
**Information technology — Object
Management Group — Common Object
Request Broker Architecture (CORBA) —
Part 3:
Components**

*Technologies de l'information — OMG (Object Management Group) —
CORBA (Common Object Request Broker Architecture) —*

Partie 3: Composants



COPYRIGHT PROTECTED DOCUMENT

© ISO/IEC 2012

All rights reserved. Unless otherwise specified, no part of this publication may be reproduced or utilized in any form or by any means, electronic or mechanical, including photocopying and microfilm, without permission in writing from either ISO at the address below or ISO's member body in the country of the requester.

ISO copyright office
Case postale 56 • CH-1211 Geneva 20
Tel. + 41 22 749 01 11
Fax + 41 22 749 09 47
E-mail copyright@iso.org
Web www.iso.org

Published in Switzerland

Table of Contents

Foreword	xi
Introduction	xiii
1 Scope	1
2 Conformance and Compliance	1
3 References	3
3.1 Normative References	3
3.2 Non-normative References	4
4 Terms and definitions	4
4.1 Terms Defined in this International Standard	4
4.2 Keywords for Requirement statements	7
5 Symbols (and abbreviated terms)	7
6 Component Model	9
6.1 Component Model	9
6.1.1 Component Levels	9
6.1.2 Ports	9
6.1.3 Components and Facets	10
6.1.4 Component Identity	11
6.1.5 Component Homes	11
6.2 Component Definition	11
6.3 Component Declaration	11
6.3.1 Basic Components	11
6.3.2 Equivalent IDL	12
6.3.3 Component Body	13
6.4 Facets and Navigation	13
6.4.1 Equivalent IDL	13
6.4.2 Semantics of Facet References	14

6.4.3 Navigation	14
6.4.4 Provided References and Component Identity	17
6.4.5 Supported interfaces	18
6.5 Receptacles	20
6.5.1 Equivalent IDL	20
6.5.2 Behavior	21
6.5.3 Receptacles Interface	22
6.6 Events	25
6.6.1 Event types	25
6.6.2 EventConsumer Interface	26
6.6.3 Event Service Provided by Container	27
6.6.4 Event Sources—Publishers and Emitters	27
6.6.5 Publisher	28
6.6.6 Emitters	29
6.6.7 Event Sinks	30
6.6.8 Events interface	30
6.7 Homes	34
6.7.1 Equivalent Interfaces	34
6.7.2 Primary Key Declarations	36
6.7.3 Explicit Operations in Home Definitions	37
6.7.4 Home inheritance	38
6.7.5 Semantics of Home Operations	39
6.7.6 CCMHome Interface	41
6.7.7 KeylessCCMHome Interface	42
6.8 Home Finders	42
6.9 Component Configuration	44
6.9.1 Exclusive Configuration and Operational Life Cycle Phases	45
6.10 Configuration with Attributes	46
6.10.1 Attribute Configurators	46
6.10.2 Factory-based Configuration	47
6.11 Component Inheritance	49
6.11.1 CCMObject Interface	50
6.12 Conformance Requirements	51
6.12.1 A Note on Tools	53
6.12.2 Changes to Object Services	53
7 OMG CIDL Syntax and Semantics	55
7.1 General	55

7.2 Lexical Conventions	55
7.2.1 Keywords	56
7.3 OMG CIDL Grammar	56
7.4 OMG CIDL Specification	58
7.5 Composition Definition	58
7.5.1 Life Cycle Category and Constraints	59
7.6 Home Executor Definition	59
7.7 Home Implementation Declaration	60
7.8 Storage Home Binding	61
7.9 Home Persistence Declaration	61
7.10 Executor Definition	61
7.11 Segment Definition	62
7.12 Segment Persistence Declaration	62
7.13 Facet Declaration	63
7.14 Feature Delegation Specification	63
7.15 Abstract Storage Home Delegation Specification	64
7.16 Executor Delegation Specification	65
7.17 Abstract Spec Declaration	66
7.18 Proxy Home Declaration	66
8 CCM Implementation Framework	67
8.1 Introduction	67
8.2 Component Implementation Framework (CIF) Architecture	67
8.2.1 Component Implementation Definition Language (CIDL)	67
8.2.2 Component persistence and behavior	67
8.2.3 Implementing a CORBA Component	67
8.2.4 Behavioral elements: Executors	68
8.2.5 Unit of implementation : Composition	68
8.2.6 Composition structure	69
8.2.7 Compositions with Managed Storage	75
8.2.8 Relationship between Home Executor and Abstract Storage Home	77
8.2.9 Executor Definition	89
8.2.10 Proxy Homes	96
8.2.11 Component Object References	97

