Glossary

activation — Preparing an object to execute an operation. For example, copying the persistent form of methods and stored data into an executable address space to allow execution of the methods on the stored data.

adapter — Same as object adapter.

attribute — An identifiable association between an object and a value. An attribute **A** is made visible to clients as a pair of operations: **get_A** and **set_A**. Readonly attributes only generate a **get** operation.

behavior — The observable effects of an object performing the requested operation including its results binding. See language binding, dynamic invocation, static invocation, or method resolution for alternatives.

class — See interface and implementation for alternatives.

client — The code or process that invokes an operation on an object.

context object — A collection of name-value pairs that provides environmental or user-preference information.

CORBA — Common Object Request Broker Architecture.

data type — A categorization of values operation arguments, typically covering both behavior and representation (i.e., the traditional non-OO programming language notion of type).

deactivation — The opposite of activation.

deferred synchronous request — A request where the client does not wait for completion of the request, but does intend to accept results later. Contrast with synchronous request and one-way request.

domain	A concept important to interoperability, it is a distinct scope, within which common characteristics are exhibited, common rules observed, and over which a distribution transparency is preserved.
dynamic invocation	Constructing and issuing a request whose signature is possibly not known until run-time.
dynamic skeleton	An interface-independent kind of skeleton, used by servers to handle requests whose signatures are possibly not known until run-time.
externalized object reference	An object reference expressed as an ORB-specific string. Suitable for storage in files or other external media.
implementation	A definition that provides the information needed to create an object and allow the object to participate in providing an appropriate set of services. An implementation typically includes a description of the data structure used to represent the core state associated with an object, as well as definitions of the methods that access that data structure. It will also typically include information about the intended interface of the object.
implementation definition language	A notation for describing implementations. The implementation definition language is currently beyond the scope of the ORB standard. It may contain vendor-specific and adapter-specific notations.
implementation inheritance	The construction of an implementation by incremental modification of other implementations. The ORB does not provide implementation inheritance. Implementation inheritance may be provided by higher level tools.
implementation object	An object that serves as an implementation definition. Implementation objects reside in an implementation repository.
implementation repository	A storage place for object implementation information.
inheritance	The construction of a definition by incremental modification of other definitions. See *interface* and *implementation inheritance*.
instance	An object is an instance of an interface if it provides the operations, signatures and semantics specified by that interface. An object is an instance of an implementation if its behavior is provided by that implementation.
interface	A listing of the operations and attributes that an object provides. This includes the signatures of the operations, and the types of the attributes. An interface definition ideally includes the semantics as well. An object *satisfies* an interface if it can be specified as the target object in each potential request described by the interface.
interface inheritance	The construction of an interface by incremental modification of other interfaces. The IDL language provides interface inheritance.
interface object	An object that serves to describe an interface. Interface objects reside in an interface repository.
interface repository	A storage place for interface information.

interface type — A type satisfied by any object that satisfies a particular interface.

interoperability — The ability for two or more ORBs to cooperate to deliver requests to the proper object. Interoperating ORBs appear to a client to be a single ORB.

language binding or **mapping** — The means and conventions by which a programmer writing in a specific programming language accesses ORB capabilities.

method — An implementation of an operation. Code that may be executed to perform a requested service. Methods associated with an object may be structured into one or more programs.

method resolution — The selection of the method to perform a requested operation.

multiple inheritance — The construction of a definition by incremental modification of more than one other definition.

object — A combination of state and a set of methods that explicitly embodies an abstraction characterized by the behavior of relevant requests. An object is an instance of an implementation and an interface. An object models a real-world entity, and it is implemented as a computational entity that encapsulates state and operations (internally implemented as data and methods) and responds to request or services.

object adapter — The ORB component which provides object reference, activation, and state related services to an object implementation. There may be different adapters provided for different kinds of implementations.

object creation — An event that causes the existence of an object that is distinct from any other object.

object destruction — An event that causes an object to cease to exist.

object implementation — Same as implementation.

object reference — A value that unambiguously identifies an object. Object references are never reused to identify another object.

objref — An abbreviation for object reference.

one-way request — A request where the client does not wait for completion of the request, nor does it intend to accept results. Contrast with deferred synchronous request and synchronous request.

operation — A service that can be requested. An operation has an associated signature, which may restrict which actual parameters are valid.

operation name — A name used in a request to identify an operation.

ORB — Object Request Broker. Provides the means by which clients make and receive requests and responses.

ORB core — The ORB component which moves a request from a client to the appropriate adapter for the target object.

parameter passing mode — Describes the direction of information flow for an operation parameter. The parameter passing modes are **IN**, **OUT**, and **INOUT**.

persistent object — An object that can survive the process or thread that created it. A persistent object exists until it is explicitly deleted.

portable object adapter — The object adapter described in Chapter 9.

referential integrity — The property ensuring that an object reference that exists in the state associated with an object reliably identifies a single object.

repository — See interface repository and implementation repository.

request — A client issues a request to cause a service to be performed. A request consists of an operation and zero or more actual parameters.

results — The information returned to the client, which may include values as well as status information indicating that exceptional conditions were raised in attempting to perform the requested service.

server — A process implementing one or more operations on one or more objects.

server object — An object providing response to a request for a service. A given object may be a client for some requests and a server for other requests.

signature — Defines the parameters of a given operation including their number order, data types, and passing mode; the results if any; and the possible outcomes (normal vs. exceptional) that might occur.

single inheritance — The construction of a definition by incremental modification of one definition. Contrast with multiple inheritance.

skeleton — The object-interface-specific ORB component which assists an object adapter in passing requests to particular methods.

state — The time-varying properties of an object that affect that object's behavior.

static invocation — Constructing a request at compile time. Calling an operation via a stub procedure.

stub — A local procedure corresponding to a single operation that invokes that operation when called.

synchronous request — A request where the client pauses to wait for completion of the request. Contrast with deferred synchronous request and one-way request.

transient object — An object whose existence is limited by the lifetime of the process or thread that created it.

type — See *data type* and *interface*.

value Any entity that may be a possible actual parameter in a request. Values that serve to identify objects are called object references.