

# 《ATL 开发指南》

## 图书基本信息

书名：《ATL 开发指南》

13位ISBN编号：9787505362987

10位ISBN编号：7505362984

出版时间：2000-11

出版社：电子工业出版社

作者：（美）Tom Armstrong Ron Patton

页数：511

译者：董梁，丁杰等

版权说明：本站所提供下载的PDF图书仅提供预览和简介以及在线试读，请支持正版图书。

更多资源请访问：[www.tushu000.com](http://www.tushu000.com)

# 《ATL 开发指南》

## 内容概要

本书是介绍使用ATL进行软件开发的参考用书。全书分为十三章：第一章引入了C++模板的概念；第二章详细介绍了组件对象模型（COM）；第三章介绍了活动模板库（ATL）的概念和使用方法；第四章涉及接口的实现和线程之间的参数调度等细节问题；第五章探讨了代码复用的两种方法——包容和集合；第六章讨论了关于自动化的问题；第七章说明了事件和连接点的管理；第八章围绕ActiveX控件展开了详细的论述；第九章介绍了COM中的两种实体——枚举器和集合；第十章研究了COM里的线程管理；第十一章讨论了OLE DB同ATL之间的关系；第十二章介绍了两种GUI接口——对话框和视窗；最后一章解释了COM+的一些基本概念。

由于书中的内容非常庞杂，并且涉及到Visual C++开发的一些内核问题，所以本书对读者有一定的要求。我们希望你开始阅读之前，已经了解COM、ActiveX控件的基本知识，并且具有使用Visual C++开发应用程序的实践经验。本书的主要对象是程序设计或开发人员，同时也可以作为大专院校计算机专业师生和计算机爱好者的参考资料。

