程中，主要的开发过程包括：

- 可行性研究
- 项目开发计划
- 软件需求
- 数据要求
- 概要设计
- 数据库设计
- 详细设计
- 测试计划

为了使读者迅速掌握 ASP.NET 和 ADO.NET 技术、面向对象的开发方法以及了解整个软件开发过程中的各个阶段，本书以新闻系统为实例，采用面向对象方法分析制作，并运用 ASP.NET+ADO.NET 技术进行开发，每章都有一个独立的主题，全书共分 12 章，涵盖了面向对象方法开发系统的主要阶段。

第 1 章 主要工具方法概述

本章主要介绍 .NET 家族中 ASP.NET、C# 语言、CASE 工具及数据库工具 SQL Server 的发展历史和特性，以及所使用的范围。

第 2 章 面向对象系统分析与设计

面向对象是一种全新的软件开发方法，其基本技术是使用对象、类、继承、封闭、消息等基本概念进行程序设计和实现，UML 是编制软件蓝图的标准化语言，用于复杂软件系

统的各种成分的可视化,说明、系统模型构造和软件文档建立。

本章就从整体上讲述面向对象设计所经历各阶段以及各阶段的主要内容,重点讲解新闻系统的分析过程和 UML 的制作。

第 3 章 面向对象的数据库设计

数据库设计在系统中也处于重要的地位,只有好的数据库设计,才能构造出强健稳定的系统。本章介绍新闻系统数据库的设计原则、表设计、存储过程设计和索引设计等。

第 4 章 软件开发规范文档

在软件开发过程中,文档的编制确实是非常必要的,一个软件开发单位应该根据本单位经营承包的应用软件的专业特点和本单位的管理能力,制定一个文档编制实施规定,说明在什么情况下应该编制哪些文档。本章主要介绍这些相关文档的编制。

