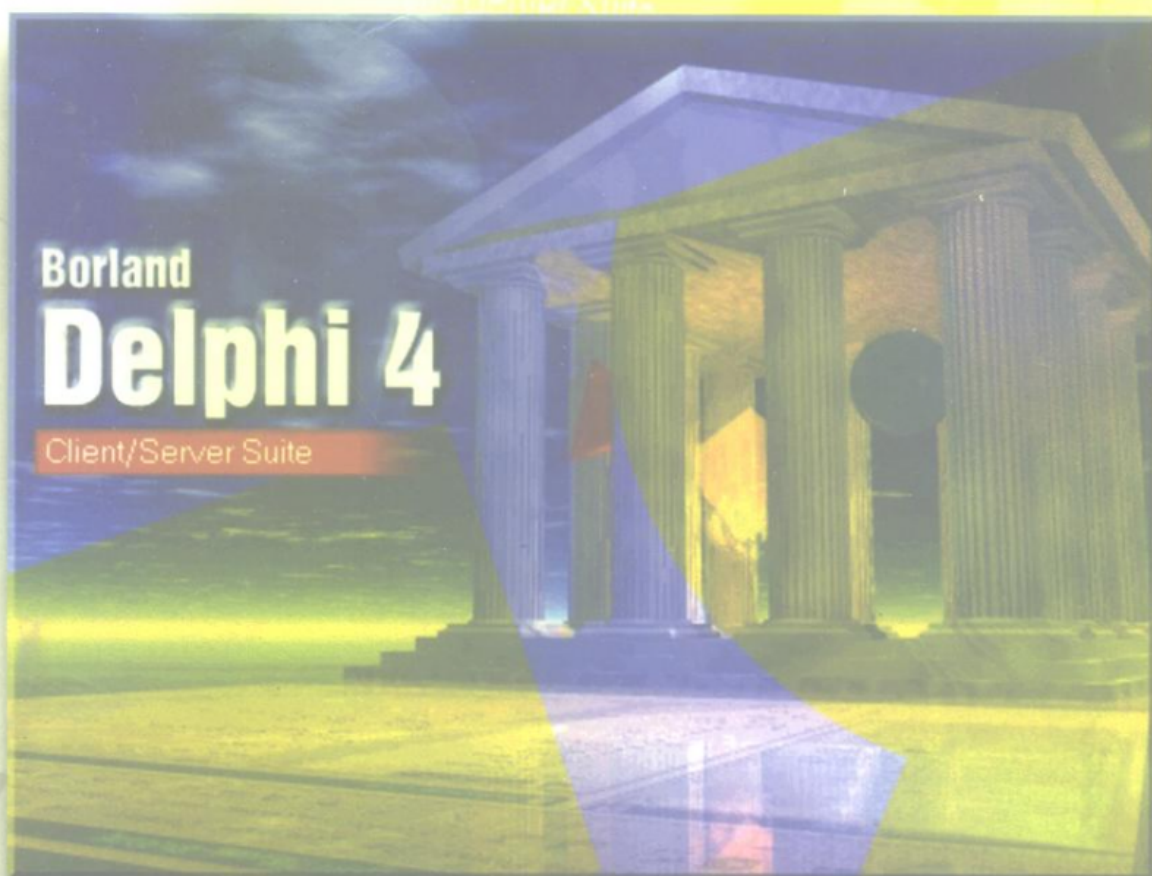


Delphi 4 高级编程丛书之四

COM CORBA 和 Internet 编程技术

徐新华 编著



人民邮电出版社
PEOPLE'S POSTS &
TELECOMMUNICATIONS
PUBLISHING HOUSE

73.963
602

Delphi 4 高级编程丛书之四

COM CORBA 和 Internet 编程技术

徐新华 编著



人民邮电出版社

9910064

JS116/H607

Delphi 4 高级编程丛书之四
COM CORBA 和 Internet 编程技术

◆ 编 著 徐新华

责任编辑 李 际

◆ 人民邮电出版社出版发行 北京市崇文区夕照寺街 14 号

北京顺义向阳胶印厂印刷

新华书店总店北京发行所经销

◆ 开本: 787 × 1092 1/16

印张: 20.25

字数: 499 千字

1998 年 12 月第 1 版

印数: 5 001 - 8 000 册

1999 年 5 月北京第 2 次印刷

ISBN 7-115-07510-7/TP·906

定价: 30.00 元

2000100

本书全面深入地介绍了 Delphi 4 在 COM、CORBA 和 Internet 领域的解决方案。

第 1 章介绍组件对象模型(COM)，包括 COM 的体系结构、GUID、CLSID、IID、引用计数、虚拟方法表、IUnknown 接口、类工厂等概念，然后用 Delphi 4 提供的向导编写了一个 COM 服务器。