## 书籍目录

### 前言

### 第一章 使用C++模板开发应用程序

#### 1.1 模板简介

##### 1.1.1 函数模板

##### 1.1.2 类模板

#### 1.2 基于模板的堆栈类

#### 1.3 成员函数的实现

#### 1.4 不定类型的模板参数

#### 1.5 基于模板的复用技术

#### 1.6 ATL如何使用模板

#### 1.7 小结

### 第二章 组件对象模型 (COM) 简介

#### 2.1 软件组件及其属性

##### 2.1.1 同语言的无关性：二进制的设计标准

##### 2.1.2 版本升级的鲁棒性

##### 2.1.3 位置的透明性

##### 2.1.4 面向对象的特性

#### 2.2 COM模型的原始实现

##### 2.2.1 客户程序和服务器程序

##### 2.2.2 OLE和ActiveX

##### 2.2.3 Microsoft Transaction Server(MTS)

#### 2.3 COM的精髓：接口

##### 2.3.1 C++的虚函数表

##### 2.3.2 COM的接口

#### 2.4 对COM接口的访问

#### 2.5 组件的多接口特性

##### 2.5.1 标准COM接口

##### 2.5.2 HRESULT

##### 2.5.3 IUnknown的实现

##### 2.5.4 组件的生存期

##### 2.5.5 全局特有标识符

##### 2.5.6 标题标识符

##### 2.5.7 注册表

##### 2.5.8 组件的类别

##### 2.5.9 OLEVIEW

##### 2.5.10 组件的宿主文件

##### 2.5.11 类工厂

#### 2.6 基于COM的复用技术

##### 2.6.1 包容 (Containment)

##### 2.6.2 集合 (Aggregation)

#### 2.7 COM的API函数

##### 2.7.1 CoInitialize和CoInitializeEx

##### 2.7.2 CoUninitialize

##### 2.7.3 CoRegisterClassObject

##### 2.7.4 CoGetClassObject

##### 2.7.5 CoCreateInstance

##### 2.7.6 DllCanUnloadNow

2.7.7 DllGetClassObject

2.8 其他涉及COM的问题

2.8.1 C++里关于COM的宏：STDMETHOD和STDMETHODIMP

2.8.2 COM和Unicode

2.8.3 COM和多态性

2.9 示例：建立一个简单的COM客户程序和服务器程序

2.9.1 第一步：创建Visual C++工程

2.9.2 第二步：创建IMATH.H文件

2.9.3 第三步：声明组件和类工厂

2.9.4 第四步：组件类和类工厂类的实现

2.9.5 第五步：创建组件的宿主（SERVER.CPP）

2.9.6 第六步：添加对自注册和组件类别的支持

2.9.7 第七步：标准入口点的导出

2.9.8 第八步：将文件插入到工程里并建立该工程

2.9.9 第九步：使用OLEVIEW测试Math组件

2.9.10 第十步：建立一个简单的COM客户

2.9.11 第十一步：建立客户工程

2.9.12 对服务器程序进行调试

2.9.13 示例中忽略的东西

2.10 小结

第三章 活动模板库

3.1 ATL的基本特征

3.2 ATL和MFC

3.3 ATL框架结构概述

3.3.1 ATL的实现

3.3.2 组件的宿主支持

3.3.3 对IUnknown的支持

3.3.4 对类工厂的支持

3.3.5 对COM程序开发的其他方面的支持

3.4 ATL向导程序的使用

3.4.1 ATL COM AppWizard

3.4.2 ATL对象向导程序

3.4.3 线程管理模型

3.4.4 接口

3.4.5 集合

3.4.6 对ISupportErrorInfo的支持

3.4.7 对连接点的支持

3.4.8 自由线程调度器

3.5 ObjectWizard里的其他选项

3.6 一个服务器程序示例

3.6.1 ATL工程的结构

3.6.2 STDAFX.H和STDAFX.CPP

3.6.3 RESOURCE.H和ProjectName.rc

3.6.4 ProjectName.CPP、ProjectName.H和ProjectName.DEF

3.6.5 ProjectName.IDL

3.7 接口定义语言

3.8 ATL工程的建立

3.8.1 ATL的基本宿主支持：CComModule

3.8.2 BEGIN\_OBJECT\_MAP和OBJECT\_ENTRY宏

- 3.8.3 CComModule
  - 3.8.4 ATL对组件的支持
  - 3.8.5 AddRef和Release所在的位置
  - 3.8.6 CComObject
  - 3.8.7 接口和多重继承
  - 3.8.8 ATL\_NO\_VTABLE
  - 3.8.9 ATL的类工厂支持：CComCoClass
  - 3.8.10 自注册功能与注册器
  - 3.8.11 ATL和组件的类别
  - 3.9 使用ATL实现Math组件
    - 3.9.1 创建Visual C++工程
    - 3.9.2 ATL COM AppWizard
    - 3.9.3 ATL Object Wizard
    - 3.9.4 Object Wizard里的Names（名称项）
    - 3.9.5 Object Wizard里的Attributes（属性项）
  - 3.10 IMath接口的实现
  - 3.11 添加IAdvancedMath及方法
    - 3.11.1 MATH.H文件的改动
    - 3.11.2 MATH.CPP文件的改动
  - 3.12 工程的创建
    - 3.12.1 建立另外一个COM客户程序
    - 3.12.2 测试服务器
  - 3.13 小结
- ## 第四章 接口、接口定义语言和调度
- 4.1 COM接口类型概述
    - 4.1.1 Vtable接口
    - 4.1.2 IDispatch接口：Dispinterfaces
    - 4.1.3 双向接口
  - 4.2 组件及其接口的描述
    - 4.2.1 类型信息
    - 4.2.2 调度
  - 4.3 分布式COM
  - 4.4 标准调度
    - 4.4.1 类型库（通用）调度
    - 4.4.2 自定义调度
  - 4.5 创建Proxy/Stub DLL
  - 4.6 接口定义语言
    - 4.6.1 基本语法和布局
    - 4.6.2 接口的声明：方法和属性
  - 4.7 IDL数据类型
    - 4.7.1 数组
    - 4.7.2 字符串
    - 4.7.3 结构
    - 4.7.4 ENUM类型
  - 4.8 ATL及COM数据类型
    - 4.8.1 接口指针
    - 4.8.2 C++智能指针
    - 4.8.3 CComPtr
    - 4.8.4 CComQIPtr