第 5 章 新闻显示模块

本章主要讲解新闻系统的前台制作,包括显示头条新闻、最新更新新闻、动态显示新闻分类、分类显示新闻、搜索新闻、显示新闻详细内容和添加新闻评论。

第 6 章 后台管理登录模块

本章主要是对后台管理登录模块制作的介绍,包括验证用户并取得权限、根据管理员权限显示栏目、显示系统信息和管理员排行。

第 7 章 新闻模块

本章主要是对系统中新闻模块制作的介绍,包括添加新闻、管理新闻、管理新闻评论、新闻审核和新闻转移。

第 8 章 分类管理模块

本章主要是对系统中分类模块的介绍,包括添加分类和管理分类。

第 9 章 用户管理模块

本章介绍用户管理,主要包括添加管理员、删除管理员、修改管理员信息和修改管理员权限等。

第 10 章 文件管理模块

本章介绍文件管理,主要包括显示文件信息和删除文件信息模块。

第 11 章 相关控件的使用和数据库操作

介绍 ASP.NET 中的几个相当重要的控件,如 DataList 和 DataGrid。并将 ADO.NET 中的主要操作方法介绍给大家。

第 12 章 系统配置

对新闻系统中 Web.Config 文件进行相应说明,并详细介绍如何安装配置新闻系统。

通过对这 12 章的学习,读者能迅速掌握 ASP.NET 和 ADO.NET,并且能够采用面向对象方法开发系统。

为了便于读书学习,可从 www.wenyuan.com.cn 上下载书中所用到的 CASE 工具系统分析文件和新闻系统源代码。

目 录

第 1 章 主要工具方法概述 1	
1.1 .NET 框架概述..... 1	
1.1.1 公共语言运行库的功能..... 2	
1.1.2 .NET 框架类库..... 3	
1.1.3 客户端应用程序开发 8	
1.1.4 服务器应用程序开发 9	
1.2 ASP.NET 10	
1.2.1 ASP.NET 简介 10	
1.2.2 ASP.NET 的设计理念 11	
1.2.3 应用程序配置文件 Web.Config..... 12	
1.3 ADO.NET 16	
1.3.1 ADO.NET 的设计目标 17	
1.3.2 ADO.NET 结构..... 17	
1.3.3 .NET 数据提供程序..... 19	
1.3.4 ADO.NET DataSet 21	
1.4 Visual C# .NET 2003..... 22	
1.4.1 概述 22	
1.4.2 Visual C# .NET 2003 功能..... 23	
1.4.3 开发新闻系统所用功能..... 24	
1.5 数据库工具概述..... 27	
1.5.1 SQL Server 2000 组件概述..... 28	
1.5.2 SQL Server 2000 的特性..... 29	
1.5.3 SQL Server 和 XML 支持 30	
1.5.4 SQL Server 在新闻系统中 所用功能 30	
1.6 CASE 工具 33	
1.6.1 Rational Rose 2003 33	
1.6.2 PowerDesigner 9.5..... 36	
第 2 章 面向对象系统分析与设计 40	
2.1 UML 综述..... 40	
2.1.1 标准建模语言 UML 的出现 40	
2.1.2 标准建模语言 UML 的内容 42	
2.1.3 标准建模语言 UML 的应用领域 43	
2.2 Rational 统一开发过程 44	
2.2.1 “统一开发过程”概述 44	
2.2.2 迭代式的和增量的统一 开发过程 46	
2.2.3 “统一开发过程”的 生命周期 47	
2.3 需求分析..... 49	
2.3.1 概念 49	
2.3.2 需求分析的任务..... 49	
2.3.3 需求分析过程..... 50	
2.3.4 需求的类型 51	
2.3.5 需求分析的原则..... 52	
2.3.6 需求分析人员和用户的 合作关系 53	
2.3.7 需求文档 55	
2.3.8 需求分析的过程..... 57	
2.4 新闻系统需求分析与设计 59	
2.4.1 需求分析 59	
2.4.2 功能设计 60	
2.5 新闻系统 UML 图的制作 62	
2.5.1 新闻系统用例图..... 62	
2.5.2 新闻系统序列图..... 65	
2.5.3 新闻系统类图..... 70	
2.6 新闻系统的页面设计 71	
2.6.1 前台显示页面设计..... 72	
2.6.2 后台管理区 74	
第 3 章 面向对象的数据库设计 83	
3.1 面向对象的数据库设计简介 83	
3.1.1 概念的区分 83	
3.1.2 数据库开发步骤..... 83	

3.1.3	应用对象模型与 RDBMS 模型的映射	84	4.4.2	数据的逻辑描述	131
3.1.4	面向对象关系数据库 设计效果	84	4.4.3	数据的采集	131
3.1.5	设计建议	85	4.5	测试计划	132
3.2	数据库设计规范	86	4.5.1	引言	132
3.2.1	数据库应用结构	86	4.5.2	计划	133
3.2.2	数据库结构原则	87	4.5.3	测试设计说明	134
3.2.3	数据库命名原则	88	4.5.4	评价准则	134
3.2.4	数据库正规化和设计技巧	89	4.6	概要设计说明书	135
3.3	新闻系统数据库设计	93	4.6.1	引言	135
3.3.1	新闻系统数据表设计	93	4.6.2	总体设计	135
3.3.2	新闻系统关系图	98	4.6.3	接口设计	136
3.3.3	新闻系统索引设计	100	4.6.4	运行设计	136
3.3.4	存储过程的设计	101	4.6.5	系统数据结构设计	136
			4.6.6	系统出错处理设计	137
第 4 章	软件开发规范文档	117	4.7	详细设计说明书	137
4.1	可行性研究报告	120	4.7.1	引言	137
4.1.1	引言	120	4.7.2	程序系统的结构	138
4.1.2	可行性研究的前提	120	4.7.3	程序 1(标识符)设计说明	138
4.1.3	对现有系统的分析	121	4.7.4	程序 2(标识符)设计说明	139
4.1.4	所建议的系统	122	4.8	数据库设计说明书	139
4.1.5	可选择的其他系统方案	123	4.8.1	引言	139
4.1.6	投资及效益分析	123	4.8.2	外部设计	140
4.1.7	社会因素方面的可行性	125	4.8.3	结构设计	140
4.1.8	结论	125	4.8.4	运用设计	141
4.2	项目开发计划	126	4.9	用户手册	141
4.2.1	引言	126	4.9.1	引言	141
4.2.2	项目概述	126	4.9.2	用途	142
4.2.3	实施计划	127	4.9.3	运行环境	142
4.2.4	支持条件	127	4.9.4	使用过程	143
4.2.5	专题计划要点	128	4.10	操作手册	145
4.3	软件需求说明书	128	4.10.1	引言	145
4.3.1	引言	128	4.10.2	软件描述	145
4.3.2	任务概述	128	4.10.3	安装与初始化	145
4.3.3	需求规定	129	4.10.4	运行说明	146
4.3.4	运行环境规定	130	4.10.5	运行 2(标识符)说明	147
4.4	数据要求说明书	130	4.10.6	非常规过程	147
4.4.1	引言	130	4.10.7	远程操作	147
			4.11	模块开发卷宗	147
			4.11.1	标题	147

