
HP NonStop CORBA 2.6.1Glossary

Part number: 525939 002. Published February 2004.

[Legal Notice](#)

Abstract

This glossary defines terms used in NonStop CORBA documentation.

Product Version: HP NonStop" CORBA® 2.6.1 C11

Table of Contents

[New and Changed Information](#)

[About This Glossary](#)

[Manuals in the NonStop CORBA Set](#)

[Glossary](#)

© Copyright 2004 Hewlett-Packard Development Company L.P.

Confidential computer software. Valid license from HP required for possession, use or copying. Consistent with FAR 12.211 and 12.212, Commercial Computer Software, Computer Software Documentation, and Technical Data for Commercial Items are licensed to the U.S. Government under vendor's standard commercial license.

The information contained herein is subject to change without notice. The only warranties for HP products and services are set forth in the express warranty statements accompanying such products and services. Nothing herein should be construed as constituting an additional warranty. HP shall not be liable for technical or editorial errors or omissions contained herein.

Export of the information contained in this publication may require authorization from the U.S. Department of Commerce.

Microsoft, Windows, and Windows NT are U.S. registered trademarks of Microsoft Corporation.

Intel, Pentium, and Celeron are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

Java is a U.S. trademark of Sun Microsystems, Inc.

Motif, OSF/1, UNIX, X/Open, and the "X" device are registered trademarks, and IT DialTone and The Open Group are trademarks of The Open Group in the U.S. and other countries.

Open Software Foundation, OSF, the OSF logo, OSF/1, OSF/Motif, and Motif are trademarks of the Open Software Foundation, Inc.

OSF MAKES NO WARRANTY OF ANY KIND WITH REGARD TO THE OSF MATERIAL PROVIDED HEREIN, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE.

OSF shall not be liable for errors contained herein or for incidental consequential damages in connection with the furnishing, performance, or use of this material.

© 1990, 1991, 1992, 1993 Open Software Foundation, Inc. The OSF documentation and the OSF software to which it relates are derived in part from materials supplied by the following:

© 1987, 1988, 1989 Carnegie-Mellon University. © 1989, 1990, 1991 Digital Equipment Corporation. © 1985, 1988, 1989, 1990 Encore Computer Corporation. © 1988 Free Software Foundation, Inc. © 1987, 1988, 1989, 1990, 1991 Hewlett-Packard Company. © 1985, 1987, 1988, 1989, 1990, 1991, 1992 International Business Machines Corporation. © 1988, 1989 Massachusetts Institute of Technology. © 1988, 1989, 1990 Mentat Inc. © 1988 Microsoft Corporation. © 1987, 1988, 1989, 1990, 1991, 1992 SecureWare, Inc. © 1990, 1991 Siemens Nixdorf Informationssysteme AG. © 1986, 1989, 1996, 1997 Sun Microsystems, Inc. © 1989, 1990, 1991 Transarc Corporation.

OSF software and documentation are based in part on the Fourth Berkeley Software Distribution under license from The Regents of the University of California. OSF acknowledges the following individuals and institutions for their role in its development: Kenneth C.R.C. Arnold, Gregory S. Couch, Conrad C. Huang, Ed James, Symmetric Computer Systems, Robert Elz. © 1980, 1981, 1982, 1983, 1985, 1986,

1987, 1988, 1989 Regents of the University of California.

New and Changed Information

New terms have been added to the glossary as needed to provide definitions of terms used with new product features at NonStop CORBA 2.6.1.

About This Glossary

[Manuals in the NonStop CORBA Set](#)

Terms defined in the *NonStop CORBA 2.6.1 Glossary* are taken from the entire NonStop CORBA documentation set. The individual manuals do not contain a glossary.

Most of the terms defined in the *NonStop CORBA 2.6.1 Glossary* are specific to NonStop CORBA. However, some are generic CORBA or SSL terms.

Manuals in the NonStop CORBA Set

The *NonStop CORBA 2.6.1 Getting Started Guide for C++* describes how to install the product and verify installation for the C++ programming environment. It also include an introduction to the product. The *NonStop CORBA 2.6.1 Getting Started Guide for Java* describes how to install the product and verify installation for the Java programming environment. It also include an introduction to the product. The *NonStop CORBA 2.6.1 Administration Guide* gives basic configuration information and describes how to use the NonStop Distributed Component Console and the command-line interface to perform configuration tasks.

The *NonStop CORBA 2.6.1 Programmer's Guide for C++* provides information for C++ application programmers about the HP-specific implementation of CORBA.

NonStop CORBA 2.6.1 Programmer's Guide for Java provides information for Java application programmers with special emphasis on the HP-specific implementation of CORBA.

The *NonStop CORBA 2.6.1 Programmer's Reference* provides information for both C++ and Java application programmers about the IDL compiler, other utilities, minor codes, and system errors. For C++ programmers, it serves as the reference complement to the *HP NonStop CORBA 2.6.1 Programmer's Guide for C++*. For Java programmers, it provides IDL to Java mapping. (Reference information for Java interfaces and classes is provided in Javadoc format as part of the product.)

The *NonStop CORBA 2.6.1 Glossary* provides definitions of CORBA terminology, with special emphasis on the NonStop CORBA implementation.

In addition, Javadoc reference material for Java programmers is provided on the product CD, and the HP NonStop" Distributed Component Console includes online help.

Glossary

[A](#) [B](#) [C](#) [D](#) [E](#) [F](#) [G](#) [H](#) [I](#) [J](#) [K](#) [L](#) [M](#) [N](#) [O](#) [P](#) [Q](#) [R](#) [S](#) [T](#) [U](#) [V](#) [W](#) [X](#) [Y](#) [Z](#)

adapter activator

A server-side object used by the ORB when a request is made for a child POA that does not exist. In such a case, the adapter activator creates the child POA.

admin DB

The database that stores the operational data used by the NSDAdminServer and NSDEnvironServer processes. The location of the database is given by the environment variable NSDOM_ADMIN_DB.