- 4.8.5 BSTR
- 4.8.6 CComBSTR
- 4.9 COM的内存管理
  - 4.9.1 CoTaskMemAlloc和CoTaskMemFree
  - 4.9.2 IDL和内存管理
- 4.10 COM中的错误处理
  - 4.10.1 ISupportErrorInfo
  - 4.10.2 CreateErrorInfo和ICreateErrorInfo
  - 4.10.3 SetErrorInfo和IErrorInfo
- 4.11 客户机和GetErrorInfo
- 4.12 ATL对错误处理的支持
  - 4.12.1 CComCoClass::Error
  - 4.12.2 ISupportErrorInfoImpl
- 4.13 示例：在一个EXE宿主程序中创建Math组件
  - 4.13.1 步骤一：创建Visual C++工程
  - 4.13.2 步骤二：使用ATL Object Wizard
  - 4.13.3 步骤三：添加IMath和IAdvancedMath接口
  - 4.13.4 步骤四：处理被零除的问题
  - 4.13.5 步骤五：添加IMath2接口
  - 4.13.6 步骤六：添加IComponentInfo接口
  - 4.13.7 步骤七：工程的编译连接
  - 4.13.8 步骤八：编连并注册Proxy/Stub DLL
- 4.14 示例：创建COM客户机程序
- 4.15 示例：创建Visual Basic客户机程序
  - 4.15.1 步骤一：创建应用程序窗体
  - 4.15.2 步骤二：为Math组件插入引用
  - 4.15.3 步骤三：添加代码
- 4.16 小结
- 第五章 包容与集合
  - 5.1 二进制复用
  - 5.2 COM包容
  - 5.3 COM集合
  - 5.4 ATL对集合的支持
  - 5.5 创建集合组件
    - 5.5.1 选择性集合
    - 5.5.2 盲目集合
  - 5.6 ATL对包容的支持
    - 5.6.1 FinalConstruct和FinalRelease
  - 5.7 ATL对集合的支持
    - 5.7.1 可集合组件的实现
    - 5.7.2 DECLARE\_AGGREGATABLE宏
    - 5.7.3 DECLARE\_NOT\_AGGREGATABLE宏
    - 5.7.4 DECLARE\_ONLY\_AGGREGATABLE宏
    - 5.7.5 DECLARE\_POLY\_AGGREGATABLE宏
  - 5.8 集合的实现
    - 5.8.1 COM\_INTERFACE\_ENTRY\_AGGREGATE
    - 5.8.2 COM\_INTERFACE\_ENTRY\_AGGREGATE\_BLIND
    - 5.8.3 COM\_INTERFACE\_ENTRY\_AUTOAGGREGATE宏和COM\_INTERFACE\_ENTRY\_AUTOAGGREGATE\_BLIND宏