4.11.2	模块开发情况表	147	5.5	分类显示新闻.....	167
4.11.3	功能说明	147	5.5.1	概述	167
4.11.4	设计说明	148	5.5.2	代码	168
4.11.5	原代码清单	148	5.6	搜索新闻.....	172
4.11.6	测试说明	148	5.6.1	概述	172
4.11.7	复审的结论	148	5.6.2	代码	173
4.12	测试分析报告.....	148	5.7	显示新闻详细信息.....	176
4.12.1	引言	148	5.7.1	概述	176
4.12.2	测试概要	149	5.7.2	代码	177
4.12.3	测试结果及发现	149	5.8	新闻评论.....	181
4.12.4	对软件功能的结论	149	5.8.1	概述	181
4.12.5	分析摘要	149	5.8.2	代码	182
4.12.6	测试资源消耗	150	第 6 章	后台管理登录模块	189
4.13	开发进度月报.....	150	6.1	验证管理用户并取得用户权限	189
4.13.1	标题	150	6.1.1	概述	189
4.13.2	工程进度与状态	150	6.1.2	代码	190
4.13.3	资源耗用与状态	150	6.2	根据管理员用户权限显示栏目	195
4.13.4	经费支出与状态	151	6.2.1	概述	195
4.13.5	下个月的工作计划	151	6.2.2	代码	195
4.13.6	建议	151	6.3	显示系统信息.....	199
4.14	项目开发总结报告.....	152	6.3.1	概述	199
4.14.1	引言	152	6.3.2	代码	200
4.14.2	实际开发结果	152	6.4	管理员工作排行.....	204
4.14.3	开发工作评价	153	6.4.1	概述	204
4.14.4	经验与教训	153	6.4.2	代码	205
第 5 章	新闻显示模块	154	第 7 章	新闻模块.....	208
5.1	显示头条新闻	154	7.1	添加新闻.....	208
5.1.1	描述	154	7.1.1	根据权限选择分类.....	208
5.1.2	代码	154	7.1.2	HTML 的显示	211
5.2	显示最新更新新闻.....	158	7.1.3	上传新闻图片.....	213
5.2.1	概述	158	7.1.4	预览新闻	215
5.2.2	代码	159	7.1.5	添加新闻	217
5.3	动态显示新闻分类.....	161	7.2	管理新闻.....	225
5.3.1	概述	161	7.2.1	显示新闻	225
5.3.2	代码	162	7.2.2	删除新闻	231
5.4	显示最热门新闻.....	165	7.2.3	修改新闻	238
5.4.1	概述	165	7.2.4	搜索新闻	249
5.4.2	代码	166			

