WebSphere Application Server Enterprise Edition
Component Broker

Glossary

Version 3.0

SC09-4450-00

WebSphere Application Server Enterprise Edition
Component Broker

Glossary

Version 3.0

SC09-4450-00

Note!
Before using this information and the product it supports, be sure to read the general information under "Notices" on page 61.

Fifth Edition (August 1999)

This edition applies to:

WebSphere Application Server Version 3.0, Enterprise Edition Development Toolkit for AIX, program number 5765–E27
WebSphere Application Server Version 3.0, Enterprise Edition Development Toolkit for Windows NT, program number 5639–I07
WebSphere Application Server Version 3.0, Enterprise Edition for AIX, program number 5765–E28
WebSphere Application Server Version 3.0, Enterprise Edition for Windows NT and Gradient DCE, program number 5639–I08
WebSphere Application Server Version 3.0, Enterprise Edition for Windows NT and IBM DCE, program number 5639–I09
WebSphere Application Server Version 3.0, Enterprise Edition Encina Client for Windows NT/Windows 95, program number 5639–I10
WebSphere Application Server Version 3.0, Enterprise Edition MQSeries Application Adapter, program number 5639–I11
WebSphere Application Server Version 3.0, Enterprise Edition Oracle Application Adapter, program number 5639–I12
WebSphere Application Server Version 3.0, Enterprise Edition CICS/IMS Application Adapter, program number 5639–I13

and to all subsequent versions, releases, and modifications until otherwise indicated in new editions. Consult the latest edition of the applicable system bibliography for current information on these products.

Order publications through your IBM representative or through the IBM branch office serving your locality.

Contents

Component Broker Glossary

This glossary defines terms and abbreviations that are used in the Component Broker documentation. Some of the terms and definitions are from the following sources:

- *American National Standard Dictionary for Information Systems*, ANSI. X3.172-1990, copyright 1990 by the American National Standards Institute (ANSI). Copies may be purchased from the American National Standards Institute, 1430 Broadway, New York, New York 10018.
- *IBM Dictionary of Computing*: *Draft*, 1999.
- *IBM MQSeries Application Programming Guide*, SC33-0807.
- X/Open CAE Specification. Commands and Utilities, Issue 4. July, 1992.
- ISO/IEC 9945-1:1990/IEEE POSIX 1003.1-1990.
- *The Information Technology Vocabulary*, developed by Subcommittee 1, Joint Technical Committee 1, of the International Organization for Standardization and the International Electrotechnical Commission (ISO/IEC JTC1/SC1).

A

AA See application adaptor.

abstract class
(1) A class with at least one pure virtual function that is used as a base class for other classes. The abstract class represents a concept. Classes derived from it represent implementations of the concept. You cannot construct an object of an abstract class. See base class. (2) A class that allows polymorphism.

abstract interface
An interface that is introduced with the programmer's expectation that the interface will be merged (as a mixin) with another interface. An abstract interface cannot be implemented as is without being further derived at an interface level.

access control information (ACI)
Defined by ISO/IEC 10181-3 as privilege attributes (initiators) or control attributes (targets) that govern access to an object by a principal.

access control list (ACL)
In computer security, a list associated with an object that identifies all the subjects that can access the object and their access rights; for example, a list associated with a file that identifies users who can access the file and their access rights to that file.

access policies
The rules that define whether a principal can perform a particular operation on a particular object.

ACI See access control information.

ACID Acronym for atomicity, consistency, isolation, and durability; the fundamental properties of a Transaction Service. Atomicity, isolation, and durability lead to the consistency of data. See atomicity, consistency, isolation, and durability.

ACL See access control list.

activate
To prepare a persistent object to execute an operation.

ActiveX client
The ActiveX client provides OLE/COM mapping to CORBA, enabling a Component Broker managed object to be used in an ActiveX environment by a managed object proxy. The functions of the ActiveX interface are primarily mapped to OLE/COM. Although this client has elements of the CORBA solution, the interface offered to developers is distinctly COM. Therefore, this is not a true CORBA client.

activity
In FlowMark, a step within a process. It represents a piece of work that an assigned person can complete by starting a program or another process. See program activity and process activity.

activity log
The log file used mainly for storing information about normal Component Broker run-time program execution. This includes informational messages that show a history of activity of Component Broker services. Another use of the activity log is for storing CORBA exceptions that are returned. There is one activity log for all Component Broker processes on each workstation running the Component Broker run-time environment.

administrative interface
The interface of an object that defines its administrative and system management behavior. Typically, the administrative interfaces of an object are only used by system management and administration programs.

