

## [Preface](#)

[Prerequisites](#)

[Scope of this Book](#)

[Acknowledgments](#)

[Michi's Acknowledgments](#)

[Steve's Acknowledgments](#)

## [1. Introduction](#)

[1.1 Introduction](#)

[1.2 Organization of the Book](#)

[1.3 CORBA Version](#)

[1.4 Typographical Conventions](#)

[1.5 Source Code Examples](#)

[1.6 Vendor Dependencies](#)

[1.7 Contacting the Authors](#)

## [I: Introduction to CORBA](#)

### [2. An Overview of CORBA](#)

[2.1 Introduction](#)

[2.2 The Object Management Group](#)

[2.3 Concepts and Terminology](#)

[2.4 CORBA Features](#)

[2.5 Request Invocation](#)

[2.6 General CORBA Application Development](#)

[2.7 Summary](#)

### [3. A Minimal CORBA Application](#)

[3.1 Chapter Overview](#)

[3.2 Writing and Compiling an IDL Definition](#)

[3.3 Writing and Compiling a Server](#)

[3.4 Writing and Compiling a Client](#)

[3.5 Running Client and Server](#)

[3.6 Summary](#)

## [II: Core CORBA](#)

### [4. The OMG Interface Definition Language](#)

[4.1 Chapter Overview](#)

[4.2 Introduction](#)

[4.3 Compilation](#)

[4.4 Source Files](#)

[4.5 Lexical Rules](#)

[4.6 Basic IDL Types](#)

[4.7 User-Defined Types](#)

[4.8 Interfaces and Operations](#)

[4.9 User Exceptions](#)

[4.10 System Exceptions](#)

[4.11 System Exceptions or User Exceptions?](#)

[4.12 Oneway Operations](#)

[4.13 Contexts](#)

[4.14 Attributes](#)

- [4.15 Modules](#)
- [4.16 Forward Declarations](#)
- [4.17 Inheritance](#)
- [4.18 Names and Scoping](#)
- [4.19 Repository Identifiers and pragma Directives](#)
- [4.20 Standard Include Files](#)
- [4.21 Recent IDL Extensions](#)
- [4.22 Summary](#)

## [5. IDL for a Climate Control System](#)

- [5.1 Chapter Overview](#)
- [5.2 The Climate Control System](#)
- [5.3 IDL for the Climate Control System](#)
- [5.4 The Complete Specification](#)

## [6. Basic IDL-to-C++ Mapping](#)

- [6.1 Chapter Overview](#)
- [6.2 Introduction](#)
- [6.3 Mapping for Identifiers](#)
- [6.4 Mapping for Modules](#)
- [6.5 The CORBA Module](#)
- [6.6 Mapping for Basic Types](#)
- [6.7 Mapping for Constants](#)
- [6.8 Mapping for Enumerated Types](#)
- [6.9 Variable-Length Types and `\_var` Types](#)
- [6.10 The String `\_var` Wrapper Class](#)
- [6.11 Mapping for Wide Strings](#)
- [6.12 Mapping for Fixed-Point Types](#)
- [6.14 Mapping for Sequences](#)
- [6.15 Mapping for Arrays](#)
- [6.16 Mapping for Unions](#)
- [6.17 Mapping for Recursive Structures and Unions](#)
- [6.18 Mapping for Type Definitions](#)
- [6.19 User-Defined Types and `\_var` Classes](#)
- [6.20 Summary](#)

## [7. Client-Side C++ Mapping](#)

- [7.1 Chapter Overview](#)
- [7.2 Introduction](#)
- [7.3 Mapping for Interfaces](#)
- [7.4 Object Reference Types](#)
- [7.5 Life Cycle of Object References](#)
- [7.6 Semantics of `\_ptr` References](#)
- [7.7 Pseudo-Objects](#)
- [7.8 ORB Initialization](#)
- [7.9 Initial References](#)
- [7.10 Stringified References](#)
- [7.11 The Object Pseudo-Interface](#)
- [7.12 `\_var` References](#)
- [7.13 Mapping for Operations and Attributes](#)
- [7.14 Parameter Passing Rules](#)
- [7.15 Mapping for Exceptions](#)
- [7.16 Mapping for Contexts](#)