第 2 章介绍对象接口，这是 Delphi 4 的一个关键技术。本章详细介绍了对象接口的声明和实现、调度接口、双重接口、获得对接口的引用、查询接口以及怎样用对象接口实现代码重用。

第 3 章介绍 Borland 引以为自豪的 DAX 技术。DAX 实际上是对一组有继承关系的类的通称，DAX 中封装了许多 COM 和 ActiveX 的接口和函数，使开发基于 COM 和 ActiveX 的应用变得非常简单，代码具有良好的重用性。

第 4 章介绍“Type Library”编辑器，这是一个可视化 RAD 工具，可以很方便地编辑 COM 对象和 CORBA 对象的类型库，并自动生成相应的代码和脚本。本章的最后还介绍了 IDL 的文法。

第 5 章详细介绍怎样创建、安装和发布 ActiveX 控件、ActiveForm 以及特性页，着重从代码级来剖析 ActiveX 的奥秘。另外还介绍了数据捆绑功能。

第 6 章包括两部分内容：第一部分介绍怎样操纵自动化服务器，包括引入类型库和 OleVariant 两种方式；第二部分介绍怎样创建、注册和调试自动化服务器。由于 Delphi 4 采用了 DAX 技术，实现了与类型信息的无缝连接，因此，创建自动化服务器是很方便的。本章最后还详细剖析了一个典型的自动化客户程序和一个典型的自动化服务器程

序。

第 7 章介绍 MTS，包括创建 MTS 对象的一般步骤、把 MTS 对象安装到 MTS 包中、MTS Explorer、TmtsAutoObject 以及怎样建立事务对象。

第 8 章介绍 CORBA 的体系结构，包括 Stub、Skeleton 和 Smart Agent 的概念、怎样激活 CORBA 服务器、创建 CORBA 服务器的一般步骤、定义对象接口、在接口库中注册接口、CORBA 客户程序、自定义 CORBA 应用程序、分发 CORBA 应用程序、配置 Smart Agent 等内容。

第 9 章介绍 WinSock 编程技术，包括 Socket 的基本概念、建立 Socket 对象、在网络上传输数据、多线程技术，最后还详细剖析了一个网上交谈(Chat)程序。

第 10~16 章涉及到 Internet/Intranet 上的许多领域如 TCP、UDP、FTP、HTML、HTTP、SMTP、POP3、NNTP，这些都是常用的协议。

第 17~18 章介绍怎样创建 Web 服务器应用程序，包括 Web 模块、动作项、HTTP 请求消息、HTTP 响应消息、HTML 模板、页面生成器、访问数据库等内容。最后还详细剖析了两个典型的 Web 服务器应用程序。

本书内容具有一定的深度，适合那些想在 Internet/Intranet 领域编程的读者阅读。

Delphi 4 是 Borland 更名为 Inprise 后推出的一个具有战略意义的产品，它标志着 Inprise 的工作重心已经从桌面转移到跨平台的分布式应用。Delphi 4 中的 MIDAS 技术奠定了它在业界的领先地位。

为了帮助广大用户全面、准确地掌握 Delphi 4 的编程思想和用法，我们编写了这套《Delphi 4 高级编程丛书》，主要是针对那些已初步掌握了 Delphi 4 的基本用法现在需要进一步精通和提高的读者。本丛书紧紧把握住 Delphi 4 的基本特征即面向对象，重点从类这一层次把 Delphi 4 的编程思想讲透。我们的体会是，只要深刻领会了面向对象的编程思想，就很容易理解那些看上去高深莫测的领域，如 COM、ActiveX、CORBA、MIDAS。