7.3 管理新闻评论	253	10.1.1 概述	356
7.3.1 显示评论	253	10.1.2 代码	356
7.3.2 搜索评论	259	10.2 显示图片文件	361
7.3.3 删除评论	263	10.2.1 概述	361
7.4 新闻审核模块	266	10.2.2 代码	362
7.4.1 显示需审核新闻	266	10.3 删除图片文件	362
7.4.2 搜索审核新闻	272	10.3.1 概述	362
7.4.3 通过审核新闻	276	10.3.2 代码	363
7.4.4 删除审核新闻	281	第 11 章 相关控件的使用和数据库	
7.5 新闻转移模块	285	操作	366
7.5.1 显示新闻分类	285	11.1 DataGrid 控件	367
7.5.2 转移新闻	288	11.1.1 DataGrid 概述	367
第 8 章 分类管理模块	296	11.1.2 示例	369
8.1 添加分类	296	11.1.3 DataGrid 分页	371
8.1.1 概述	296	11.2 DataList 控件	376
8.1.2 代码	297	11.2.1 DataList 概述	376
8.2 管理分类	300	11.2.2 示例	378
8.2.1 显示所有分类	300	11.2.3 DataList 控件的分页	380
8.2.2 修改分类	305	11.3 ADO.NET 数据操作	384
8.2.3 删除新闻分类	312	11.3.1 ADO.NET 数据操作一	
第 9 章 用户管理模块	316	(insert, update, delete)	385
9.1 添加管理员	316	11.3.2 ADO.NET 数据操作二	
9.1.1 概述	316	(query, Parameters)	387
9.1.2 代码	317	11.3.3 ADO.NET 数据操作三	
9.2 用户管理	321	(存储过程, 数据集)	389
9.2.1 显示所有管理员信息	321	11.3.4 ADO.NET 数据操作四	
9.2.2 修改管理员信息	327	(表关联, DataAdapter)	393
9.2.3 修改管理员权限	335	第 12 章 系统配置	395
9.2.4 删除管理员用户	345	12.1 新闻系统 Web.config 文件	395
9.3 修改自身信息	348	12.2 配置新闻系统	395
第 10 章 文件管理模块	356		
10.1 显示文件信息	356		

第 1 章 主要工具方法概述

本书采用面向对象方法介绍网站新闻管理系统的分析与建立，并采用当今最新的 .NET 框架下的 ASP.NET 和 ADO.NET 技术来构建新闻系统。新闻系统的分析、设计和编码都离不开各种工具的支持，它们紧密配合才能高效率地制作程序。本章重点介绍 .NET 框架以及制作新闻系统所需的工具及相关功能。

1.1 .NET 框架概述

Microsoft 公司主推的 .NET 技术已经越来越受到重视，并且运用的范围也越来越广。.NET 框架是一种新的计算平台，它简化了在高度分布式 Internet 环境中的应用程序开发。.NET 框架旨在实现下列目标：

- 提供一个一致的面向对象的编程环境，无论对象代码是在本地存储和执行，还是在本地执行并在 Internet 上分布，或者是在远程执行。
- 提供一个将软件部署和版本控制冲突最小化的代码执行环境。
- 提供一个保证代码(包括由未知的或不完全受信任的第三方创建的代码)安全执行的代码执行环境。
- 提供一个可解决脚本环境或解释环境性能问题的代码执行环境。
- 使开发人员的经验在面对类型大不相同的应用程序(如基于 Windows 的应用程序和基于 Web 的应用程序)时保持一致。
- 按照工业标准生成所有通信，以确保基于 .NET 框架的代码可与任何其他代码集成。

.NET 框架具有两个主要组件：公共语言运行库和 .NET 框架类库。公共语言运行库是 .NET 框架的基础。运行库可以看作一个在执行时管理代码的代理，提供核心服务(如内存管理、线程管理和远程处理)，而且还强制实施严格的类型安全以及可确保安全性和可靠性的其他形式的代码准确性。以运行库为目标的代码称为托管代码，而不以运行库为目标的代码称为非托管代码。类库是一个综合性的面向对象的可重用类型集合，用户可以用它开发包含从传统的命令行或图形用户界面(GUI)应用程序到基于最新的 ASP.NET 技术(如 Web 窗体和 XML Web 服务)在内的应用程序。

.NET 框架可由非托管组件承载，这些组件将公共语言运行库加载到它们的进程中并启动托管代码的执行，从而创建一个可以同时利用托管和非托管功能的软件环境。.NET 框架不但提供若干个运行库宿主，而且还支持第三方运行库宿主的开发。