[7.17 Summary](#)

[8. Developing a Client for the Climate Control System](#)

[8.1 Chapter Overview](#)

[8.2 Introduction](#)

[8.3 Overall Client Structure](#)

[8.4 Included Files](#)

[8.5 Helper Functions](#)

[8.6 The main Program](#)

[8.7 The Complete Client Code](#)

[8.8 Summary](#)

[9. Server-Side C++ Mapping](#)

[9.1 Chapter Overview](#)

[9.2 Introduction](#)

[9.3 Mapping for Interfaces](#)

[9.4 Servant Classes](#)

[9.5 Object Incarnation](#)

[9.6 Server main](#)

[9.7 Parameter Passing Rules](#)

[9.8 Raising Exceptions](#)

[9.9 Tie Classes](#)

[9.10 Summary](#)

[10. Developing a Server for the Climate Control System](#)

[10.1 Chapter Overview](#)

[10.2 Introduction](#)

[10.3 The Instrument Control Protocol API](#)

[10.4 Designing the Thermometer Servant Class](#)

[10.5 Implementing the Thermometer Servant Class](#)

[10.6 Designing the Thermostat Servant Class](#)

[10.7 Implementing the Thermostat Servant Class](#)

[10.8 Designing the Controller Servant Class](#)

[10.9 Implementing the Controller Servant Class](#)

[10.10 Implementing the Server main Function](#)

[10.11 The Complete Server Code](#)

[10.12 Summary](#)

[11. The Portable Object Adapter](#)

[11.1 Chapter Overview](#)

[11.2 Introduction](#)

[11.3 POA Fundamentals](#)

[11.4 POA Policies](#)

[11.5 POA Creation](#)

[11.6 Servant IDL Type](#)

[11.7 Object Creation and Activation](#)

[11.8 Reference, ObjectId , and Servant](#)

[11.9 Object Deactivation](#)

[11.10 Request Flow Control](#)

[11.11 ORB Event Handling](#)

[11.12 POA Activation](#)

[11.13 POA Destruction](#)

[11.14 Applying POA Policies](#)

[11.15 Summary](#)

[12. Object Life Cycle](#)

[12.1 Chapter Overview](#)

[12.2 Introduction](#)

[12.3 Object Factories](#)

[12.4 Destroying, Copying, and Moving Objects](#)

[12.5 A Critique of the Life Cycle Service](#)

[12.6 The Evictor Pattern](#)

[12.7 Garbage Collection of Servants](#)

[12.8 Garbage Collection of CORBA Objects](#)

[12.9 Summary](#)

[III: CORBA Mechanisms](#)

[13. GIOP, IIOP, and IORs](#)

[13.1 Chapter Overview](#)

[13.2 An Overview of GIOP](#)

[13.3 Common Data Representation](#)

[13.4 GIOP Message Formats](#)

[13.5 GIOP Connection Management](#)

[13.6 Detecting Disorderly Shutdown](#)

[13.7 An Overview of IIOP](#)

[13.8 Structure of an IOR](#)

[13.9 Bidirectional IIOP](#)

[13.10 Summary](#)

[14. Implementation Repositories and Binding](#)

[14.1 Chapter Overview](#)

[14.2 Binding Modes](#)

[14.3 Direct Binding](#)

[14.4 Indirect Binding via an Implementation Repository](#)

[14.5 Migration, Reliability, Performance, and Scalability](#)

[14.6 Activation Modes](#)

[14.7 Race Conditions](#)

[14.8 Security Considerations](#)

[14.9 Summary](#)

[VI: Dynamic CORBA](#)

[15. C++ Mapping for Type any](#)

[15.1 Chapter Overview](#)

[15.2 Introduction](#)

[15.3 Type any C++ Mapping](#)

[15.4 Pitfalls in Type Definitions](#)

[15.5 Summary](#)

[16. Type Codes](#)

[16.1 Chapter Overview](#)

[16.2 Introduction](#)

[16.3 The TypeCode Pseudo-Object](#)

[16.4 C++ Mapping for the TypeCode Pseudo-Object](#)

