

## *Preface*

---

### *0.1 About This Document*

Under the terms of the collaboration between OMG and X/Open Co Ltd., this document is a candidate for endorsement by X/Open, initially as a Preliminary Specification and later as a full CAE Specification. The collaboration between OMG and X/Open Co Ltd. ensures joint review and cohesive support for emerging object-based specifications.

X/Open Preliminary Specifications undergo close scrutiny through a review process at X/Open before publication and are inherently stable specifications. Upgrade to full CAE Specification, after a reasonable interval, takes place following further review by X/Open. This further review considers the implementation experience of members and the full implications of conformance and branding.

#### *0.1.1 Object Management Group*

The Object Management Group, Inc. (OMG) is an international organization supported by over 800 members, including information system vendors, software developers and users. Founded in 1989, the OMG promotes the theory and practice of object-oriented technology in software development. The organization's charter includes the establishment of industry guidelines and object management specifications to provide a common framework for application development. Primary goals are the reusability, portability, and interoperability of object-based software in distributed, heterogeneous environments. Conformance to these specifications will make it possible to develop a heterogeneous applications environment across all major hardware platforms and operating systems.

OMG's objectives are to foster the growth of object technology and influence its direction by establishing the Object Management Architecture (OMA). The OMA provides the conceptual infrastructure upon which all OMG specifications are based.

---

### 0.1.2 X/Open

X/Open is an independent, worldwide, open systems organization supported by most of the world's largest information system suppliers, user organizations and software companies. Its mission is to bring to users greater value from computing, through the practical implementation of open systems. X/Open's strategy for achieving its mission is to combine existing and emerging standards into a comprehensive, integrated systems environment called the Common Applications Environment (CAE).

The components of the CAE are defined in X/Open CAE specifications. These contain, among other things, an evolving portfolio of practical application programming interfaces (APIs), which significantly enhance portability of application programs at the source code level. The APIs also enhance the interoperability of applications by providing definitions of, and references to, protocols and protocol profiles.

The X/Open specifications are also supported by an extensive set of conformance tests and by the X/Open trademark (XPG brand), which is licensed by X/Open and is carried only on products that comply with the CAE specifications.

## 0.2 Intended Audience

The architecture and specifications described in this manual are aimed at software designers and developers who want to produce applications that comply with OMG standards for the Object Request Broker (ORB). The benefit of compliance is, in general, to be able to produce interoperable applications that are based on distributed, interoperating objects. As defined by the Object Management Group (OMG) in the *Object Management Architecture Guide*, the ORB provides the mechanisms by which objects transparently make requests and receive responses. Hence, the ORB provides interoperability between applications on different machines in heterogeneous distributed environments and seamlessly interconnects multiple object systems.

## 0.3 Context of CORBA

The key to understanding the structure of the CORBA architecture is the Reference Model, which consists of the following components:

- **Object Request Broker**, which enables objects to transparently make and receive requests and responses in a distributed environment. It is the foundation for building applications from distributed objects and for interoperability between applications in hetero- and homogeneous environments. The architecture and specifications of the Object Request Broker are described in this manual.
- **Object Services**, a collection of services (interfaces and objects) that support basic functions for using and implementing objects. Services are necessary to construct any distributed application and are always independent of application domains. For example, the Life Cycle Service defines conventions for creating, deleting, copying, and moving objects; it does not dictate how the objects are implemented in an application. Specifications for Object Services are contained in *CORBAServices: Common Object Services Specification*.

- 
- **Common Facilities**, a collection of services that many applications may share, but which are not as fundamental as the Object Services. For instance, a system management or electronic mail facility could be classified as a common facility. Information about Common Facilities will be contained in *CORBAfacilities: Common Facilities Architecture*.
  - **Application Objects**, which are products of a single vendor or in-house development group which controls their interfaces. Application Objects correspond to the traditional notion of applications, so they are not standardized by OMG. Instead, Application Objects constitute the uppermost layer of the Reference Model.

The Object Request Broker, then, is the core of the Reference Model. It is like a telephone exchange, providing the basic mechanism for making and receiving calls. Combined with the Object Services, it ensures meaningful communication between CORBA-compliant applications.

## 0.4 Associated Documents

The CORBA documentation set includes the following books:

- *Object Management Architecture Guide* defines the OMG's technical objectives and terminology and describes the conceptual models upon which OMG standards are based. It also provides information about the policies and procedures of OMG, such as how standards are proposed, evaluated, and accepted.
- *CORBA: Common Object Request Broker Architecture and Specification* contains the architecture and specifications for the Object Request Broker.
- *CORBAservices: Common Object Services Specification* contains specifications for the Object Services.
- *CORBAfacilities: Common Facilities Architecture* contains the architecture for Common Facilities.