Advanced Interactive Executive (AIX)
IBM's implementation of the UNIX (trademark of AT&T Bell Laboratories) operating system. The RISC System/6000 system, among others, runs the AIX operating system.

agent (1) In systems management, a user who, for a particular interaction, has assumed an agent role. (2) An entity that represents one or more

managed objects by either emitting notifications regarding the objects, or handling requests from managers for management operations to modify or query the objects. (3) A system that assumes an agent role.

aggregate type
A user-defined data type that combines basic types (such as char, short, and float into more complex types (such as structs, arrays, strings, sequences, unions, or enums).

aggregation
The process of forming an aggregate object using other objects as its component parts.

AIX See advanced interactive executive.

API See application programming interface.

applet An application program, written in the Java programming language, that can be retrieved from a Web server and executed by a Web browser.

application
(1) The use to which an information processing system is put; for example, a payroll application, an airline reservation application, a network application. (2) A collection of software components used to perform specific types of user-oriented work on a computer. (3) A complete, self-contained program that performs a specific function directly for the user. This is in contrast to system software such as the operating system kernel and server processes and libraries that exist to support application programs. (4) A program that runs as part of a business application package. Besides the program, the application also has DLL files, classes, and other objects, and makes use of homes, containers, and other objects. For system management, the application package is represented by an application family install object and the application is represented by application model, application install, and application image objects.

application access policy
The mechanisms built into an application to control access to resources that it contains. Application access policies are enforced within the application implementation, although they can also be enforced with the assistance of a Security Service for acquiring principal credentials. See object invocation access policy.

application adaptor (AA)
(1) A database which is targeted to a specific backend system and contains the API and access information to build and decompose contacts to and from that backend system. (2) A location for instances of managed objects that is responsible for providing system capabilities for its managed object instances.

application adaptor mixin
A special object provided to a business object by an application adaptor. The mixin object provides an implementation of various interfaces that are inherited in the CB Server environment.

application data
Valid data that conforms to the application meta model. The application data, also called XML instance data, is contained in an XML file.

application model
A model that describes in detail the steps to be taken to accomplish the purpose of the application. This model is usually defined in a DTD file.

application object
An object that uses one or more business objects to perform some application function. An application object exists in a DLL that is loaded on the Component Broker server. Application objects implement server applications that are accessed from clients, and concepts that are defined by object-oriented analysis and object-oriented design.

application programming interface (API)
A software interface calling convention by which an application accesses an operating system and other services. An API is defined at source code level and provides a level of abstraction between the application and the kernel (or other privileged utilities) to ensure the portability of the code.

atomicity
Each transaction is a separate entity that either succeeds or fails as a unit. Each transaction must be capable of being committed (completed) on successful termination, or rolled back (backed out) on failure. If a transaction fails to complete, there is no danger of it executing partially and corrupting data. Most transaction services (including the Component Broker Transaction Service) provide atomicity using a process known as two-phase commit.

attributes
(1) In the CORBA IDL programming model, an identifiable link or association between the object and some other entity (for example, object, data value) that describes the object. (2) In the Java language, attributes are equivalent to fields that generally refer to data that belongs to the class or to the objects of the class.

audit identity
The identity of a principal used to audit that principal's actions in the security system. Typically, a principal's audit identity is anonymous to the principal.

Audit Security Service
A service that enables system managers to monitor Object Request Broker events, including logons and the names of servers and objects that are being used.

authentication
(1) In computer security, verification of the identity of a user or the user's eligibility to access an object. (2) In computer security, verification that a message has not been altered or corrupted. (3) In computer security, a process used to verify the user of an information system or protected resources. (4) The process of assuring that a principal is authentic. There are several ways to perform an authentication, which generally depend on something the principal knows (such as a secret or password), something the principal has (such as a badge or door key), or something the principal is (such as a signature, finger-print, voice-print, or retina-pattern).