图 1.1 显示公共语言运行库、类库、应用程序之间以及与整个系统之间的关系。

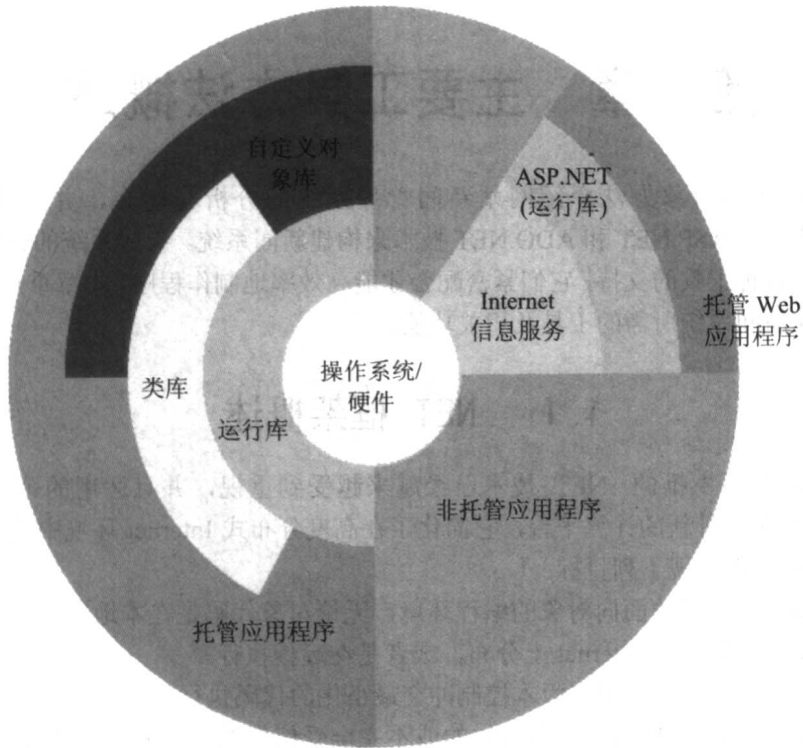


图 1.1 .NET 框架环境

1.1.1 公共语言运行库的功能

公共语言运行库管理内存、线程执行、代码执行、代码安全验证、编译及其他系统服务。这些功能是在公共语言运行库上运行的托管代码所固有的。

安全性取决于包括托管组件的来源(如 Internet、企业网络或本地计算机)在内的一些因素,托管组件被赋予不同程度的信任。

运行库强制实施代码访问安全。例如,用户可以相信嵌入在 Web 页中的可执行文件能够在屏幕上播放动画或音乐,但不能访问他们的个人数据、文件系统或网络。这样,运行库的安全性功能就使通过 Internet 部署的合法软件具有特别丰富的功能。

运行库还通过实现称为通用类型系统(CTS)的严格的类型验证和代码验证基础结构来加强代码可靠性。CTS 确保所有托管代码都是可以自我描述的。Microsoft 公司和各种第三方语言编译器生成符合 CTS 的托管代码。这意味着托管代码可在严格实施类型保真和类型安全的同时使用其他托管类型和实例。

此外,运行库的托管环境还消除了许多常见的软件问题。例如,运行库自动处理对象布局并管理对对象的引用,在不再使用它们时将它们释放。这种自动内存管理解决了两个最常见的应用程序错误:内存泄漏和无效内存引用。

运行库还提高了开发人员的工作效率。例如,开发人员可以用他们选择的开发语言编写应用程序,却仍能充分利用其他开发人员用其他语言编写的运行库、类库和组件。任何选择以运行库为目标的编译器供应商都可以这样做。以 .NET 框架为目标的语言编译器使得用该语言编写的现有代码可以使用 .NET 框架的功能,这大大减轻了现有应用程序迁移

过程的工作负担。

尽管运行库是为未来的软件设计的，但是它也支持现在和以前的软件。托管和非托管代码之间的互操作性使开发人员能够继续使用所需的 COM 组件和 DLL。

运行库旨在增强性能。尽管公共语言运行库提供许多标准运行库服务，但是它从不解释托管代码。一种称为实时(JIT)编译的功能使所有托管代码能够以它在其上执行的系统的本机语言运行。同时，内存管理器排除了出现零碎内存的可能性，并增大了内存引用区域以进一步提高性能。

最后，运行库可由高性能的服务器端应用程序，如 Microsoft SQL Server 和 Internet 信息服务(IIS)承载。此基础结构使用户在享受服务器优越性能的同时，能够使用托管代码编写业务逻辑。

1.1.2 .NET 框架类库

.NET 框架类库是一个与公共语言运行库紧密集成的可重用类型集合。该类库是面向对象的，并提供自己的托管代码可从中导出功能的类型。这不但使 .NET 框架类型易于使用，而且还减少了学习 .NET 框架的新功能所需要的时间。此外，第三方组件可与 .NET 框架中的类无缝集成。