AdminServer

See [NSDAdminServer](#).

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`. Read-only attributes generate only get operations. An attribute is a characteristic or property of an object, and it is typically implemented as a simple data member or as an association with another object or group of objects.

authentication

The verification of a claimant's entitlement to use a claimed identity and/or privilege set. Authentication is encryption's complement. While encryption ensures against eavesdroppers, authentication ensures against imposters. Often, it is not enough to ensure that only its intended receiver can read a message; there must also be a way to verify that the sender of a message is in fact who he or she says they are.

availability

Continuous service. NonStop systems are designed to provide high availability.

base class

A class from which some other class (a subclass) is derived, so that the subclass inherits characteristics from the base class and may also define characteristics of its own.

Basic Object Adapter (BOA)

An early [object adapter](#) that was replaced by the Portable Object Adapter (POA). NonStop CORBA supports the POA only.

binding

(1) In IDL, the relationship between an IDL construct and the corresponding construct in a programming language.

(2) In the Naming Service, the relationship between a name and an object.

(3) In networking, binding is the establishment of a logical connection between communicating entities.

Bootstrap Daemon (BSD)

A protocol that provides operations for resolving an initial object reference and for listing the supported initial reference ids.

business domain

A collection of related business application processes, configuration data, and NonStop" services for CORBA.

cast

To assign a value of one type to a variable of another type. An instance of a class can be cast to an instance of another class if the latter is a subclass of the former or if an interface implements the former.

certificate

A file obtained from a Certificate Authority that contains an encryption key. Certificates are digital documents attesting to the binding of a public key to an individual or other entity. They allow verification of the claim that a given public key does in fact belong to a given individual. Certificates help prevent an imposter from using a key to impersonate someone else.

certification authority (CA)

A party trusted to vouch for the binding between names or identities and public keys. In some systems, certification authorities generate public keys.

child class

An object class that inherits characteristics from another class.

cipher suite

A set of cryptographic algorithms.

class data

Variables that apply, not to a specific object instance, but to an object class as a whole; for example, a variable that reports the number of instances of a class.

cleartext

A message that is not encrypted and can be understood by anyone.

CLI

See [Command Line Interface](#).

client

In object technology, an application component that makes requests of an object. More generally, an application component that makes requests of a server.

client event handler

An object defined by the NonStop CORBA event framework to support asynchronous communication between a NonStop CORBA client and a server that uses other technology. NonStop CORBA provides various types of client event-handler objects to support transport functions such as I/O completions and error handling. Client event handlers are used in legacy server wrappers.

client event handler user

An object, defined by the NonStop CORBA event framework, that uses the services of a client event handler. A legacy server wrapper typically acts as a client event-handler user.

client/server computing

A style of application in which function is divided between a client, which makes requests on behalf of a user, and a server, which executes those requests. In many cases, the client runs on a workstation and the server runs on a larger server machine, but this pattern of distribution is not a necessary characteristic of client/server computing.

Comm Server

A NonStop CORBA process that manages the linkage between remote clients and local servers. A configuration can have multiple Comm Servers, and many clients can use the same Comm Server.

command-line interface (CLI)

The method for accepting input lines from either `stdin` or a script file; that is, a primitive command-line interface.

commit

Declare (a transaction) complete.

Common Object Request Broker Architecture (CORBA)

A definition of standard components and required interfaces to support the interaction of object-oriented components across heterogeneous networks. CORBA is defined by the Object Management Group (OMG).

Common Object Services (COS)

A series of OMG specifications for services useful to applications in the context of CORBA. COS includes a Naming Service, an Event Service, a Lifecycle Service, and a Persistence Service, among others. Also called CORBAServices.

component

A discrete unit of application function, such as a client or server, typically suitable for mixing and matching with other components to create one or more different applications. (One advantage of component-based application development is the ability to use the same component in different applications.)

concurrency

Simultaneous operation. This term can apply to processes running in parallel, simultaneous transactions, or multiple clients using the same object (or other resource) at the same time.

concurrency service

An application service that manages simultaneous access to objects by multiple clients, primarily by supporting read-locking and write-locking requests.

configuration database

A database that stores execution information such as program profiles and operating information needed by NonStop CORBA services. The database is dynamically maintained, and it may be changed by users of the NonStop Distributed Component Console, the configuration management tool, or by CORBA servers.

constructor

A method that creates an object. In the C++ and Java languages, constructors are instance methods that have the same name as their class. Constructors are invoked by using the `new` keyword.

container

A specialized component that contains and manages other components.

control object

A type of object that coordinates other objects, for example, to perform transactions on a set of objects that represent stored data.

CORBA

See [Common Object Request Broker Architecture](#).

CORBA object

An object abstraction, defined using an IDL interface, to which client requests are directed. A CORBA object can be located by its reference. Exactly how the CORBA object satisfies a request is transparent to the client.

CORBAServices

See [Common Object Services](#).

COS

See [Common Object Services](#).

CSMAP

The NonStop CORBA database whose entries determine which Comm Server (CS) each configured client will use. Entities in the configuration database that hold CS load mappings between remote clients and client/servers.