本套丛书分为四册，第一册是《IDE 和 Object Pascal 语言》，第二册是《GUI 编程技术》，第三册是《Database 和 MIDAS 编程技术》，第四册是《COM CORBA 和 Internet 编程技术》。

本书是此套丛书的第四册，全面深入地介绍了当今最热门的编程技术，包括 COM 的体系结构、对象接口、DAX 技术、“Type Library”编辑器、ActiveX 控件、自动化服务器、MTS、CORBA、Winsock 以及 TCP、UDP、HTTP、HTML、FTP、SMTP、POP3、NNTP 等协议，最后还详细介绍了怎样创建 Web 服务器应用程序。要说明的是，本书虽然是针对 Delphi 4 写的，但其中很多内容具有普遍性，对使用其他开发工具的读者也有一定的参考作用。

本书由徐新华执笔，参加编写的人员有郭平、周学成、徐存聪等。

由于我们的水平有限，再加上时间很紧，尽管我们作了严格的审核和测试，书中可能还是难免有一些错误，敬请广大读者不吝赐教，我们谨在此表示感谢。

考虑到 Delphi 4 中增加了许多崭新的技术，有些技术具有相当的难度，为了帮助广大程序员更好地掌握这个优秀的开发工具，我们愿意为购买此书的读者提供技术咨询。我们将热情、及时地答复大家提出的问题。

电子函件地址: p_inprise@mail.263.net.cn

作者

1998 年 9 月

目 录

第一章

组件对象模型 (COM)

1.1 几个基本概念	1
1.2 客户与服务器	2
1.3 认识 GUID、CLSID、IID	3
1.4 引用计数	4
1.5 虚拟方法表	4
1.6 IUnknown 接口	5
1.7 创建 COM 对象的一般步骤	6

第二章

对象接口

2.1 接口的语法	11
2.1.1 怎样声明接口	11
2.1.2 祖先接口	13
2.1.3 接口标识符	14
2.1.4 前向声明	14
2.1.5 怎样实现对象接口	15
2.1.6 怎样实现对象接口中的方法	16
2.1.7 方法解析子句	16
2.1.8 使用方法解析子句的限制	17
2.1.9 委托一个特性来实现接口	19
2.2 调度接口	20
2.3 双重接口	22
2.4 对接口的引用	24
2.4.1 用赋值语句获得接口引用	24
2.4.2 赋值相容和类型强制转换	26
2.4.3 对派生接口的引用	26
2.4.4 接口查询	27
2.4.5 IInterfacedObject	28
2.5 接口与代码重用	29

第三章

DAX 技术

3.1	什么是 DAX	31
3.2	TInterfacedobject	32
3.3	TComobject	33
3.4	TTypedcomobject	34
3.5	TAutoObject	35
3.6	TAutointfObject	36
3.7	TActivexControl	36
3.8	TComserverObject	40
3.9	TComServer	41
3.10	TActiveForm	44
3.11	TPropertytPage	44
3.12	TComObjectFactory	45
3.13	TTypedComObjectFactory	49
3.14	TActiveXPropertyPageFactory	50
3.15	TAutoObjectFactory	50
3.16	TActiveXcontrolFactory	51
3.17	TActiveFormFactory	53

第四章

“Type Library”

编辑器

4.1	关于类型库的概述	55
4.2	“Type Library”编辑器的基本操作	56
4.2.1	创建一个新的类型库	56
4.2.2	打开一个已有的类型库	58
4.2.3	刷新类型库	58
4.2.4	保存和注册类型库	58
4.2.5	把类型库引入到当前项目中	58
4.2.6	引出类型库	59
4.2.7	发布类型库	60
4.3	“Type Library”编辑器的窗口	60
4.4	类型库的一般信息	61
4.5	接 口	64
4.6	在接口中加入成员	65
4.7	调度接口	67
4.8	类型库枚举	68
4.9	组件类	69
4.10	别名、记录、联合、模块	70
4.11	IDL 文法	71

4.11.1	属性表述	71
4.11.2	接口的文法	72
4.11.3	调度接口的文法	72
4.11.4	组件类的文法	73
4.11.5	枚举的文法	73
4.11.6	别名的文法	74
4.11.7	记录的文法	74
4.11.8	联合的文法	75
4.11.9	模块的文法	75