## 5.8.4 DECLARE\_GET\_CONTROLLING\_UNKNOWN宏

### 5.9 示例：创建一个简单的Math组件

#### 5.9.1 步骤一：创建实现IMath接口的简单Math组件

#### 5.9.2 步骤二：实现IMath接口

#### 5.9.3 步骤三：工程的编译连接（简单示例）

### 5.10 示例：使用集合实现高级Math组件

#### 5.10.1 步骤一：实现IAdancedMath接口

#### 5.10.2 步骤二：与简单的Math组件集合

#### 5.10.3 步骤三：纳入集合的CLSID和IID定义的头文件

#### 5.10.4 步骤四：向类中添加IUnknown指针

#### 5.10.5 步骤五：覆盖FinalConstruct接口

#### 5.10.6 步骤六：将被集合组件的接口添加到接口映射中

#### 5.10.7 步骤七：释放被集合的组件

#### 5.10.8 步骤八：测试集合组件

### 5.11 示例：通过包容实现高级Math组件

#### 5.11.1 步骤一：为IMath添加实现代码

#### 5.11.2 步骤二：测试包容组件

## 5.12 小结

## 第六章 自动化

### 6.1 后绑定

### 6.2 Dispinterface

### 6.3 双向接口

### 6.4 自动化数据类型

#### 6.4.1 VARIANT数据类型

#### 6.4.2 SafeArray数据类型

### 6.5 实现一个派发接口

#### 6.5.1 实现IDispatch接口

#### 6.5.2 IDispatch函数

### 6.6 双向接口的实现

### 6.7 访问基于IDispatch的接口

#### 6.7.1 后绑定（动态绑定）

#### 6.7.2 ID绑定（前绑定）

### 6.8 前绑定的需求信息

### 6.9 ATL对自动化的支持

#### 6.9.1 IDispatchImpl

#### 6.9.2 CComTypeInfoHolder

#### 6.9.3 CComVariant

### 6.10 自动化与接口版本更新

#### 6.10.1 IDispatch的版本更新

#### 6.10.2 双向接口的版本更新

### 6.11 示例：使用自动化实现Math组件

#### 6.11.1 步骤一：创建Math组件

#### 6.11.2 步骤二：实现IMath接口

#### 6.11.3 步骤三：设置IDL属性

#### 6.11.4 步骤四：编连（Build）所生成的工程

### 6.12 添加第二个IDispatch接口

### 6.13 实现第三个双向接口

### 6.14 创建C++双向接口客户机程序

### 6.15 创建Visual Basic客户机程序

6.15.1 后绑定

6.15.2 前绑定

6.16 小结

## 第七章 事件和连接点

7.1 接口的回调

7.2 引入和引出接口

7.2.1 Advise方法

7.2.2 Visual Basic中的Implements关键字

7.3 回调示例

7.3.1 步骤一：创建Chapter7\_Server工程和Math组件

7.3.2 步骤二：实现IMath接口

7.3.3 步骤三：添加Advise方法

7.3.4 步骤四：定义引出接口

7.3.5 步骤五：通知客户机程序

7.4 Visual Basic客户机程序

7.4.1 步骤一：引用Chapter7\_CallbackServer的类型库

7.4.2 步骤二：添加CallBack类

7.4.3 步骤三：通知并传送组件

7.5 可连接对象

7.5.1 IConnectionPointContainer

7.5.2 IConnectionPoint

7.5.3 连接点、自动化和IProvideClassInfo2

7.5.4 IProvideClassInfo2Impl

7.6 ATL与连接点

7.6.1 Implement Connection Point Wizard向导程序

7.6.2 连接映射

7.6.3 IConnectionPointImpl

7.6.4 IConnectionPointContainerImpl

7.6.5 AtlAdvise

7.6.6 Visual Basic的WithEvents关键字

7.7 使用连接点的一个示例

7.7.1 步骤一：创建Math组件

7.7.2 步骤二：实现IMath接口

7.7.3 步骤三：定义Event接口

7.7.4 步骤四：运行Implement Connection Point Wizard

7.7.5 步骤五：创建一个C++连接点客户程序

7.7.6 步骤六：编写Visual Basic连接点客户机程序

7.8 小结

## 第八章 ActiveX控件

8.1 ActiveX控件和容器

8.1.1 容器模式

8.1.2 控件和容器接口

8.2 什么是ActiveX控件

8.2.1 OLE Controls '94规范

8.2.2 OLE Controls '96规范

8.2.3 控件和容器设计指南

8.3 ActiveX控件功能类属

8.3.1 基本的COM支持

8.3.2 自注册



- 8.4 控件注册表项
  - 8.4.1 Control
  - 8.4.2 Programmable
  - 8.4.3 Insertable
  - 8.4.4 MiscStatus
  - 8.4.5 Verb
  - 8.4.6 ToolboxBitmap32
  - 8.4.7 TypeLib
- 8.5 组件类属
- 8.6 OLE文档接口
- 8.7 自动化支持：IDispatch
  - 8.7.1 标准属性和固有属性
  - 8.7.2 Ambient属性
  - 8.7.3 ATL和Ambient属性
  - 8.7.4 标准控件接口
- 8.8 可连接对象与控件事件
  - 8.8.1 标准事件
  - 8.8.2 属性页
- 8.9 属性的持续性
  - 8.9.1 IPersistStream
  - 8.9.2 IPersistStreamInit
  - 8.9.3 IPersistPropertyBag
- 8.10 ATL对ActiveX控件的支持
  - 8.10.1 ATL的ActiveX控件支持文件
  - 8.10.2 ATL Object Wizard
  - 8.10.3 Names和Attributes选项卡
  - 8.10.4 Miscellaneous选项卡
  - 8.10.5 View Status选项和透明控件
  - 8.10.6 Misc Status选项
  - 8.10.7 Add Control Based On选项
  - 8.10.8 对话框中的Other选项区
  - 8.10.9 Stock Properties选项卡
  - 8.11 ATL Object Wizard创建的文件
    - 8.11.1 CComControl
    - 8.11.2 CComControlBase
    - 8.11.3 CWindowImpl
    - 8.11.4 CStockPropImpl
  - 8.12 初始化固有字体属性
  - 8.13 IOleObjectImpl
  - 8.14 限制控件的大小
  - 8.15 IViewObjectImpl
  - 8.16 CComControl::OnDraw
  - 8.17 IDataObjectImpl
  - 8.18 IOleInPlaceObjectImpl
  - 8.19 IOleInPlaceActiveObjectImpl
  - 8.20 IOleControlImpl
  - 8.Z1 IPersistStreamInitImpl和IPersistStorageImpl
  - 8.22 属性映射
  - 8.23 ATL对属性页的支持