Current interface

An object class, defined by the Object Transaction Service (OTS), which gives the OTS client

control over application transactions. For example, the Current interface defines methods to mark the beginning of a transaction, to commit a transaction, to suspend a transaction, and to undo (roll back) a transaction.

data access objects (DAOs)

Objects that isolate process-blocking database access operations, which cannot take advantage of multithreading, from functions that can be implemented using multithreading. In the NonStop CORBA environment, data access objects can be implemented as context-free TS/MP servers, configured as server pools, that instantiate only one object per process and are created by a POA with a lifespan policy of PERSISTENT, a state policy of STATELESS, and a threading policy of SINGLE_THREAD_MODEL.

database mapping

See [mapping](#).

data type

A categorization of values, operations, and arguments that usually cover both behavior and representation.

delegation

A style of program design that isolates CORBA functions from non-CORBA functions.

deprecated

Possibly available as a feature for upward compatibility of existing applications, but not recommended for use by new applications.

derived class

An object class that inherits characteristics from another class.

digital certificate

See certificate.

direct TCP server

A NonStop CORBA server that is configured to communicate directly with TCP sockets.

Distinguished Name (DN)

The complete name of a directory entry, consisting of the Relative Distinguished Name (RDN) of the entry and the RDNs of its superior entries.

distributed object computing

A style of application in which object-based components distributed across processes or across a network communicate by using a message-based interface.

dynamic context

Information that must be retained in the memory of a process (as opposed to being saved on disk) between method calls.

Dynamic Invocation Interface (DII)

A request style in which the client constructs a request at runtime, not needing to know the object interface (the calling sequences for requesting specific operations) at compile time.

dynamic link library (DLL)

A way of packaging application functions that lets the functions be loaded into a process at runtime rather than at compile time. Such functions can be unloaded when they are no longer required in the application while the rest of the application continues to run.

Dynamic Skeleton Interface (DSI)

A way to 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. This contrasts with the type-specific, OMG IDL-based skeletons, but serves the same architectural role. DSI is the server side's analogue to the client side's Dynamic Invocation Interface (DII).

encapsulation

The localization of knowledge within a module. Because objects encapsulate data and implementation, the user of an object can view the object as a black box that provides services. Instance variables and methods can be added, deleted, or changed, but as long as the services provided by the object remain the same, code that uses the object can continue to use it without being rewritten.

encipherment

See encryption.

encryption

Generation of ciphertext from corresponding cleartext by application of a cryptographic algorithm and a key.

entity object

A type of object that represents external data, such as a database record or an external device.

environment parameter

A CORBA-defined variable included in method invocations to receive exception information.

environment structure

A CORBA-defined variable included in method invocations to receive exception information.

environment variable

An element of a system or product-specific configuration file used to specify the path to a processing component or to control some aspect of system or product operation. For example, you use an environment variable in the OSS profile file to specify the location of the NonStop CORBA configuration file.

EnvironServer

See [NSDEnvironServer](#).

esadmin tool

A development tool for debugging a NonStop CORBA application. While the esadmin tool can also be used to perform Event Service administrative and management tasks, the Console is the preferred tool for management in production environments.

event

(1) The link between the behavior that a component wants to react to and the code that tells the application how to respond. Java differentiates between the source of an event (the provider) and the handler for an event (what to do).

(2) In NonStop CORBA, an asynchronous message sent by one object to another.

(3) In Distributed System Management (DSM), an error or other occurrence of significance to the operator, described in a message from a software subsystem to the Event Management Service (EMS).

event handler

An object defined by the NonStop CORBA event framework to support asynchronous communication between a NonStop CORBA application process and other processes, such as legacy clients or servers. NonStop CORBA provides various types of event-handler objects for use in legacy server wrappers.

event handler user

An object that makes requests to an event handler on behalf of an application process.

Event Management Service (EMS)

A set of processes and interfaces through which NonStop products report significant events to a common location for subsequent distribution to operators and management applications. EMS supports filters through which the user can request messages from specific products or messages having specific content or characteristics.

Event Service

A standard application service (one of the OMG Common Object Services) for asynchronous communication among objects. See also [esadmin tool](#).

exception

An event during program execution that prevents the program from continuing normally; generally, an error. Java supports the `try`, `catch`, and `throw` keywords for exceptions.

exception handler

A block of code that reacts to a specific type of exception. If the exception is for an error that the program can recover from, the program can resume running after the exception handler has executed.

externalization service

A standard application service (one of the OMG Common Object Services) that supports the movement of object-state information from one process or medium to another using the abstract concept of a "stream." The sender performs any transformations required to write the data to the

stream; the receiver performs transformations required to process the data in its local environment.

factory object

An object that creates other objects. An application might place factory objects into the Naming Service. You must implement a factory as a static method.

fleshed-out script

The results of editing skeletons, for example, nsdstart or default.db. Once configured, the fleshed-out script can be used to start, stop, or uninstall an existing NonStop CORBA installation.

framework

A set of related object classes, technically different from a class library in that the framework makes requests of a user program. However, many software vendors use the terms "class library" and "framework" interchangeably.

General Inter-ORB Protocol (GIOP)

A required CORBA protocol that defines the format of the data units that can be sent by any transport. NonStop CORBA can support three versions of GIOP: 1.0, 1.1, and 1.2. Various differences exist between the versions, for example, the handling of `wchar` and `wstring` differs.

GIOP over Guardian file system

A proprietary transport protocol, provided by NonStop CORBA, that uses Guardian file-system addressing to access a specific instance of an object (that is, a stateful object). Guardian file-system addressing uses a process name to specify a particular server process.

GIOP over TS/MP

A proprietary transport protocol, provided by NonStop CORBA, that addresses stateless objects through server pools.

granularity

The complexity of an object; whether it consists of few or many discrete operations or data items.

host

Provide execution space for an object. A server can host instances of one or more object classes.

HOSTS file

A file that contains IP addresses and aliases representing systems known to NonStop TCP/IP.

identifier

(1) In IDL, an identifier is an arbitrarily long sequence of alphabetic, numeric, and underscore characters. The first character must be alphabetic. All characters are significant. Identifiers that differ only in case collide and yield a compilation error. An identifier for a definition must be spelled consistently (with respect to case) throughout a specification.