例如，.NET 框架类实现一组可用于开发用户自己的集合类的接口。用户自己的类将与 .NET 框架中的类无缝地集成。

.NET 框架类型使用户能够完成一系列常见编程任务(包括字符串管理、数据收集、数据库连接及文件访问等任务)。除这些常见任务之外，类库还包括支持多种专用开发方案的类型。例如，可使用 .NET 框架开发下列类型的应用程序和服务：

- 控制台应用程序。
- 脚本应用程序或寄宿的应用程序。
- Windows GUI 应用程序(Windows 窗体)。
- ASP.NET 应用程序。
- XML Web 服务。
- Windows 服务。

.NET 框架包括类、接口和值类型，它们可加速和优化开发过程并提供对系统功能的访问。为推动语言间的互操作性，.NET 框架类型必须符合公共语言规范(CLS)，并由此可在任何编译器符合公共语言规范(CLS)的编程语言中使用。

.NET 框架类型是生成 .NET 应用程序、组件和控件的基础。.NET 框架包括的类型执行下列功能：

- 表示基础数据类型和异常。
- 封装数据结构。
- 执行 I/O。
- 访问关于加载类型的信息。
- 调用 .NET 框架安全检查。
- 提供数据访问、多客户端 GUI 和服务器控制的客户端 GUI。

.NET 框架提供一组丰富的接口、抽象类和具体(非抽象)类。可以按原样使用这些具体

类，或者从这些类派生自己的类。若要使用接口，既可以创建实现接口的类，也可以从某个实现接口的 .NET 框架类中派生类。

1. 命名约定

.NET 框架类型使用点语法命名方案，该方案隐含了层次结构的意思。此技术将相关类型分为不同的命名空间组，以便可以更容易地搜索和引用它们。全名的第一部分(最右边的点之前的内容)是命名空间名。全名的最后一部分是类型名。例如，System.Collections.ArrayList 表示 ArrayList 类型，该类型属于 System.Collections 命名空间。System.Collections 中的类型可用于操作对象集合。

此命名方案使扩展 .NET 框架的库开发人员可以轻松创建分层类型组，并用一致的、带有提示性的方式对其进行命名。库开发人员在创建命名空间的名称时应使用以下原则：

公司名称.技术名称

例如，Microsoft.Word 命名空间就符合此原则。

利用命名模式将相关类型分组为命名空间是生成和记录类库的一种非常有用的方式。但是，此命名方案对可见性、成员访问、继承、安全性或绑定无效。一个命名空间可以被划分在多个程序集中，而单个程序集可以包含来自多个命名空间的类型。程序集为公共语言运行库中的版本控制、部署、安全性、加载和可见性提供外形结构。

2. System 命名空间

System 命名空间是 .NET 框架中基本类型的根命名空间。此命名空间包括所有基础数据类型类：Object(继承层次结构的根)、Byte、Char、Array、Int32 和 String 等。在这些类型中，有许多与编程语言所使用的基元数据类型相对应。当使用 .NET 框架类型编写代码时，可以在应使用 .NET 框架基础数据类型时使用编程语言的相应关键字。

表 1.1 列出了 .NET 框架支持的一些值类型，简要描述了每个类型，并指出了 Visual Basic、C#和 C++托管扩展中的相应类型。表中还包括 Object 和 String 类的项，这些项在许多语言中都有相应的关键字。

表 1.1 .NET 框架值类型

类别	类名	说明	Visual Basic 数据类型	C#数据 类型	C++ 托管扩 展数据类型	JScript 数 据类型
整数	Byte	8 位的无符号整数	Byte	byte	char	byte
	Sbyte	8 位的有符号整数 不符合 CLS	Sbyte 无内置类型	sbyte	signed char	Sbyte
	Int16	16 位的有符号整数	Short	short	short	short
	Int32	32 位的有符号整数	Integer	int	int -或- long	int
	Int64	64 位的有符号整数	Long	long	__int64	long