[16.5 Type Code Comparisons](#)

- [16.6 Type Code Constants](#)
- [16.7 Type Code Comparison for Type any](#)
- [16.8 Creating Type Codes Dynamically](#)
- [16.9 Summary](#)

## [17. Type DynAny](#)

- [17.1 Chapter Overview](#)
- [17.2 Introduction](#)
- [17.3 The DynAny Interface](#)
- [17.4 C++ Mapping for DynAny](#)
- [17.5 Using DynAny for Generic Display](#)
- [17.6 Obtaining Type Information](#)
- [17.7 Summary](#)

## [V: CORBAservices](#)

### [18. The OMG Naming Service](#)

- [18.1 Chapter Overview](#)
- [18.2 Introduction](#)
- [18.3 Basic Concepts](#)
- [18.4 Structure of the Naming Service IDL](#)
- [18.5 Semantics of Names](#)
- [18.6 Naming Context IDL](#)
- [18.7 Iterators](#)
- [18.8 Pitfalls in the Naming Service](#)
- [18.9 The Names Library](#)
- [18.10 Naming Service Tools](#)
- [18.11 What to Advertise](#)
- [18.12 When to Advertise](#)
- [18.13 Federated Naming](#)
- [18.14 Adding Naming to the Climate Control System](#)
- [18.15 Summary](#)

### [19. The OMG Trading Service](#)

- [19.1 Chapter Overview](#)
- [19.2 Introduction](#)
- [19.3 Trading Concepts and Terminology](#)
- [19.4 IDL Overview](#)
- [19.5 The Service Type Repository](#)
- [19.6 The Trader Interfaces](#)
- [19.7 Exporting Service Offers](#)
- [19.8 Withdrawing Service Offers](#)
- [19.9 Modifying Service Offers](#)
- [19.10 The Trader Constraint Language](#)
- [19.11 Importing Service Offers](#)
- [19.12 Bulk Withdrawal](#)
- [19.13 The Admin Interface](#)
- [19.14 Inspecting Service Offers](#)
- [19.15 Exporting Dynamic Properties](#)
- [19.16 Trader Federation](#)
- [19.17 Trader Tools](#)
- [19.18 Architectural Considerations](#)
- [19.19 What to Advertise](#)

- [19.20 Avoiding Duplicate Service Offers](#)
- [19.21 Adding Trading to the Climate Control System](#)
- [19.22 Summary](#)

## [20. The OMG Event Service](#)

- [20.1 Chapter Overview](#)
- [20.2 Introduction](#)
- [20.3 Distributed Callbacks](#)
- [20.4 Event Service Basics](#)
- [20.5 Event Service Interfaces](#)
- [20.6 Implementing Consumers and Suppliers](#)
- [20.7 Choosing an Event Model](#)
- [20.8 Event Service Limitations](#)
- [20.9 Summary](#)

## [VI: Power CORBA](#)

### [21. Multithreaded Applications](#)

- [21.1 Chapter Overview](#)
- [21.2 Introduction](#)
- [21.3 Motivation for Multithreaded Programs](#)
- [21.4 Fundamentals of Multithreaded Servers](#)
- [21.5 Multithreading Strategies](#)
- [21.6 Implementing a Multithreaded Server](#)
- [21.7 Servant Activators and the Evictor Pattern](#)
- [21.8 Summary](#)

### [22. Performance, Scalability, and Maintainability](#)

- [22.1 Chapter Overview](#)
- [22.2 Introduction](#)
- [22.3 Reducing Messaging Overhead](#)
- [22.4 Optimizing Server Implementations](#)
- [22.5 Federating Services](#)
- [22.6 Improving Physical Design](#)
- [22.7 Summary](#)

### [A. Source Code for the ICP Simulator](#)

- [A.1 Overview](#)
- [A.2 Transient Simulator Code](#)
- [A.3 Persistent Simulator Code](#)

### [B. CORBA Resources](#)

- [B.1 World Wide Web](#)
- [B.2 Newsgroups](#)
- [B.3 Mailing Lists](#)
- [B.4 Magazines](#)

### [Bibliography](#)