(2) In a C++ or Java program, the name of an item.

IDL compiler

See [Interface Definition Language \(IDL\)](#).

IDL specification

An OMG IDL specification consists of one or more type definitions, constant definitions, exception definitions, or module definitions. See also [Interface Definition Language \(IDL\)](#).

IIOP

Internet Inter-ORB Protocol. An object-oriented protocol that allows distributed programs written in different programming languages to communicate over the Internet. IIOP is a critical part of CORBA. Using CORBA IIOP and related protocols, a company can write programs that can communicate with other programs wherever located and without having to understand anything about the other programs except their services and names.

IIOP/SSL

Secure Sockets Layer Inter-ORB Protocol. This is the IIOP protocol over SSL. The NonStop CORBA implementation of IIOP/SSL is called SSLIOP.

ILSD

Interoperable Location Service Daemon. A component of the Interoperable Naming Service that services requests for name resolution in the URL (Uniform Resource Locator) format.

implementation

The data definitions and methods of an object class. A programmer produces the implementation from a skeleton emitted by the IDL compiler.

implementation binding

See [binding](#).

implementation language

The programming language used to create the source code for an object class.

Implementation Repository

A database that records information about object implementations and the servers in which they can reside.

indirect TCP server

A NonStop CORBA server that is configured to use a Comm Server to communicate with TCP. This configuration is useful with stateless servants, and for TCP connection concentration.

inheritance

The propagation of characteristics from a base (or parent) class to a subclass. More generally, the propagation of a characteristic from a data type to a subtype.

install host

The system, in a network of systems, on which you first install NonStop CORBA.

instance

An object of a particular class. For example, an object that represents a particular person's bank account is an instance of the object class that generically defines a bank account. In Java programs, an instance of a class is created by using the new operator followed by the class name.

interface

(1) In IDL, an interface is a description of a set of possible operations that a client may request of an object. An object satisfies an interface if it can be specified as the target object in each potential request described by the interface.

(2) In Java, a group of methods that can be implemented by several classes regardless of where the classes are in the class hierarchy.

Interface Definition Language (IDL)

A language specified by the OMG that describes the interface of an object in language-neutral terms. IDL is the means by which a particular object implementation tells its potential clients what operations are available and how they should be invoked. From the IDL definitions, the IDL compiler produces language-specific files according to CORBA-defined mappings between IDL and supported implementation languages.

interface object

An object that serves to describe an interface. Interface objects reside in an interface repository.

Interface Repository (IR)

(1) A database containing information about object interfaces: the data accessible through an object and the calling sequence for each operation the object supports.

(2) The component of the Object Request Broker that provides persistent storage of the IR definition. The IR definitions are maintained in the Interface Repository as a set of IR definition objects.

Interface Repository Daemon (IRD)

The Interface Repository Daemon (IRD) is a NonStop CORBA component that services requests for implementations stored in the Interface Repository. It starts by getting IR from the Naming Service and activating it with a persistent POA.

Interoperable Naming Service (INS)

The Interoperable Naming Service extends the Naming Service to allow access to an object through human-readable URL names.

Internet Inter-ORB Protocol (IIOP)

See [Inter-ORB Protocols \(IOPs\)](#).

Inter-ORB Protocols (IOPs)

Rules for transmitting messages among distributed systems. CORBA defines the Internet Inter-ORB Protocol (IIOP), which uses TCP/IP.

IOR

Interoperable Object Reference.

IP address

The type of network address associated with the TCP/IP protocol.

IPC

Interprocess communication. NonStop systems use a message-based style of IPC.

IR

See [Interface Repository \(IR\)](#).

jacket routine

A procedure or function call that hides a protocol, translating it into another protocol that can more readily be handled by a calling program. An example is the DCE_SERVERCLASS_SEND_ procedure, which translates a call to the process-blocking SERVERCLASS_SEND_ procedure into a thread-blocking call.

Java Transaction Service (JTS)

A special client stub, provided as part of NonStop CORBA, that implements the JTS API.

key

A variable that provides an index into a dictionary.

language binding or mapping

See [binding](#).

language independence

The characteristic that permits a client to use an object written in another language.

legacy client

A non-CORBA client.

legacy client wrapper

A NonStop CORBA application that acts as a legacy server, transforming legacy-system requests into CORBA method invocations, and transforming CORBA method completions into legacy-system replies.

legacy server

A non-CORBA server.

legacy server wrapper

A NonStop CORBA application that acts as a server, transforming CORBA method invocations into legacy-server requests, and transforming replies into CORBA method invocations.

legacy system

For NonStop CORBA, an existing system that is not based on the CORBA model.

lifecycle

The series of steps through which an object is created and deleted, activated and deactivated.

lifecycle service

A standard application service for creating and deleting objects. See also [Common Object Services \(COS\)](#).

link

A logical connection between processes. In NonStop TS/MP, the logical connection between a requesting process and a server in a server pool.

LINKMON process

The NonStop TS/MP process that obtains links for all requesters in a CPU.

listener event handler

An object, defined by the NonStop CORBA event framework to support asynchronous communication between a NonStop CORBA server and a client that uses other technology, that establishes and maintains connections between a legacy client and a NonStop CORBA server (such as a server event handler). Listener event handlers and server event handlers are used in legacy client wrappers.

listener event-handler user

An object, defined by the NonStop CORBA event framework, that uses the services of a listener event handler. A legacy client wrapper typically acts as both a listener event-handler user and a server event-handler user.

literal

A value that identifies an entity that is not an object.