OMG collects information for each book in the documentation set by issuing Requests for Information, Requests for Proposals, and Requests for Comment and, with its membership, evaluating the responses. Specifications are adopted as standards only when representatives of the OMG membership accept them as such by vote.

To obtain books in the documentation set, or other OMG publications, refer to the enclosed subscription card or contact the Object Management Group, Inc. at:

OMG Headquarters  
492 Old Connecticut Path  
Framingham, MA 01701  
USA  
Tel: +1-508-820 4300  
Fax: +1-508-820 4303  
pubs@omg.org  
<http://www.omg.org>

---

## 0.5 Definition of CORBA Compliance

As described in the *OMA Guide*, the OMG's Core Object Model consists of a core and components. Likewise, the body of *CORBA* specifications is divided into core and component-like specifications. The structure of this manual reflects that division.

The *CORBA* specifications are categorized as follows:

**CORBA Core**, as specified in Chapters 1-9

**CORBA Interoperability**, as specified in Chapters 10-14

**CORBA Interworking**, as specified in Chapters 15, 16, and 17

**Mapping of OMG IDL to the C programming language**, as specified in Chapter 18

**Mapping of OMG IDL to the C++ programming language**, as specified in Chapter 19

**Mapping of OMG IDL to the Smalltalk programming language**, as specified in Chapter 20

**Mapping of OMG IDL to the COBOL programming language**, as specified in Chapter 21

**Mapping of OMG IDL to the Ada programming language**, as specified in Chapter 22

**Mapping of OMG IDL to the Java programming language**, as specified in Chapter 23

The minimum required for a CORBA-compliant system is adherence to the specifications in CORBA Core and one mapping. Each additional language mapping is a separate, optional compliance point. Optional means users aren't required to implement these points if they are unnecessary at their site, but if implemented, they must adhere to the *CORBA* specifications to be called CORBA-compliant. For instance, if a vendor supports C++, their ORB must comply with the OMG IDL to C++ binding specified in this manual.

Interoperability and Interworking are separate compliance points. For detailed information about Interworking compliance, refer to "Products Subject to Compliance" on page 15-34.

---

## 0.6 *Structure of This Manual*

This manual is divided into the categories of Core, Interoperability, Interworking, and individual Language Mappings. These divisions reflect the compliance points of CORBA. In addition to this preface, *CORBA: Common Object Request Broker Architecture and Specification* contains the following chapters:

### Core

**Chapter 1 -- The Object Model** describes the computation model that underlies the CORBA architecture.

**Chapter 2 -- CORBA Overview** describes the overall structure of the ORB architecture and includes information about CORBA interfaces and implementations.

**Chapter 3 -- OMG IDL Syntax and Semantics** describes OMG interface definition language (OMG IDL), which is the language used to describe the interfaces that client objects call and object implementations provide.

**Chapter 4-- ORB Interface** describes the interface to the ORB functions that do not depend on object adapters: these operations are the same for all ORBs and object implementations.

**Chapter 5-- The Dynamic Invocation Interface** describes the DII, the client's side of the interface that allows dynamic creation and invocation of request to objects.

**Chapter 6 -- The Dynamic Skeleton Interface** describes the DSI, the server's-side interface that can deliver requests from an ORB to an object implementation that does not have compile-time knowledge of the type of the object it is implementing. DSI is the server's analogue of the client's Dynamic Invocation Interface (DII).

**Chapter 7 -- Dynamic Management of Any Values** describes the interface for the Dynamic Any type. This interface allows statically-typed programming languages such as C and Java to create or receive values of type Any without compile-time knowledge that the typer contained in the Any.

**Chapter 8 -- Interface Repository** describes the component of the ORB that manages and provides access to a collection of object definitions.

**Chapter 9-- Portable Object Adapter** describes a group of IDL interfaces than an implementation uses to access ORB functions.

### Interoperability

**Chapter 10-- Interoperability Overview** explains the interoperability architecture and introduces the subjects pertaining to interoperability: inter-ORB bridges; general and Internet inter-ORB protocols (GIOP and IIOP); and environment-specific, inter-ORB protocols (ESIOPs).

---

**Chapter 11 -- ORB Interoperability Architecture** introduces the framework of ORB interoperability, including information about domains; approaches to inter-ORB bridges; what it means to be compliant with ORB interoperability; and ORB Services and Requests.

**Chapter 12 -- Building Inter-ORB Bridges** explains how to build bridges for an implementation of interoperating ORBs.

**Chapter 13 -- General Inter-ORB Protocol** describes the general inter-ORB protocol (GIOP) and includes information about the GIOP's goals, syntax, format, transport, and object location. This chapter also includes information about the Internet inter-ORB protocol (IIOP).