B

bag In FlowMark, a container for data structures and programs. Programs contain data from the FlowMark program registrations as well as data required to enable FlowMark to invoke a method of a managed business object from CB Server.

base class
A class from which other classes are derived. A base class may itself be derived from another base class. See abstract class.

basic object adaptor
A component of the ORB that exists on each CB Server. It analyzes each request received by the ORB and dispatches it to the object implementation that is the target of the request.

bean-managed persistence (BMP)
Persistence that is managed by an entity bean.

bean-managed persistence (BMP) entity bean
Entity beans that manage their own persistence. Compare with container-managed persistence (CMP) entity bean.

behavior
The methods that an object responds to. These methods are either introduced or inherited by the class of the object. See state.

bind file
In DB2, a file produced by the precompiler when the command line

processor's BIND command is used. This file includes information on all SQL statements in the application program.

bind policy
The function used to define how to rank servers in a server group for selection for an object request. The bind policy includes a definition of the class and DLL files that implement the policy.

BMP See bean-managed persistence.

BOA See basic object adaptor.

BOIM See business object application adaptor and application adaptor.

business object
(1) An object that represents a business function. Business objects contain attributes that define the state of the object, and methods that define the behavior of the object. A business object also has relationships with other business objects. It can cooperate with other business objects to perform a specific task. Business objects are independent of any individual application. They can be used in any combination to perform a desired task. (2) In Component Broker, a business object functions as part of a component, and includes the business object interface, and the business object implementation.

business object application adaptor
A default application adaptor provided by Component Broker.

business process modeling
A process by which end users and domain experts begin the Component Broker application development process by developing a detailed, abstract model of the business function, roles performed by employees, and business operations. Business process modeling defines the application requirements and the functions that must be performed and performs use-case analysis. In addition, business process modeling provides input for object-oriented analysis, class modeling, design modeling, and workflow modeling.

C

capabilities
A set of object groups and associated rights. Each principal has an associated capability list, that is managed with the sec_edit utility. Capabilities are retrieved by clients through the security server, and then used to grant access.

cascaded invocation
A running method that executes in the server. The server becomes a client for the lifetime of the cascaded invocation. Security information must indicate the client making the request, as well as any servers in the cascade chain.

CASE See computer-aided software engineering.

casted dispatching
A form of method dispatching that uses casted method resolution.

CBToolkit
See Component Broker Toolkit.

CBRuntime
A solution that consists of server component services and subsystems that are responsible for executing new and pre-existing distributed applications in your enterprise computing environment. CBRuntime supports a wide variety of underlying resource managers which support the execution of business objects created using the WebSphere programming model.

CDM See common data model.

CDS See Cell Directory Service.

CDSPI
See cell directory service programming interface.

Cell Directory Service (CDS)
A Distributed Computing Environment (DCE) component that provides the ability to assign a set of attributes to a name structure in a directory hierarchy. It is used primarily within DCE to store remote procedure call bindings, but its use is not limited to this. Contrast with Global Directory Service.

cell directory service programming interface (CDSPI)
An internal interface used to program to the Cell Directory Service.

cell model
The type of model object that defines a logical grouping of host computers that are represented by host models. This allows system management to manage as a unit a number of host computers where the hosts are logically related, typically by the business or geographical area that they support. On one of the hosts in the cell there is a server that provides a dedicated naming service for the cell. That naming server resolves all requests by name for hosts, servers, and other objects in the cell. Hosts in a cell can also be members of a workgroup. You can have an enterprise that comprises a number of cells that group hosts similar to Windows NT, or DCE cells for general administration and several workgroups that group the same hosts into business areas for administration along business lines.

CGI See common gateway interface.

CICS See customer information control system.