- 8.23.1 CDialogImpl类
- 8.23.2 IpropertyPageImpl类
- 8.24 创建一个示例控件
  - 8.24.1 创建控件的宿主程序
  - 8.24.2 ATL Object Wizard的使用
  - 8.24.3 ATL Object Wizard的Names和Attributes选项卡
  - 8.24.4 ATL Object Wizard的Miscellaneous选项卡
  - 8.24.5 ATL Object Wizard的Stock Properties选项卡
- 8.25 创建一个基本的控件
  - 8.25.1 初始化固有属性
  - 8.25.2 CComControl::OnDraw
  - 8.25.3 实现控件属性的持久性：Property Map
- 8.26 添加固有属性页
- 8.27 添加自定义属性页
- 8.28 向控件中添加事件
- 8.29 使用Connection Point Wizard
- 8.30 小结

## 第九章 COM的枚举器和集合

- 9.1 COM枚举器
  - 9.1.1 内部枚举器接口
  - 9.1.2 IEnumString接口
- 9.2 客户枚举器的访问
- 9.3 ATL的枚举器支持
- 9.4 CComEnum的应用
- 9.5 建立一个简单的枚举器客户应用程序
- 9.6 实现一个简单的枚举器对象
  - 9.6.1 步骤1：创建ATL服务器工程
  - 9.6.2 步骤2：实现枚举器对象
  - 9.6.3 步骤3：访问这一简单的枚举器服务器
- 9.7 COM集合技术
  - 9.7.1 集合对象的实现
  - 9.7.2 ATL对集合的支持
- 9.8 movie集合的实现
  - 9.8.1 步骤1：创建ATL集合服务器工程
  - 9.8.2 步骤2：实现movie组件
  - 9.8.3 步骤3：修改IDL文件
  - 9.8.4 步骤4：实现movie的集合组件
  - 9.8.5 步骤5：定义IMovieCollection接口
  - 9.8.6 步骤6：实现IApplication接口
  - 9.8.7 步骤7：创建工程
- 9.9 创建Visual Basic客户程序
- 9.10 创建C++客户程序
- 9.11 小结

## 第十章 COM的线程管理

- 10.1 COM线程管理概述
  - 10.1.1 单线程单元
  - 10.1.2 多线程单元
  - 10.1.3 单元
  - 10.1.4 进程内组件