8.3 Language Mapping	99
8.3.1 Overview	99
8.3.2 Common Interfaces	100
8.3.3 Mapping Rules	101
9 The Container Programming Model	109
9.1 General	109
9.2 Introduction	109
9.2.1 External API Types	110
9.2.2 Container API Type	111
9.2.3 CORBA Usage Model	111
9.2.4 Component Categories	111
9.3 The Server Programming Environment	112
9.3.1 Component Containers	112
9.3.2 CORBA Usage Model	113
9.3.3 Component Factories	114
9.3.4 Component Activation	114
9.3.5 Servant Lifetime Management	114
9.3.6 Transactions	115
9.3.7 Security	117
9.3.8 Events	117
9.3.9 Persistence	118
9.3.10 Application Operation Invocation	119
9.3.11 Component Implementations	120
9.3.12 Component Levels	120
9.3.13 Component Categories	120
9.4 Server Programming Interfaces - Basic Components	124
9.4.1 Component Interfaces	124
9.4.2 Interfaces Common to both Container API Types	125
9.4.3 Interfaces Supported by the Session Container API Type	130
9.4.4 Interfaces Supported by the Entity Container API Type	132
9.5 Server Programming Interfaces - Extended Components	134
9.5.1 Interfaces Common to both Container API Types	134
9.5.2 Interfaces Supported by the Session Container API Type	136
9.5.3 Interfaces Supported by the Entity Container API Type	138
9.6 The Client Programming Model	144
9.6.1 Component-aware Clients	144
9.6.2 Component-unaware Clients	148

10	Integrating with Enterprise JavaBeans	151
10.1	Introduction	151
10.2	Enterprise JavaBeans Compatibility Objectives and Requirements	152
10.3	CORBA Component Views for EJBs	153
10.3.1	Mapping of EJB to Component IDL definitions	153
10.3.2	Translation of CORBA Component requests into EJB requests	157
10.3.3	Interoperability of the View	158
10.3.4	CORBA Component view Example	160
10.4	EJB views for CORBA Components	162
10.4.1	Mapping of Component IDL to Enterprise JavaBeans specifications	162
10.4.2	Translation of EJB requests into CORBA Component Requests	164
10.4.3	Interoperability of the View	166
10.4.4	Example	168
10.5	Compliance with the Interoperability of Integration Views	169
10.6	Comparing CCM and EJB	169
10.6.1	The Home Interfaces	170
10.6.2	The Component Interfaces	171
10.6.3	The Callback Interfaces	173
10.6.4	The Context Interfaces	174
10.6.5	The Transaction Interfaces	175
10.6.6	The Metadata Interfaces	176
11	Interface Repository Metamodel	177
11.1	Introduction	177
11.1.1	BaseIDL Package	177
11.1.2	ComponentIDL Package	188
11.2	Conformance Criteria	196
11.2.1	Conformance Points	197
11.3	MOF DTDs and IDL for the Interface Repository Metamodel	197
11.3.1	XMI DTD	197
11.3.2	IDL for the BaseIDL Package	222
11.3.3	IDL for the ComponentIDL Package	244
12	CIF Metamodel	263
12.1	CIF Package	263
12.2	Classes and Associations	263

12.2.1 ComponentImplDef	264
12.2.2 SegmentDef	265
12.2.3 ArtifactDef	265
12.2.4 Policy	265
12.2.5 HomImplDef	266
12.3 Conformance Criteria	267
12.3.1 Conformance Points	267
12.4 MOF DTDs and IDL for the CIF Metamodel	267
12.4.1 XMI DTD	268
12.4.2 IDL for the CIF Package	268
13 Lightweight CCM Profile	275
13.1 Summary	275
13.2 Changes associated with excluding support for persistence	276
13.3 Changes associated with excluding support for introspection, navigation and type-specific operations redundant with generic operations	278
13.4 Changes associated with excluding support for segmentation ...	279
13.5 Changes associated with excluding support for transactions	280
13.6 Changes associated with excluding support for security	280
13.7 Changes associated with excluding support for configurators ...	281
13.8 Changes associated with excluding support for proxy homes	281
13.9 Changes associated with excluding support for home finders	281
13.10 Changes adding additional restrictions to the extended model not represented by exclusions above	282
14 Deployment PSM for CCM	283
14.1 Overview	283
14.2 Definition of Meta-Concepts	284
14.2.1 Component	284
14.2.2 ImplementationArtifact	285
14.2.3 PackageI	285
14.3 PIM to PSM for CCM Transformation	285
14.3.1 ComponentInterfaceDescription	285