Literals can be categorized as follows:

- Integer literals--one or more digits
- Character literals--one or more characters
- Floating-point literals--an integer, decimal point, fraction, an e or E, and an optionally signed integer exponent
- String literals--a sequence of zero or more characters enclosed within double quotes

load balancing

The distribution of network traffic or application workload across available lines or processes, respectively.

local/remote transparency

Also called location transparency. The characteristic that makes it unnecessary for a client to know whether an object it uses is local or remote.

Location Service Daemon (LSD)

A NonStop CORBA process that assigns a client to a Comm Server and looks up the addresses of direct TCP servers having persistent policy. See also [Comm Server](#).

makefile

A file that describes the dependencies among application modules that together constitute an executable program. The make utility (present in NonStop OSS and other implementations of UNIX or POSIX) interprets a makefile and initiates recompiling of changed and dependent modules.

mapping

In IDL, language mapping includes a definition of the language-specific data types and procedure interfaces to access objects through the ORB. Mapping includes the structure of the client stub interface (not required for object-oriented languages), the Dynamic Invocation Interface, the implementation skeleton, the object adapters, and the direct ORB interface. A language mapping also defines the interaction between object invocations and the threads of control in the client or implementation.

marshaling

When a client requests an object that resides on a remote host, a TCP/IP connection is established between the client and object server. Marshaling involves creating a proxy for the object on the local machine and a stub on the remote machine. The proxy communicates with the stub, handling all details. The server on the remote machine will demarshal the request, invoke the desired method, and send results back to the client.

The term *demarshal* refers to the processing of incoming messages for delivery to local objects and clients.

message

(1) In NonStop CORBA, a body of data intended for interprocess transfer by an event handler. A message object consists of a list of pointers to message data (MD) segments, which contain the actual data. The NonStop CORBA message object includes methods that applications can use to manipulate a message.

(2) In general object technology, information passed from one object to another: a method invocation results in a message to the target object, and the reply results in a message from the target object to the client. Physical movement of data does not necessarily occur.

message data (MD) segment

A logical subdivision of a message. A NonStop CORBA message object creates segments automatically as the application writes to a message. Alternatively, the application can explicitly create and operate on message data segments.

message factory

A NonStop CORBA object class that applications use to create and free messages and message data segments.

method

A Java function. An operation defined for an object, implemented as a procedure or function in a programming language. The calling sequence for a method is called a method signature. A call to a method is called a method invocation.

multi-tiered architecture

See [tiered application architecture](#).

name binding

Also known as name-object binding. An object consisting of a name and a binding.

name component

A structure containing a name-id string and a name-kind string.

naming context

An object that contains a set of name bindings in which each name is unique. To resolve a name is to determine the object associated with the name in a given context.

Naming Service

A standard application service (one of the OMG Common Object Services) for storing object names, identifiers, and possibly other information. One important use of the Naming Service is to store the names of factory objects to use in creating other objects. See also [ns browse](#).

narrowing

A term used in CORBA when moving from a base interface to a derived interface. Java refers to this action as downcasting.

nested transactions

Transactions that begin and end within another transaction.

network client

A client process that resides on some other system (not the same NonStop system or Expand network as the NonStop CORBA ORB). The network client process acts as a CORBA client. Network clients send IIOP requests for the services of an application through the NonStop CORBA communication services. Also referred to as a remote client.

NonStop CORBA event framework

The core of the NonStop CORBA run-time environment, which supports the handling of I/O events over various transport protocols. It provides APIs that are used internally by NonStop CORBA but can also be used in legacy server wrapper and legacy client wrapper programs. It is a low-level mechanism that does not require (but does enable) both communicating components to use CORBA.

NonStop Distributed Component Console

A tool that allows you to manage NonStop CORBA domains and services.

NonStop DOM/MP (NonStop" DOM)

A component of NonStop CORBA that implements a C++ programming environment. NonStop

DOM was replaced as a separate product by the NonStop CORBA product.

NonStop JORB/MP (NonStop JORB)

A component of NonStop CORBA that implements a Java programming environment. NonStop JORB was replaced as a separate product by the NonStop CORBA product.

NonStop TCP/IP

An HP product that implements the TCP/IP protocol. NonStop CORBA uses NonStop TCP/IP for communication with remote clients and servers.

NonStop TS/MP

NonStop TS/MP software manages server pools by establishing links between clients and servers, balancing the workload across servers, automatically creating and deleting servers in response to changes in request traffic, and restarting servers after CPU or process failures.

NSDAdminServer

A process that is the initial point of contact the NonStop Distributed Component Console uses to manage NonStop CORBA systems. This process supplies basic support services to the console. There is one NSDAdminServer per NonStop system.

NSDEnvironServer

A process that provides services for a particular security domain in a NonStop CORBA system. There are one or more of these processes in a NonStop CORBA system, depending upon how many security domains are configured.

ns_browse

A tool that allows you to view the Naming Service database as a tree of naming contexts.

NSJTS

The set of NonStop CORBA components that implement the Transaction Service and the Java Transaction Service (JTS) for Java programs. These components include the JTS client stub, the NSotsTM process, and the NSotsXID process.

NSOTS

The set of NonStop CORBA components that implement the Transaction Service for C++ programs. These components include the OTS client stub, the NSotsTM process, and the NSotsXID process.

NSotsTM

A process, provided as part of NonStop CORBA, that collaborates with the Transaction Management Facility (TMF) on the NonStop system to manage transactions and that can run as a server pool.

NSotsXID

A singleton process, provided as part of NonStop CORBA, that is used to broker transaction identifiers (XIDs) when NSotsTM runs as a server pool with multiple processes.

object

The principal building block of object-oriented programs. An object is an instance of an implementation and an interface. Each object is a programming unit consisting of data and methods or functions. See also [CORBA object](#).

object adapter

A CORBA component, residing in a server, which provides an interface between the Object Request Broker and the objects running in a server. CORBA defines a [Portable Object Adapter \(POA\)](#).