CICS transaction
In CICS, a unit of application processing, usually comprising one or more units of work.

class (1) A group of objects that share a common definition and therefore share common properties, operations, and behavior. (2) A C++ aggregate that may contain functions, types, and user-defined operators in addition to data. Classes can be defined hierarchically, allowing one class to be an expansion of another, and classes can restrict access to their members. (3) An information set required to create and manage an object.

class library
A library of reusable classes for use with an object-oriented programming system.

client (1) A computer system or process that requests the services of another computer system or process. For example, a workstation that requests the contents of a file from a file server is a client of the file server. (2) In MQSeries, a run-time component that provides access to queuing services on a server for local user applications. The queues used by the applications reside on the server.

client application
In MQSeries, an application, running on a workstation and linked to a client, that gives the application access to queuing services on a server.

client code
(1) In Component Broker, an application program that invokes methods on objects that are instances of Component Broker classes. (2) In the Object Request Broker, a program that invokes a method on a remote object.

client enablement functions
Functions that allow non-ORB client applications to use the Component Broker server through an ORB.

client programming model (CPM)
A model that explains how application developers create objects that are clients of Component Broker application objects and business objects. Application developers can use the CPM and the Component Broker application development tools to build client and server applications whenever they use a managed object to implement a new object.

client/server
A common form of distributed system in which software is split between server tasks and client tasks. A client sends requests to a server, according to some protocol, asking for information or action,

and the server responds. There can be either one centralized server or several distributed ones. This model allows clients and servers to be placed independently on nodes in a network, possibly on different hardware and operating systems appropriate to their function.

CMP See container-managed persistence.

common data model (CDM)
A template of the CB Server configuration data structure that describes the structure of configuration data for objects managed by system management. The CDM is comprised of folders of objects that form a hierarchical tree structure. This model describes the hierarchy that can exist; the attribute names, types, limits, and default values of objects in those folders; and the relationships that can exist between objects. The CDM is composed of three conceptual parts, called "worlds" that include the model world, the install world, and the image world.

common data store
A Component Broker container that stores a portion of the configuration data held by a System Management agent. This data contains a portion of the definition type data that Component Broker needs access to, and is held for administrative convenience. The common data store holds this configuration data rather than allowing the data to be stored in the internal storage mechanism of the System Management agent.

common data store interface
The interface used for access by both System Management agents and components.

common gateway interface (CGI)
A standard for running external programs from a World-Wide Web HTTP server. The CGI specifies how to pass arguments to the executing program as part of the HTTP request. It also defines a set of environment variables. Commonly, the program will generate some HTML code that is passed back to the browser, but it can also request redirection to a different URL.

Common Object Request Broker Architecture (CORBA)
A specification produced by the Object Management Group (OMG) that presents standards for various types of Object Request Brokers (such as client-resident ORBs, server-based ORBs, system-based ORBs, and library-based ORBs). Implementation of CORBA standards enables ORBs from different software vendors to interoperate. See ORB.

Compare and Merge Tool for XML
A tool to compare (based on node identification) the XML files generated from project models, then merge them.

complex attribute
One that is made up of multiple entities. For example, a structure such as Address which is composed of street, city, country, and zip strings.

component
(1) A collection of related objects that work together to represent the logic and data relationships of the business function. (2) In Object Builder, an object that is composed of the following objects: business object interface, business object implementation, data object interface, data object implementation, persistent object and schema, key, copy helper, and managed object. See Component Broker component.

Component Broker
A run-time, development, and server management middleware product that supports developing and executing distributed object-oriented applications in a client/server environment. Component Broker provides the VisualAge Component Development package to develop distributed applications, the CB Server to execute the applications, and System Management for component administration. The purpose of Component Broker is to provide you with a complete solution for developing, executing, and managing distributed applications while leveraging pre-existing enterprise data, business logic and applications.

Component Broker component
A number of related objects that provide Component Broker functionality for a class. A class designed in Rational Rose, for example, becomes a component in Object Builder that typically consists of the following objects: business object interface, business object implementation, data object interface, data object implementation, key copy helper, and managed object. The inheritance structure of objects in a component mirrors the inheritance structure of the original class in Rational Rose. For example, if the class PolicyHolder inherits from the class Policy, then in Object Builder the PolicyHolder managed object inherits from the Policy managed object, the PolicyHolder key inherits from the Policy key, and so forth.

Component Broker Object Model
A logical, client/server object model that can be implemented in either a single or multiple tier application design space. The tiers that make up a typical client/server application design space are servers for objects that keep together client code and presentation, business logic, and supporting frameworks and object services. This object model

supports construction and implementation of JavaBeans objects and Enterprise JavaBeans objects as well as C++ objects. The object model includes the concepts of managed objects and a reuse model for separation of business logic and implementation specifics. The object model is incorporated into the structure of the VisualAge Component Development package.