14.3.2	PlanSubcomponentPortEndpoint	286
14.3.3	Application	286
14.3.4	RepositoryManager	287
14.3.5	SatisfierProperty	287
14.4	PSM for CCM to PSM for CCM for IDL Transformation	287
14.4.1	Generic Transformation Rules	287
14.4.2	Special Transformation Rules	289
14.4.3	Mapping to IDL	290
14.5	PSM for CCM to PSM for CCM for XML Transformation	290
14.5.1	Generic Transformation Rules	290
14.5.2	Special Transformation Rules	291
14.5.3	Transformation Exceptions and Extensions	295
14.5.4	Interpretation of Relative References	296
14.5.5	Mapping to XML	297
14.6	Miscellaneous	297
14.6.1	Entry Points	297
14.6.2	Homes	298
14.6.3	Valuetype Factories	298
14.6.4	Discovery and Initialization	298
14.6.5	Location	299
14.6.6	Segmentation	299
14.7	Migration Issues	300
14.7.1	Component Implementations	300
14.7.2	Component and Assembly Packages and Metadata	300
14.7.3	Component Deployment Systems	300
14.8	Metadata Vocabulary	301
14.8.1	Implementation Selection Requirements	301
14.8.2	Monolithic Implementation Resource Requirements	301
15	Deployment IDL for CCM	303
16	XML Schema for CCM	317
Annex A	- Legal Information	337

Foreword

ISO (the International Organization for Standardization) is a worldwide federation of national standards bodies (ISO member bodies). The work of preparing International Standards is normally carried out through ISO technical committees. Each member body interested in a subject for which a technical committee has been established has the right to be represented on that committee. International organizations, governmental and non-governmental, in liaison with ISO, also take part in the work. ISO collaborates closely with the International Electrotechnical Commission (IEC) on all matters of electrotechnical standardization.

International Standards are drafted in accordance with the rules given in the ISO/IEC Directives, Part 2.

The main task of technical committees is to prepare International Standards. Draft International Standards adopted by the technical committees are circulated to the member bodies for voting. Publication as an International Standard requires approval by at least 75 % of the member bodies casting a vote.

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. ISO shall not be held responsible for identifying any or all such patent rights.

ISO/IEC 19500-3 was prepared by Technical Committee ISO/IEC JTC1, Information technology, in collaboration with the Object Management Group (OMG), following the submission and processing as a Publicly Available Specification (PAS) of the OMG Common Object Request Broker Architecture (CORBA) specification Part 3 Version 3.1 CORBA Components.

ISO/IEC 19500-3 is related to:

- ITU-T Recommendation X.902 (1995) | ISO/IEC 10746-2:1996, Information Technology - Open Distributed Processing - Reference Model: Foundations
- ITU-T Recommendation X.903 (1995) | ISO/IEC 10746-3:1996, Information Technology - Open Distributed Processing - Reference Model: Architecture
- ITU-T Recommendation X.920 (1997) | ISO/IEC 14750:1997, Information Technology - Open Distributed Processing - Interface Definition Language
- ISO/IEC 19500-2, Information Technology - Open Distributed Processing - CORBA Specification Part 1: CORBA Interfaces
- ISO/IEC 19500-3, Information Technology - Open Distributed Processing - CORBA Specification Part 2: CORBA Interoperability

ISO/IEC 19500 consists of the following parts, under the general title *Information technology - Open distributed processing - CORBA specification*:

- Part 1: CORBA Interfaces
- Part 2: CORBA Interoperability
- Part 3: CORBA Components

ISO/IEC 19500-3:2012(E)

It is the common core of the CORBA specification. Optional parts of CORBA, such as mappings to particular programming languages, Real-time CORBA extensions, and the minimum CORBA profile for embedded systems are documented in the other specifications that together comprise the complete CORBA specification. Please visit the CORBA download page at http://www.omg.org/technology/documents/corba_spec_catalog.htm to find the complete CORBA specification set.

Apart from this Foreword, the text of this International Standard is identical with that for the OMG specification for CORBA, v3.1.1, Part 3.

Introduction

The rapid growth of distributed processing has led to a need for a coordinating framework for this standardization and ITU-T Recommendations X.901-904 | ISO/IEC 10746, the Reference Model of Open Distributed Processing (RM-ODP) provides such a framework. It defines an architecture within which support of distribution, interoperability and portability can be integrated.