## 10.2 单线程单元

### 10.2.1 主STA

### 10.2.2 STA组件的需求（概要）

## 10.3 多线程单元

### 10.3.1 MTA组件的需求（概要）

### 10.3.2 混合模型组件的需求

## 10.4 自由线程管理的调度器

## 10.5 CoMarshalInterThreadInterfaceInStream

## 10.6 代码示例

## 10.7 ATL和COM的线程管理

### 10.7.1 Single和Apartment选项

### 10.7.2 CComSingleThreadModel

### 10.7.3 Free和Both选项

### 10.7.4 Free-Threaded Marshaler选项

### 10.7.5 CComObjectRoot

## 10.8 线程管理的示例

### 10.8.1 步骤1：创建math组件

### 10.8.2 步骤2：IMath接口的实现

### 10.8.3 步骤3：定义事件接口

### 10.8.4 步骤4：实现Advise方法

### 10.8.5 步骤5：在线程内执行计算

### 10.8.6 步骤6：工程的创建

## 10.9 一个C++客户应用程序

## 10.10 小结

## 第十一章 OLE DB和ATL

### 11.1 Microsoft数据访问技术的历史回顾

#### 11.1.1 ODBC

#### 11.1.2 OLEDB

### 11.2 ATL和OLEDB

#### 11.2.1 ATL OLE DB Provider模板

#### 11.2.2 数据源类

#### 11.2.3 对话类

#### 11.2.4 行集合类

#### 11.2.5 命令类

### 11.3 ATL OLE DB Consumer模板

#### 11.3.1 通用数据源支持

#### 11.3.2 行集合和绑定支持

#### 11.3.3 表格和命令支持

#### 11.3.4 用户记录

### 11.4 活动数据对象

## 11.5 示例

### 11.5.1 通用工程概述

### 11.5.2 数据源文件

### 11.5.3 表格文件

### 11.5.4 创建Provider工程

### 11.5.5 模式信息的提供

### 11.5.6 数据管理器的建立

### 11.5.7 通过ADO访问OLE DB Providers

### 11.5.8 MFC/ATL Consumer的例子

## 11.6 小结

## 第十二章 对话框和窗口

### 12.1 ATL的窗口创建类

#### 12.1.1 CWindow

#### 12.1.2 CWindowImpl和消息映射

#### 12.1.3 CWinTraits

#### 12.1.4 CSimpleDialog

#### 12.1.5 CDialogImpl

### 12.2 建立一个简单的Window GUI

#### 12.2.1 步骤1：创建一个简单的窗口

#### 12.2.2 步骤2：创建模态对话框

#### 12.2.3 步骤3：创建非模态对话框

### 12.3 小结

## 第十三章 COM+入门

### 13.1 Window DNA

### 13.2 COM+MTS=COM+

### 13.3 COM+提供的服务

### 13.4 COM+的目录 ( Catalog )

### 13.5 COM+的装载均衡服务

### 13.6 COM+的IMDB服务

### 13.7 COM+的对象缓存服务

### 13.8 COM+的队列组件

### 13.9 COM+事件

### 13.10 COM+的安全性服务

### 13.11 Visual C++未来的变化

#### 13.11.1 基于属性的编程思路

#### 13.11.2 属性语法

#### 13.11.3 编译器的变化

#### 13.11.4 接口描述符

### 13.12 最后一个例子

### 13.13 迈向COM+环境

### 13.14 小结

## 附录 Visual C++自带的COM支持

### A.1 智能指针：\_com\_ptr\_t

### A.2 BSTR (\_bstr\_t)

### A.3 Variant (\_variant\_t)

### A.4 COM的错误处理 (\_com\_error)

### A.5 #import指令

### A.6 TLI和TLH文件

### A.7 应该使用哪一种技术

### A.8 客户应用程序示例

### A.9 Chapter 6\_NativeClient示例

## 精彩短评

1、入门覆盖广不深入有一些用处

## 精彩书评

- 1、这本书从基本C++讲起，不过只是蜻蜓点水，交代给你一个概念，越往后看，基础不好的，或者对ActiveX、COM技术不太了解的同学，疑问会越多，我看到第八章就看不下去了，之前看过《COM技术内幕》，所以对前面几章不费多少力气还可以看懂，到第七章看是就不行了，所以，建议看这本书的同学，先要多COM，ActiveX有一定的了解再下手
- 2、因为对COM已经有一定的了解，所以直接跳过前面的几章，直接看ATL部分。作为入门，如果你想对COM的原理进行一翻探究，《COM技术内幕》和《COM本质论》这两本不错，尤其是《COM技术内幕》很值得COM新手和进阶阶段的朋友反复研读。但ATL方面我觉得这本书不是很给力。首先教材是基于VC++6.0的，这在实际项目开发中已经显得很老旧了。基于这个原因，代码老旧，导致书中罗列的示例代码会跟VC++6.0以上版本的IDE中生成的代码有很大的不同！所以想用VS2008生成的代码与本书进行对照学习几乎是不可能的。其次，专业术语的翻译不是很贴切，导致不少地方看起来觉得怪异和晦涩生硬，例如COM技术中广为谈及的“聚合”在本书中被翻译成了“集合”。后面的还没有看，有待开发.....

## 版权说明

本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问:[www.tushu000.com](http://www.tushu000.com)