Object by Value

Object by Value (OBV) is a way of passing objects and parameters by value rather than by reference. The receiving side of an object passed by value receives a description of the "state" of the object. It then instantiates a new object with that state but having a separate identity from that of the sending side.

object class

A collection of data and the operations that can be performed on the data.

object ID

A value used to identify a CORBA object. The value is unique only within a POA scope. An object ID is not guaranteed to be unique with respect to a server because there could be more than one POA object at any time on a given server. An object ID is opaque and is not visible to the client.

object identity service

A service that distinguishes among instances of the same object class, even in cases where the data values are identical.

object lifecycle

The stages in the existence of an object, such as its creation, its location through use of the Naming Service, and its release.

Object Management Group (OMG)

The vendor consortium responsible for defining the Common Object Request Broker Architecture (CORBA) and other standards.

object-oriented

A software design method that models the characteristics of abstract or real objects using classes and objects.

object reference

A value that precisely identifies an object. Object references are never reused to identify another object.

Object Request Broker (ORB)

A component that transmits messages between clients and servers in a distributed object

environment. The ORB provides services such as locating the target object and performing any transformations required for transmission of a request or receipt of a response across the network. See also [marshaling](#).

object services

See [Common Object Services](#).

Object Transaction Service (OTS)

(1) Former name for the OMG Transaction Service. See [transaction service](#)

(2) A special client stub, provided as part of NonStop CORBA, that implements the OMG Transaction Service API.

OBV

See [Object by Value](#).

object wrapper

The result of encapsulating services from a non-CORBA (or legacy) application or program interface to treat the encapsulated application or interface as an object. See also [wrapping](#).

OMG

See [Object Management Group](#).

Open SSL

Open SSL is a project to develop an open-source toolkit that implements the Secure Sockets Layer (SSL) and Transport Layer Security (TLS) protocols. It also includes a general purpose cryptography library.

Open System Services (OSS)

A set of POSIX-compliant operating-system interfaces for the HP NonStop Kernel.

operator

A language symbol that represents an arithmetic operation or logical operation or relationship.

ORB

See [Object Request Broker](#).

osh

The utility through which you invoke any other utility in the Open System Services (OSS) environment. In UNIX terminology, this type of utility is called a shell.

OSS

See [Open System Services](#).

OTS

See [Object Transaction Service \(OTS\)](#).

overloading

In C++, the redefinition of a language element to provide for different behavior depending on context. For example, the addition operator (+) could be overloaded to support concatenation of text strings. In Java, using one identifier to refer to multiple items in the same scope. You can overload methods but not variables or operators.

overriding

Providing a different implementation of a method in a subclass of the class that originally defined the method. For example, if the base class defines a method called report, which produces a report in a specific format, a subclass can override (redefine) that method so a report request to an object of the subclass will produce a report in a different format.

package

A group of related Java classes. Also, a Java keyword that defines the package that a class belongs to.

parallelism

A means of improving performance, and potentially fault tolerance, by allowing multiple processing tasks to proceed simultaneously. Threads and server pools are two mechanisms for achieving parallelism.

Parallel Library TCP/IP

An HP networking product that provides increased performance, scalability, and fault-tolerance by creating parallel TCP/IP protocol stacks implemented as a shared runtime library in the NonStop Kernel operating system. A multiprocessor NonStop system using Parallel Library TCP/IP can be configured to appear as a single IP host to the outside world.

parent class

A class from which some other class (a subclass) is derived, so that the subclass inherits characteristics from the parent class and also defines characteristics of its own.

partitioning

The distribution of data across files and storage media, the distribution of application logic across processes, or the distribution of objects among server processes.

PATHCOM

The Command Line Interface (CLI) utility used to define and manage TS/MP server pools and other components of NonStop TS/MP.

PATHCOM configuration file

A file that defines processes and operational parameters for a PATHMON environment (the set of processes under control of a single PATHMON process).

PATHMON

The NonStop TS/MP process that starts, restarts, and monitors server processes, establishes links between requesters and servers, and balances loads. The set of processes managed by an instance

of PATHMON is called a PATHMON environment.

Pathsend facility

The set of procedures a requester uses to exchange requests and replies with a NonStop TS/MP server pool. (The actual procedure call to send a messages can have a different name, such as SERVERCLASS_SEND.) NonStop CORBA uses the Pathsend facility internally to support invocations of objects running in NonStop TS/MP server pools. A client program can also use the facility explicitly to send a non-CORBA message to a server pool that is not CORBA compliant. (Jpathsend)

pax

A utility for extracting a set of files that have been archived as a single file. pax is available on UNIX and POSIX systems, including HP systems using Open Systems Services (OSS).

persistent identifier (PID)

A value that relates an object to the external entity (for instance, a database record) that the object represents.

persistent object

An object that must be accessible after the process that hosted it has stopped.

POA

See [Portable Object Adapter \(POA\)](#).

POA manager

A server-side object that contains the processing state for one or more POAs. Developers can use the POA manager to process, queue, or discard requests. The manager can also activate or deactivate POAs.

policy

A policy specifies a POA characteristic behavior for objects implemented in that particular POA. For example, a policy can determine threading, activation, object ID assignment, and so on. Every POA has default policies if no other one is specified.

polymorphism

The characteristic of object technology that allows a client to make the same request of different objects and assume that the operation will be performed appropriately for each class of object. For example, a display operation on a video object does not result in the same behavior as a display operation on a text file, but the client treats both in the same way.

port number

An addressing component that associates multiple local destinations with a single network (IP) address. For example, a TCP/IP process listens on one IP address but can route requests to different local processes based on port number.