RM-ODP Part 2 (ISO/IEC 10746-2) defines the foundational concepts and modeling framework for describing distributed systems. The scopes and objectives of the RM-ODP Part 2 and the UML, while related, are not the same and, in a number of cases, the RM-ODP Part 2 and the UML specification use the same term for concepts which are related but not identical (e.g., interface). Nevertheless, a specification using the Part 2 modeling concepts can be expressed using UML with appropriate extensions (using stereotypes, tags, and constraints).

RM-ODP Part 3 (ISO/IEC 10746-3) specifies a generic architecture of open distributed systems, expressed using the foundational concepts and framework defined in Part 2. Given the relation between UML as a modeling language and Part 3 of the RM-ODP standard, it is easy to show that UML is suitable as a notation for the individual viewpoint specifications defined by the RM-ODP.

This part of ISO/IEC 19500 (CORBA Components) is a standard for the technology specification of an ODP system. It defines a technology to provide the infrastructure required to support functional distribution of an ODP system, specifying functions required to manage physical distribution, communications, processing and storage, and the roles of different technology objects in supporting those functions.

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 that 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.

ISO/IEC 19500-3:2012(E)

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.

The architecture and specifications described in this standard are aimed at software designers and developers who want to produce applications that comply with OMG specifications for the Object Request Broker (ORB), or this standard (ISO/IEC 19500). The benefit of compliance is, in general, to be able to produce interoperable applications that are based on distributed, interoperating objects. 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.

This Part of this International Standard includes a non-normative annex.

Information technology - Object Management Group Common Object Request Broker Architecture (CORBA), Components

1 Scope

This part of ISO/IEC 19500 defines:

- The syntax and semantics of a component model (see Clause 6, 'Component Model'), based on CORBA IDL, and a corresponding meta-model (see Clause 11, 'Interface Repository Metamodel').
- A language to describe the structure and state of component implementations (see Clause 7, 'OMG CIDL Syntax and Semantics'), and a corresponding meta-model (see Clause 12, 'CIF Metamodel').
- A programming model for constructing component implementations (see Clause 8, 'CCM Implementation Framework').
- A runtime environment for component implementations (see Clause 9, 'The Container Programming Model').
- Interaction between components and Enterprise Java Beans (see Clause 10, 'Integrating with Enterprise JavaBeans').
- Meta-data for describing component-based applications, and interfaces for their deployment (see Clause 14, 'Deployment PSM for CCM').
- A lightweight subset of the component model, programming model and runtime environment (see Clause 13, 'Lightweight CCM Profile').

2 Conformance and Compliance

The following conformance points are defined:

1. A CORBA COS vendor shall provide the relevant changes to the Lifecycle, Transaction, and Security Services identified in "Changes to Object Services" on page 53.
2. A CORBA Component vendor shall provide a conforming implementation of the Basic Level of CORBA Components. A Lightweight CORBA Component vendor shall provide a conforming implementation of the Lightweight CCM Profile as specified in item 8 below.
3. A CORBA Component vendor may provide a conforming implementation of the Extended Level of CORBA Components.
4. To be conformant at the Basic level a non-Java product shall implement (at a minimum) the following:
 - the IDL extensions and generation rules to support the client and server side component model for basic level components.
 - CIDL. The multiple segment feature of CIDL ("Segment Definition" on page 62) need not be supported for basic components.
 - a container for hosting basic level CORBA components.

- the XML deployment descriptors and associated zip files for basic components.

Such implementations shall work on a CORBA ORB as defined in #1 above.

5. To be conformant at the Basic level a Java product shall implement (at a minimum):

- EJB1.1, including support for the EJB 1.1 XML DTD.
- the java to IDL mapping, also known as RMI/IIOP.
- EJB to IDL mapping as defined in “Translation of CORBA Component requests into EJB requests” on page 157.

Such implementations shall work in a CORBA interoperable environment, including interoperable support for IIOP CORBA transactions, and CORBA security.

6. To be conformant at the extended level, a product shall implement (at a minimum) the requirements needed to achieve Basic PLUS:

- IDL extensions to support the client and server side component model for extended level components.
- A container for hosting extended level CORBA components.
- The XML deployment descriptors and associated zip files for basic and enhanced level components in the format defined in “Deployment PSM for CCM” on page 283.

Such implementations shall work on a CORBA ORB as defined in #1 above.

7. The Lightweight CCM profile is a conformance point based on the extended model as defined above. “Lightweight CCM Profile” on page 275 defines the specific parts of this CCM specification that are impacted and the normative specific subsetting of CCM. In summary, the following general capabilities (and associated machinery) are excluded from the extended model to define this conformance point:

- Persistence (only session and service components are supported)
- Introspection
- Navigation
- Redundancies, preferring generic over specific
- Segmentation (not allowed for session or service components)
- Transactions
- Security
- Configurators
- Proxy homes
- Home finders
- CIDL
- POA related mandates

8. A CORBA Component vendor may optionally support EJB clients interacting with CORBA Components, by implementing the IDL to EJB mapping as defined in “Translation of EJB requests into CORBA Component Requests” on page 164.