续表

类别	类名	说明	Visual Basic 数据类型	C#数据 类型	C++ 托管扩 展数据类型	JScript 数 据类型
	UInt16	16 位的无符号整数 不符合 CLS	UInt16 无内置类型	ushort	unsigned short	UInt16
	UInt32	32 位的无符号整数 不符合 CLS	UInt32 无内置类型	uint	unsigned int -或- unsigned long	UInt32
	UInt64	64 位的无符号整数 不符合 CLS	UInt64 无内置类型	ulong	unsigned __int64	UInt64
浮点	Single	单精度(32 位)浮点 数字	Single	float	float	float
	Double	双精度(64 位)浮点 数字	Double	double	double	double
逻辑	Boolean	布尔值(真或假)	Boolean	bool	bool	bool
其他	Char	Unicode(16 位)字符	Char	char	wchar_t	char
	Decimal	96 位十进制值	Decimal	decimal	Decimal	Decimal
	IntPtr	大小取决于基础平 台(32 位平台上为 32 位值, 64 位平 台上为 64 位值)的有 符号整数	IntPtr 无内置类型	IntPtr 无内置 类型	IntPtr 无内置类型	IntPtr
	UIntPtr	大小取决于基础平 台的无符号整数(32 位平台上为 32 位 值, 64 位平台上为 64 位值) 不符合 CLS	UIntPtr 无内置类型	UIntPtr 无内置 类型	UIntPtr 无内置类型	UIntPtr
类对象	Object	对象层次结构的根	Object	object	Object*	Object
	String	Unicode 字符的不 变的定长字符串	String	string	String*	String
整数	Byte	8 位的无符号整数	Byte	byte	char	byte
	SByte	8 位的有符号整数 不符合 CLS	SByte 无内置类型	sbyte	signed char	SByte
	Int16	16 位的有符号整数	Short	short	short	short
	Int32	32 位的有符号整数	Integer	int	int 或 long	int

续表

类别	类名	说明	Visual Basic 数据类型	C#数据 类型	C++ 托管扩 展数据类型	JScript 数 据类型
	Int64	64 位的有符号整数	Long	long	int64	long
	UInt16	16 位的无符号整数 不符合 CLS	UInt16 无内置类型	ushort	unsigned short	UInt16
	UInt32	32 位的无符号整数 不符合 CLS	UInt32 无内置类型	uint	unsigned int 或 unsigned long	UInt32
	UInt64	64 位的无符号整数 不符合 CLS	UInt64 无内置类型	ulong	unsigned int64	UInt64
浮点	Single	单精度(32 位)浮点 数字	Single	float	float	float
	Double	双精度(64 位)浮点 数字	Double	double	double	double
逻辑	Boolean	布尔值(真或假)	Boolean	bool	bool	bool
其他	Char	Unicode(16 位)字符	Char	char	wchar_t	char
	Decimal	96 位十进制值	Decimal	decimal	Decimal	Decimal
	IntPtr	大小取决于基础平 台(32 位平台上为 32 位值, 64 位平 台上为 64 位值)的有 符号整数	IntPtr 无内置类型	IntPtr 无内置 类型	IntPtr 无内置类型	IntPtr
	UIntPtr	大小取决于基础平 台的无符号整数(32 位平台上为 32 位 值, 64 位平台上为 64 位值) 不符合 CLS	UIntPtr 无内置类型	UIntPtr 无内置 类型	UIntPtr 无内置类型	UIntPtr
类对象	Object	对象层次结构的根	Object	object	Object*	Object
	String	Unicode 字符的不变 的定长串	String	string	String*	String

除基础数据类型外, System 命名空间还包含近 100 个类, 范围从处理异常的类到处理核心运行时概念的类, 如应用程序域和垃圾回收器。

System 命名空间还包含许多二级命名空间。表 1.2 显示 System 命名空间包括的功能类别、每个类别中的二级命名空间和对每个命名空间的简要说明。尽管许多二级命名空间还包含其他命名空间, 但表 1.2 只包括部分三级命名空间。