**Chapter 14 -- DCE ESIOP - Environment-Specific Inter-ORB Protocol (ESIOP)** describes a protocol for the OSF DCE environment. The protocol is called the DCE Environment Inter-ORB Protocol (DCE ESIOP).

## Interworking

**Chapter 15 -- Interworking Architecture** describes the architecture for communication between two object management systems: Microsoft's COM (including OLE) and the OMG's CORBA.

**Chapter 16 -- Mapping: COM and CORBA** describes the data type and interface mapping between COM and CORBA. The mappings are described in the context of both Win16 and Win32 COM.

**Chapter 17 -- Mapping: OLE Automation and CORBA** describes the two-way mapping between OLE Automation (in ODL) and CORBA (in OMG IDL).

Note: Chapter 17 also includes an appendix describing solutions that vendors might implement to support existing and older OLE Automation controllers and an appendix that provides an example of how the Naming Service could be mapped to an OLE Automation interface according to the Interworking specification.

## Language Mappings

**Chapter 18 -- C Language Mapping** defines the mapping of OMG IDL to the C programming language.

**Chapter 19 -- Mapping of OMG IDL to C++** - Includes the following information:

- Mapping of OMG IDL to C++ maps the constructs of OMG IDL to the C++ programming language.
- Mapping of Pseudo Objects to C++ maps OMG IDL pseudo objects to the C++ programming language.
- Server-Side Mapping explains the portability constraints for an object implementation written in C++.
- The C++ language mapping also includes several appendices. One contains C++ definitions for CORBA, another contains alternate C++ mappings, and another contains C++ keywords.

---

**Chapter 20-- Mapping OMG IDL to Smalltalk** - includes the following information:

- Mapping of OMG IDL to Smalltalk maps the constructs of OMG IDL to the Smalltalk programming language.
- Mapping of Pseudo Objects to Smalltalk maps OMG IDL pseudo-objects to Smalltalk.

**Chapter 21 -- Mapping of OMG IDL to COBOL** maps the constructs of OMG IDL to the COBOL programming language.

**Chapter 22 - Mapping of OMG IDL to Ada** maps the constructs of OMG IDL to the Ada programming language.

**Chapter 23 - Mapping of OMG IDL to Java** maps the constructs of OMG IDL to the Java programming language.

**Appendix A-** contains OMG IDL tags that can identify an Object Service, a component, or a profile.

## 0.7 Acknowledgements

The following companies submitted parts of the specifications that were approved by the Object Management Group to become *CORBA*:

- BNR Europe Ltd.
- Defense Information Systems Agency
- Expersoft Corporation
- FUJITSU LIMITED
- Genesis Development Corporation
- Gensym Corporation
- IBM Corporation
- ICL plc
- IONA Technologies Ltd.
- Digital Equipment Corporation
- Hewlett-Packard Company
- HyperDesk Corporation
- Micro Focus Limited
- MITRE Corporation
- NCR Corporation
- Novell USG
- Object Design, Inc.
- Objective Interface Systems, Inc.
- OC Systems, Inc.
- Open Group - Open Software Foundation
- Siemens Nixdorf Informationssysteme AG
- Sun Microsystems Inc.
- SunSoft, Inc.
- Sybase, Inc.

- 
- Telefónica Investigación y Desarrollo S.A. Unipersonal
  - Visual Edge Software, Ltd.

In addition to the preceding contributors, the OMG would like to acknowledge Mark Linton at Silicon Graphics and Doug Lea at the State University of New York at Oswego for their work on the C++ mapping.

## 0.8 References

IDL Type Extensions RFP, March 1995. OMG TC Document 95-1-35.

The Common Object Request Broker: Architecture and Specification, Revision 2.1, August 1997.

CORBA services: Common Object Services Specification, Revised Edition, OMG TC Document 95-3-31.

COBOL Language Mapping RFP, December 1995. OMG TC document 95-12-10.

COBOL 85 ANSI X3.23-1985 / ISO 1989-1985.

IEEE Standard for Binary Floating-Point Arithmetic, ANIS/IEEE Std 754-1985.

XDR: External Data Representation Standard, RFC1832, R. Srinivasan, Sun Microsystems, August 1995.

OSF Character and Code Set Registry, OSF DCE SIG RFC 40.1 (Public Version), S. (Martin) O'Donnell, June 1994.

RPC Runtime Support For I18N Characters — Functional Specification, OSF DCE SIG RFC 41.2, M. Romagna, R. Mackey, November 1994.

X/Open System Interface Definitions, Issue 4 Version 2, 1995.