3 References

3.1 Normative References

- [CORBA] Object Management Group, "Common Object Request Broker Architecture," version 3.0.3, OMG document number formal/04-03-01. Available from <http://www.omg.org/cgi-bin/doc?formal/04-03-01>
- [D+C] Object Management Group, "*Deployment and Configuration of Component-based Distributed Applications Specification*," version 1.1. OMG document number ptc/05-01-07. Available from <http://www.omg.org/cgi-bin/doc?ptc/05-01-07>
- [EJB] Java Community Process, "Enterprise JavaBeans Specification - Version 1.1" (with errata), June 8, 2000. Available from <http://jcp.org/aboutJava/communityprocess/maintenance/jsr905>
- [HTTP] R. Fielding, J. Gettys, J. Mogul, H. Frystyk, L. Masinter, P. Leach, T. Berners-Lee, RFC 2616: "*Hypertext Transfer Protocol -- HTTP/1.1*." June 1999. Available from <http://www.ietf.org/rfc/rfc2616.txt>
- [INS] Object Management Group, "*Naming Service Specification*," version 1.1. February 2001. OMG document number formal/01-02-65. Available from <http://www.omg.org/cgi-bin/doc?formal/01-02-65>
- [JIDL] Object Management Group, "Java™ to IDL Language Mapping Specification," version 1.3. September 2003. OMG document number formal/03-09-04. Available from <http://www.omg.org/cgi-bin/doc?formal/03-09-04>
- [MDA] Object Management Group, "MDA Guide - Version 1.0.1" Available from <http://www.omg.org/cgi-bin/doc?omg/03-06-01>
- [MOF1] ISO/IEC 19503:2005 Information technology -- XML Metadata Interchange (XMI)
- [PSS] Object Management Group, "*Persistent State Service*," version 2.0. September 2002. OMG document number formal/02-09-06. Available from <http://www.omg.org/cgi-bin/doc?formal/02-09-06>
- [RFC2119] IETF RFC2119, "Key words for use in RFCs to Indicate Requirement Levels", S. Bradner, March 1997. Available from <http://ietf.org/rfc/rfc2119>
- [SSS] Object Management Group, "*Security Service Specification*," version 1.8. March 2002. OMG document number formal/02-03-11. Available from <http://www.omg.org/cgi-bin/doc?formal/02-03-11>
- [TSS] Object Management Group, "*Transaction Service Specification*," version 1.4. September 2003. OMG document number formal/03-09-02. Available from <http://www.omg.org/cgi-bin/doc?formal/03-09-02>

ISO/IEC 19500-3:2012(E)

- [UML1] /IEC 19501:2005 Information technology - Unified Modeling Language (UML)
- [UPC] Object Management Group, “UML™ Profile for CORBA™ Specification,” version 1.0. Adopted specification. April 2002. OMG document number formal/02-04-01. Available from <http://www.omg.org/cgi-bin/doc?formal/02-04-01>
- [URI] T. Berners-Lee, R. Fielding, L. Masinter, RFC 2396: “Uniform Resource Identifiers (URI): Generic Syntax.” August 1998, available from <http://www.ietf.org/rfc/rfc2396.txt>
- [XMI] ISO/IEC 19503:2005 Information technology -- XML Metadata Interchange (XMI)
- [XML] World Wide Web Consortium (W3C), “Extensible Markup Language (XML),” version 1.0 (second edition). W3C Recommendation, October 6, 2000. Available from <http://www.w3.org/TR/REC-xml>
- [XSD] World Wide Web Consortium (W3C), “XML Schema Part 1: Structures.” W3C Recommendation, May 2, 2001. Available from <http://www.w3.org/TR/xmlschema-1/>
- World Wide Web Consortium (W3C), “XML Schema Part 2: Datatypes.” W3C Recommendation, May 2, 2001. Available from <http://www.w3.org/2001/xmlschema-2/>

3.2 Non-normative References

- [LCS] Object Management Group, “Life Cycle Service Specification,” version 1.2. September 2002. OMG document number formal/02-09-01. Available from <http://www.omg.org/cgi-bin/doc?formal/02-09-01>
- [NSS] Object Management Group, “Notification Service Specification,” version 1.1. October 2004. OMG document number formal/04-10-11. Available from <http://www.omg.org/cgi-bin/doc?formal/04-10-11>