表 1.2 System 命名空间

类 别	命名空间	功 能
组件模型	System.CodeDom	源代码文档的元素和结构的表示形式, 以及此代码的编译和处理
配置	System.ComponentModel	组件的实现, 包括授权和设计时调整
	System.Configuration	应用程序配置数据的检索
数据	System.Data	数据和数据源的访问与管理
	System.Xml	对处理 XML 的基于标准的支持
框架服务	System.Xml.Serialization	对象到 XML 的双向映射
	System.Diagnostics	应用程序的配置和诊断
	System.DirectoryServices	访问 Active Directory。该命名空间中的类可与任何 Active Directory 服务提供程序(例如, Internet 信息服务(IIS))一同使用
	System.Management	使用基于 Web 的企业管理(WBEM)标准的服务和应用程序管理工具
	System.Messaging	Microsoft 消息队列(MSMQ)访问和管理, 以及消息的发送和接收
	System.ServiceProcess	基于 Windows 的服务应用程序的安装和执行。不访问特定的服务, 如 Active Directory 或 Web 服务
全球化与本地化	System.Timers	基于时间间隔或更复杂的时间安排引发的事件
	System.Globalizatiion	对代码和资源的国际化与全球化的支持
网络	System.Resources	资源的管理和访问, 包括对本地化的支持
	System.Net	对通过网络发送和接收数据的支持, 包括用于公共网络协议的简单编程接口
公用任务	System.Collections	对象(如列表、队列、数组、哈希表和词典)的集合
	System.IO	基本数据流的访问和管理, 包括文件 I/O、内存 I/O 和独立存储
	System.Text	字符编码、字符转换和字符串操作
	System.Text.RegularExpressions	完全的正则表达式支持
反射	System.Threading	多线程编程支持, 包括锁定和同步
	System.Reflection	对类型元数据的访问和类型的动态创建和调用
多客户端 GUI	System.Drawing	丰富的二维图形功能和对 GDI+ 的访问
	System.Windows.Forms	基于 Windows 的应用程序的丰富用户界面功能

续表

类别	命名空间	功能
编译服务	System.Runtime.CompilerServices	对以运行库为目标的编译器的支持
	System.Runtime.InteropServices	对与 COM 和其他非托管代码的互操作性的支持
	System.Runtime.Remoting	对创建紧耦合分布式应用程序或松耦合分布式应用程序的支持
	System.Runtime.Serialization	对象序列化和反序列化, 包括二进制和 SOAP 编码支持
.NET 框架安全性	System.Security	对 .NET 框架安全系统的基础机制的访问, 包括策略解析、堆栈遍历和权限
	System.Security.Cryptography	加密服务, 包括数据的编码和解码、散列、随机数生成、消息身份验证和数字签名的构成
Web 服务	system.web	对 Web 服务器和客户端的管理、通信和设计的支持。为 ASP.NET 提供核心结构, 包括 Web 窗体支持
	System.Web.Services	对基于 SOAP 的 Web 服务的客户端和服务端支持

1.1.3 客户端应用程序开发

客户端应用程序在基于 Windows 的编程中最接近于传统风格的应用程序。这些是在桌面上显示窗口或窗体从而使用户能够执行任务的应用程序类型。客户端应用程序包括字处理器和电子表格等应用程序以及自定义的业务应用程序(如数据输入工具和报告工具等)。客户端应用程序通常使用窗口、菜单、按钮和其他 GUI 元素, 并且它们可能访问本地资源(如文件系统)和外围设备(如打印机)。

另一种客户端应用程序是作为 Web 页并通过 Internet 部署的传统 ActiveX 控件(现在被托管 Windows 窗体控件所替代)。此类应用程序类似于其他客户端应用程序: 它在本机执行, 可以访问本地资源, 并包含图形元素。

过去, 开发人员使用 C/C++ 与 Microsoft 基础类(MFC)或应用程序快速开发(RAD)环境(如 Microsoft Visual Basic)来创建客户端应用程序, 而 .NET 框架将这些现有产品的特点合并到了单个且一致的开发环境中, 该环境大大简化了客户端应用程序的开发。

包含在 .NET 框架中的 Windows 窗体类旨在用于 GUI 开发。用户可以轻松创建具有适应多变的商业需求所需的灵活性的命令窗口、按钮、菜单、工具栏和其他屏幕元素。

和 ActiveX 控件不同, Windows 窗体控件具有对用户计算机的不完全受信任的访问权限。这意味着二进制代码或在本机执行的代码可访问用户系统上的某些资源(如 GUI 元素和受限制的文件访问), 而不能访问或危害其他资源。由于具有代码访问安全性, 许多曾经需要安装在用户系统上的应用程序现在可以通过 Web 安全地部署。应用程序可以像 Web 页