第五章

创建 ActiveX 控件

5.1	创建 ActiveX 控件	77
5.1.1	怎样转换基于 TGraphicControl 的控件	77
5.1.2	ActiveX 控件向导	78
5.2	向导创建了哪些文件	80
5.2.1	项目文件	81
5.2.2	类型库的接口源文件	81
5.2.3	类型库接口的实现文件	89
5.2.4	About 框的源文件	92
5.3	编辑类型库	93
5.3.1	在接口中加入新的成员	93
5.3.2	加入新的特性	94
5.3.3	加入新的方法	95
5.3.4	加入新的事件	95
5.4	数据捆绑	97
5.5	创建特性页	99
5.6	注册和安装 ActiveX 控件	101
5.7	怎样使用 ActiveX 控件	104
5.8	Activeform	104
5.9	在 Web 上发布 ActiveX	106
5.9.1	Project 页	106
5.9.2	Packages 页	107
5.9.3	Additional Files 页	108
5.9.4	Code Signing 页	108

第六章

OLE 自动化

6.1	引入自动化服务器的类型库	111
-----	--------------------	-----

6.2	OleVariant 类型	114
6.2.1	OleVariant 是怎样声明的	114
6.2.2	给 OleVariant 类型的变量赋值	114
6.2.3	可变数组	115
6.3	用 OleVariant 操纵自动化对象	117
6.3.1	创建自动化对象的实例	117
6.3.2	访问失败	118
6.3.3	怎样访问自动化对象的特性	119
6.3.4	怎样访问自动化对象的方法	119
6.4	创建自动化服务器	120
6.5	自动化对象的类型库	121
6.5.1	类型库的接口描述文件	121
6.5.2	类型库接口的实现文件	123
6.5.3	编辑类型库	125
6.5.4	加入新的特性	125
6.5.5	加入新的方法	126
6.5.6	加入新的事件	127
6.6	注册和调试自动化服务器	128
6.7	一个典型的自动化客户程序	128
6.8	一个典型的自动化服务器	131

第七章

创建 MTS 对象

7.1	创建 MTS 对象的一般步骤	139
7.2	向导生成了哪些文件	141
7.2.1	类型库的接口源文件	141
7.2.2	接口的实现单元	142
7.2.3	MTS 对象的类型库	143
7.3	把 MTS 对象安装到 MTS 包中	144
7.4	MTS Explorer	145
7.5	TMtsautoObject	146
7.6	建立事务对象	147

第八章

CORBA 应用程序

8.1	CORBA 应用程序的体系结构	149
8.2	进一步理解 Stub、Skeleton 和 Smart Agent	150
8.3	激活 CORBA 服务器	151
8.4	创建 CORBA 服务器的一般步骤	151

8.5	定义对象接口	152
8.6	自动生成的代码	154
8.7	在接口库中注册接口	158
8.8	CORBA 客户程序	158
8.8.1	使用 Stub	159
8.8.2	使用 DII	159
8.9	自定义 CORBA 应用程序	160
8.9.1	在客户程序中显示 CORBA 对象的名称 ..	161
8.9.2	暴露或隐藏 CORBA 对象	161
8.9.3	传递客户信息给服务器	162
8.10	分发 CORBA 应用程序	162
8.11	配置 Smart Agent	162
8.11.1	启动 Smart Agent	163
8.11.2	配置 ORB 域	163
8.11.3	连接不同局域网上的 Smart Agent	163

第九章

使用 WINSOCK

9.1	关于 Socket 的概述	165
9.2	建立服务器端 Socket	166
9.3	建立客户端 Socket	167
9.4	怎样在网络上传输数据	168
9.5	在客户端使用多线程技术	169
9.6	在服务器端使用多线程技术	170
9.7	TCustomWinSocket	171
9.8	TClientWinSocket	176
9.9	TServerWinSocket	177
9.10	TServerClientWinSocket	181
9.11	TWinSocketStream	182
9.12	一个网上交谈(Chat)程序	183