Portable Interceptor

Interceptors intercept the flow of a request-reply sequence. Request interceptors are used to enable

ORB services to transfer context information between clients and servers. IOR interceptors are used to establish tagged components in the profiles within an IOR.

Portable Object Adapter (POA)

The CORBA-specified object adapter that connects an object residing on a server with an ORB. The adapter provides a namespace within which the client and the server can connect and exchange information.

porting

Rewriting or modifying an existing component to run on a different system, language, or platform.

POSIX

A standard set of operating-system interfaces similar to UNIX. The HP NonStop Kernel Open System Services (OSS) product provides POSIX interfaces.

privacy

The property that information is not made available or disclosed to unauthorized individuals, entities, or processes.

private key

In a public key cryptography system, the component of a key pair that is not divulged by its owner.

process-blocking operation

Operations that block the calling process while they are in progress. Examples are accesses to NonStop SQL/MX and NonStop SQL/MP databases. Application processes that must perform process-blocking calls should not be multithreaded.

profile

(1) For IORs, tagged profiles are elements of the IOR containing the profile's tag, address information, and target object key. There is no limit on the number of tagged profiles in an IOR.

(2) For NonStop CORBA processes that use CORBA::ORB, such as the Name Service or LSD, the <profile>@ORB entity holds the ORB's configuration.

profile file

The configuration file that defines the operational parameters for Open System Services (OSS).

proxy

A local reference to a remote object, used in such a way that an operation invoked on the proxy is performed on the remote object.

pthread API

NonStop CORBA uses the DCE pthread API, which is the basis for the object-oriented vthread API.

Ptrace

A utility program that produces a formatted display of SCF trace information.

public key

In a public-key (asymmetric) cryptosystem, the component of a key pair that is revealed.

QIO

An HP product that uses shared memory segments to optimize the performance of its client process. QIO stands for Queued Input/Output.

reentrant code

See [thread-safe code](#).

reuse

The principle of using a software component in multiple applications, for example by deriving application-specific object classes from general-purpose object classes, or using existing class libraries or frameworks for common application functions. Reuse is one of the goals of object technology.

RMI

Remote Method Invocation. A distributed object model for one Java virtual machine to invoke methods on an object running in a different Java virtual machine, possibly on different hosts.

RMI-IIOP

A version of Remote Method Invocation (RMI) implemented to use the CORBA IIOP protocol. RMI over IIOP provides interoperability with CORBA objects implemented in any language if all the remote interfaces are originally defined as RMI interfaces.

roll back

Undo (a transaction).

Root POA

The POA provided by the ORB to which all other POAs are related.

scalability

The characteristic that allows a system to grow to accommodate increased usage. On NonStop systems, scalability is achieved largely through parallel processing.

Secure Sockets Layer (SSL)

A secure interoperability protocol that works at the communications transport layer.

Secure Sockets Layer Inter-ORB Protocol (SSLIOP)

See IIOP/SSL.

security domain

The processes and configuration data that are manageable under a given user ID.

security policy

The data that defines what protection a system's security services must provide. There are many

kinds of security policies, including access control policy, audit policy, message protection policy, non-repudiation policy, etc.

security service

An application service that verifies user identity as a means of restricting certain operations to users who have proper identification (authentication).

sequence

An IDL data type that is essentially a one-dimensional array of specified maximum size and length. IDL sequences can be bounded or unbounded. They are mapped to a Java class that has a current length. In C++ they are mapped to a class that has a current length, a maximum length, and a `_var` type for the data.

servant

An instance of a CORBA object. A servant resides in a server and handles and responds to a particular request from a client.

servant manager

A POA component that acts on demand from the ORB to activate or deactivate servants. The servant manager associates an ORB object with the appropriate servant. The two types of managers are `ServantActivator` and `ServantLocator`. POA policies determine which manager is used.

server

In distributed object technology, an application component that satisfies the requests of a client by providing an execution space (hosting) one or more objects invoked by the client. More generally, an application component that services the requests of a client. A server program is the source code for a server.

server class

A NonStop TS/MP term synonymous with a [server pool](#).

server event handler

An object, defined by the NonStop CORBA event framework to support asynchronous communication between a NonStop CORBA server and a client that uses other technology, that transfers data across a connection established by a listener event handler. Server event handlers and listener event handlers are used in legacy client wrappers.

server event-handler user

An object, defined by the NonStop CORBA event framework, that uses the services of a server event handler. A legacy client wrapper typically acts as both a server event-handler user and a listener event-handler user.

server pool

A set of processes that execute the same program in parallel to increase the throughput and fault-tolerance of an application. A NonStop CORBA application can use server pools managed by NonStop Transaction Services/MP (NonStop TS/MP).

server selection

The process of choosing a server process to handle a request. In NonStop CORBA stateless processing, each request can be handled by any server that can host the object interface. In stateful processing, a specific server is chosen to handle all of a series of requests.

service

A set of related functions available to applications on a system or network. See also [Common Object Services](#).

setting

In general, the value of a parameter or variable. In NonStop CORBA, a value specified in a configuration file.

shared runtime library (SRL)

An object file that the operating system links to a program at runtime.

showior

A tool that produces output that can be used to track down problems in NonStop CORBA configurations.

skeleton

Entity residing on the server. A skeleton is a base class containing pointers to servant methods. The base class is generated by the IDL compiler. The skeleton connects the servant to an object adapter.

slot

The portable interceptors `PICurrent` interface is a slot table. The slots are used by each service to transfer context data between the service's context and the requestor's or reply's service context. Each service that wishes to use `PICurrent` reserves a slot or slots at initialization time.

socket

A logical connection to the TCP/IP process.