第十章

使用 FTP 控件

10.1	TNMFTP 元件的特性	189
10.2	TNMFTP 元件的方法	194
10.3	TNMFTP 元件的事件	197

第十一章

使用 UDP 控件

11.1 使用 UDP 控件的一般步骤	201
11.2 UDP 控件的特性	202
11.3 UDP 控件的方法	203
11.4 UDP 控件的事件	204

第十二章

使用 HTTP 客户控件

12.1 HTTP 控件的特性	207
12.2 HTTP 控件的方法	212
12.3 HTTP 控件的事件	214

第十三章

使用 HTML 控件

13.1 HTML 控件概述	217
13.2 HTML 控件的特性	218
13.3 HTML 控件的方法	223
13.4 HTML 控件的事件	225

第十四章

使用 SMTP 控件

14.1 SMTP 控件的特性	229
14.2 SMTP 控件的方法	232
14.3 SMTP 控件的事件	233

第十五章

使用 POP 控件

15.1 POP 控件的特性	237
15.2 POP 控件的方法	241
15.3 POP 控件的事件	242

第十六章

使用 NNTP 控件

16.1 NNTP 控件的特性	245
16.2 NNTP 控件的方法	251
16.3 NNTP 控件的事件	252

第十七章

创建 Web 服务器 应用程序

17.1	WWW 是怎样工作的	257
17.2	静态的 HTML 页面	258
17.3	动态的 HTML 页面	262
17.4	怎样与客户交互	263
17.5	交互生成页面	265
17.6	与数据库的连接	267
17.7	怎样调试 Web 服务器应用程序	269
17.7.1	调试 ISAPI 或 NSAPI 类型的 Web 服务器 应用程序	269
17.7.2	调试 CGI 或 Win - CGI 类型的 Web 服务 器应用程序	271

第十八章

Web 服务器 应用程序的细节

18.1	Web 服务器应用程序的逻辑结构	273
18.2	Web 模块	274
18.3	Web 调度器	276
18.4	动作项	277
18.5	HTTP 请求消息	280
18.5.1	HTTP 请求消息是怎样传递的	280
18.5.2	TWebRequest 是怎样声明的	280
18.5.3	TWebRequest 对象的特性和方法	282
18.5.4	TISAPIRequest 对象	287
18.5.5	TCGIRequest 对象	288
18.5.6	TWinCGIRequest 对象	289
18.6	HTTP 响应消息	289
18.6.1	怎样建立 HTTP 响应消息	289
18.6.2	怎样传递 HTTP 响应消息	290
18.6.3	自己传递 HTTP 响应消息	290
18.6.4	TWebResponse 是怎样声明的	290
18.6.5	TWebResponse 对象的特性和方法	291
18.6.6	TISAPIResponse 对象	295
18.6.7	TCGIResponse 对象	296
18.6.8	TWinCGIResponse 对象	296
18.7	页面生成器	297
18.7.1	怎样建立 HTML 模板	297
18.7.2	指定 HTML 模板	298
18.7.3	转换并返回转换后的结果	299

18.7.4	怎样转换(OnHTMLTag 事件)	299
18.7.5	TDataSetPageProducer	300
18.8	操纵 Web 服务器应用程序	300
18.8.1	TWebApplication 对象	301
18.8.2	TCGIApplication 对象	302
18.8.3	TISAPIApplication 对象	303
18.9	Web 服务器与数据库	303
18.9.1	用 TSession 管理与数据库的连接	303
18.9.2	TDataSetTableProducer 元件	304
18.9.3	TQueryTableProducer 元件	307
18.10	两个典型的 Web 服务器应用程序	307

第一章

组件对象模型 (COM)

组件对象模型(Component Object Model, 简称 COM)是组件对象之间相互接口的规范。凡是遵循 COM 接口规范的对象彼此之间能相互通信和交互,即使这些对象由不同的厂商、不同的语言、不同的 Windows 版本甚至在不同的机器上编写和建立。

Delphi 4 支持 COM 接口规范, Object Pascal 语言增加了对象接口的语法。用 Delphi 4 创建的 COM 对象还可以工作在 MTS(Microsoft Transaction Server)环境中。本章详细介绍组件对象模型的体系结构,并用 Delphi 4 实际创建一个 COM 服务器。