SSLIOP

Secure Sockets Layer Inter-ORB Protocol. SSLIOP is the NonStop CORBA implementation of IIOP over Secure Sockets Layer.

stateful processing

A style of application logic in which all operations directed to the same object (for the duration of a link acquired by the client) must be handled by the same servant. Operations directed to other objects in the class can be handled by different servants in other processes. A stateful object can maintain state information (data associated with a particular object instance) in memory between operations, because all requests for that reference go to the same instance. With stateful objects, a client's object must always reference the specific instance of that object. Stateful objects are created by a POA whose lifespan policy is `TRANSIENT`, such as the root POA, or by explicitly setting the NonStop CORBA proprietary state policy to `STATEFUL`. Stateful processing is the

default behavior for objects as specified by the root POA.

stateless processing

A style of application logic in which operations directed to the same object can result in requests to different servants, which in turn can reside in different processes in a server pool. A stateless object cannot maintain state information in memory between operations, because the state information would be available to only one process. A client reference can address any of a set of instances. Stateless objects are accessible only when hosted in a server pool. Stateless processing is selected by using a lifespan policy of PERSISTENT (or by setting the NonStop CORBA proprietary state policy to STATELESS) for the POA that creates the object reference, and by enabling the GIOP over TS/MP protocol in the server.

Static Invocation Interface (SII)

A request style in which the client must know the object interface (the calling sequence for requesting specific operations) at compile time. Contrast [Dynamic Invocation Interface](#).

Structured Query Language (SQL)

A standard technology for defining and using relational databases. NonStop SQL/MP is an HP implementation of this standard.

subclass

An object class that inherits characteristics from another object class.

subnet

A concept that applies to TCP/IP, which defines the relationship between a TCP/IP process, one or more TLAM processes, and one or more Internet (IP) addresses.

Subsystem Control Facility (SCF)

The interactive utility for configuring, monitoring, and controlling networking resources on a NonStop system.

Subsystem Programmatic Interface (SPI)

A set of conventions that provide a uniform way to automate the management of many products on a NonStop system. In general, a management programming interface based on SPI makes available the same set of services available to an operator through the corresponding interactive utility. For example, NonStop TS/MP offers a management programming interface that gives applications access to the same set of functions available through PATHCOM.

TCP/IP (Transmission Control Protocol/Internet Protocol)

One of the most widespread networking protocols, the basis for communications on the Internet. The CORBA IIOP protocol uses TCP/IP as its transport service. NonStop TCP/IP is the HP product that implements TCP/IP on NonStop systems.

thread

The basic unit of program execution. A process can have several threads running concurrently, each performing a different job, such as waiting for events or performing a time-consuming job that the program doesn't need to complete before going on. When a thread has finished its job, the

thread is suspended or destroyed.

Threads differ from [server pools](#) in that threads require the same process to handle multiple requests, whereas server pools allow requests to be handled by different processes running in parallel.

thread-blocking operation

Operations that block only the calling thread, rather than the entire process, while they are in progress. Examples are nowait file I/O operations and operations directed to distributed objects.

thread-safe code

Code in a multithreaded program that protects an object from access by more than one thread at a time. Mechanisms to provide thread-safe code include mutexes in C++ and monitors in Java.

three-tiered architecture

See [tiered application architecture](#).

tiered application architecture

An application model in which function is divided into a few major categories of objects, such as interface, control, and entity objects.

TMF

See [Transaction Management Facility \(TMF\)](#).

trace

A sequential record of data or program flow, useful for troubleshooting. NonStop CORBA includes facilities to trace the flow of control within an application.

trace facility

An object API that writes trace messages to an ASCII file. The trace facility can be used to implement instrumentation in NonStop CORBA applications. You can add trace statements to specific portions of NonStop CORBA applications or components to help troubleshoot program code.

transaction

A series of logically related operations. A transaction is complete only when all operations in the series have been performed. This manual makes a distinction between user or application transactions, which are defined and controlled by application components, and services transactions, which are defined and controlled by objects services such as the Naming Service.

transaction management

A set of tasks required to ensure satisfactory transaction processing. Transaction management includes transaction control (the application logic that determines when a transaction starts and stops, and when and how to abort a transaction), transaction protection, and auditing (logging the progress of transactions to support request tracking and recovery). A NonStop CORBA application controls transactions by using the Object Transaction Service (OTS).

Transaction Management Facility (TMF)

The facility that manages transactions on NonStop systems. TMF collaborates with the transactional components of NonStop CORBA.

transaction protection

A feature that ensures consistency of databases and other stored data.

Transaction Service

A standard application service (one of the OMG Common Object Services) for ensuring the consistency of a database or other stored information when a request involves multiple updates to databases or state repositories.

Transport Layer Security (TLS)

A secure interoperability protocol from the Internet Engineering Task Force that works at the communications transport layer. It is a successor to the Secure Sockets Layer (SSL) protocol.

trustworthiness

The ability of a system to protect resources from exposure to misuse through malicious or inadvertent means.

usage binding

See [binding](#).

vthread API

An object-oriented, portable API for multithreading provided by NonStop CORBA for use in C++ programs. (For Java programs, standard Java threading is recommended.) This API is implemented as a set of jacket procedures built upon the NonStop pthread API.

wrapping

A technique in which a component is given an interface that permits interoperation with components based on a different technology. For example, a set of interacting Java object classes can be wrapped in a single NonStop CORBA class, or the code of a legacy server can be used as the implementation of an object supporting methods that correspond to functions of the server.

A recommended approach to wrapping is to develop a wrapper component that translates method invocations to some other message format, or vice versa, to allow communication between object-oriented clients and servers and legacy clients and servers.

widening

A term used in CORBA when moving from a derived interface to a base interface.