1.1 几个基本概念

软件重用是业界追求的目标,人们一直希望能够像搭积木一样随意“装配”应用程序,组件对象就充当了积木的角色。所谓组件对象,实际上就是预定义好的、能完成一定功能的服务或接口。问题是,这些组件对象如何与应用程序、如何与其他组件对象共存并相互通信和交互,这就需要制定一个规范,让这些组件对象按统一的标准方式工作。

COM 是个二进制规范,它与源代码无关。这样,即使 COM 对象由不同的编程语言创建,运行在不同的进程空间和不同的操作系统平台,这些对象也能相互通信。COM 既是规范,也是实现,它以 COM 库(OLE 32.dll 和 OLEAut32.dll)的形式提供了访问 COM 对象核心功能的标准接口以及一组 API

函数。这些 API 用于实现创建和管理 COM 对象的功能。

COM 本质上仍然是客户/服务器模式。客户(通常是应用程序)请求创建 COM 对象并通过 COM 对象的接口操纵 COM 对象。服务器根据客户的请求创建并管理 COM 对象。客户和服务器这两种角色并不是绝对的。

组件对象与一般意义上的对象既相似也有区别。一般意义上的对象是一种把数据和操纵数据的方法封装在一起的数据类型的实例,而组件对象使用接口(Interface)而不是方法来描述自己并提供服务。所谓接口,其精确定义是“基于对象的一组语义上相关的功能”。实际上是一个纯虚类,真正实现接口的是接口对象(Interface Object)。一个 COM 对象可以只有一个或几个接口:例如,Windows 95 外壳扩展,也可以有许多接口;ActiveX 控件一般就有多个接口,客户可以从很多方面来操纵 ActiveX 控件。

接口是客户与服务器通信的唯一途径。客户只能通过接口访问接口对象中的方法,而不能访问该对象中的数据。如果一个组件对象有多个接口,通过一个接口不能直接访问其他接口。但是,COM 模型提供了这样一种策略,就是允许客户调用 COM 库中的 QueryInterface 函数去查询组件对象所支持的其他接口。从这个意义上讲,组件对象有点像接口对象的经纪人(Broker)。当客户调用 QueryInterface 函数后,如果组件对象正好支持要查询的接口,函数就返回该接口的指针。如果组件对象不支持该接口,函数就返回一个出错信息。所以,QueryInterface 函数是很有用的,它可以动态了解组件对象所支持的接口。

接口实际上是面向对象编程思想的一种体现,它隐藏了 COM 对象实现服务的细节。COM 对象可以完全独立于访问它的客户,只要接口本身保持不变。如果实在需要更新接口,可以重新定义一个新的接口,对于使用老接口的客户来说,代码得到了最大程度的保护。

1.2 客户与服务器

COM 本质上仍然是客户/服务器模式。COM 服务器实际上是组件对象的容器。COM 服务器中的组件对象向 COM 客户提供服务。COM 客户通常是 EXE,也可能是 DLL,甚至就是 Windows 自己。COM 客户一般应独立于 COM 服务器,因为 COM 客户并不知道 COM 服务器在哪儿、怎样把它唤醒,甚至有没有这样的 COM 服务器都不知道。

当一个客户请求某个 COM 对象的服务时,客户需要传递一个类的标识符(CLSID),请求 Windows 去查找组件对象在哪儿,找到以后把接口的指针传递给客户。Windows 将从注册表中查找 COM 服务器的位置并定位一个合适的 COM 对象。

根据 COM 服务器与 COM 客户是否运行在同一个进程地址空间,COM 服务器分为三类:In-Process 服务器、Out-of-Process 服务器和 Remote 服务器。

In-Process 服务器通常是 DLL,它映射到客户的进程地址空间中运行。例如,一个嵌入在 Web 网页中的 ActiveX 控件,与 Internet Explorer 在同一个进程地址空间中运行。对于 In-Process 服务器来说,客户可以直接调用 COM 对象的接口。

Out-of-Process 或 Local 服务器通常是 EXE,它与 COM 客户在同一个机器上,但在不同的进程地址空间运行。例如,一个 Excel 电子表格嵌入到 Word 文档中。