Component Broker server
A tier-two object server that supports development, run-time, and management dimensions of Component Broker. It is also a container where the run-time components, the managed object framework for building business objects, the instance manager framework for adding support for additional resource managers, and the instance manager implementations that integrate existing resource managers into the server are created.

Component Broker Toolkit (CBToolkit)
Obsolete term for VisualAge Component Development package.

composed business object
An object that consists of multiple basic business objects. A composed business object is the combination of information and actions from different back end systems to provide new views and uses for existing systems.

composite business object
A business object based on a composition. It acts as a normal business object in a component, except calls to attributes and methods are delegated to its composition.

composite component
A component that consists of a composite business object, composite key, normal data object, and normal managed object.

composite group
See composition.

composite key
The key for a composite business object. See composite business object and key object.

composition
An object that provides a single interface to a number of components. A composite business object is based on a composition. Calls from composite business objects are delegated to the components that make up the composition.

compound name
A sequence of simple names that represent a traversal path through a name tree relative to some starting point. Contrast with simple name.

computer-aided software engineering (CASE)
The use of computerized tools for the collection and transformation of application domain information necessary during software construction.

Concurrency Service
The object service that provides a way to coordinate multiple, concurrent access to resources so that those resources are kept in a consistent state. When several clients try to use the same resource, the Concurrency Service serializes the access by using locks on the resource. The Concurrency Service grants locks to the different clients in such a way that there are never conflicting locks on a resource. The Concurrency Service is intended primarily for use in a transactional environment. See lock.

configuration
(1) The manner in which the hardware and software of an information processing system are organized and interconnected. (2) The devices and programs that make up a system, subsystem, or network. (3) In CCP, the arrangement of controllers, lines, and terminals attached to an IBM 3710 Network Controller. Also, the collective set of item definitions that describe such a configuration. (4) A folder that contains the models that define the servers, clients, and applications that you want to manage as one implementation of a management zone. Typically, a management zone may have several configurations; for example, to respond to regular changes in business needs. Within a configuration, each server that can be started is defined by a server model. The applications that can be used by those servers are defined by application models to which the server models are related. Other model objects define other things that are required in the management zone; for example, DLL files and types of clients.

consistency
Transactions are consistent when they move business data from one logically consistent state to another. When repeated, they produce predictable results. An application writer must ensure that data updates are grouped in transactions to ensure that at the end of a transaction all updates are consistent.

container
A component of an application adaptor that provides object services for instances of managed objects, which are organized in homes. Each container can contain one or more homes. Containers provide translation, termination, and memory management policies, as well as caching mechanisms for the managed objects within those homes, and maintain a list of the managed object instances.

container-managed persistence (CMP)
Persistence that is delegated to an enterprise bean's container, as opposed to being managed by the bean itself. See EJB container.

container-managed persistence (CMP) entity bean
Entity beans that delegate their persistence to their EJB container. Compare with bean-managed persistence (BMP) entity bean.

control descriptor
A special class (javax.ejb.deployment.ControlDescriptor) that defines the database isolation levels and other constant values for each method in an enterprise bean that is used by the deployment descriptor.

controlled server group
A server group that is configured for workload management control. Using a controlled server group enables the workload to be balanced across the servers in the group. It increases the throughput and decreases the response time of method invocations. When a server becomes unavailable, other available servers in the server group can continue to process requests.

control region
In OS/390 Component Broker, one of two kinds of address spaces that make up a server instance. The control regions queue messages to other parts of the server instance, called server regions. See server region.

conversation
In OS/390 Component Broker, an object that represents a group of changes being made to a Component Broker configuration. You use the Administration application to work with a conversation.

copy helper
An optional class that provides an efficient way for the client application to create new instances of the component on the server. The copy helper contains the same attributes as the business object, or a subset of them. Without a copy helper, the client might need to make many calls to the server for each new instance: one call to create the instance, and then an additional call to initialize each of the instance's attributes. With a copy helper, the client can create a local instance of the copy helper, set values for its attributes, and then create the server component and initialize its attributes in one call, by passing it the copy helper.

copy helper instance
An instance of a copy helper that is created using the _create()

function. The client developer sets values locally, then creates a managed object from the copy helper using the createFromCopy() function.

CORBA
See Common Object Request Broker Architecture.

CORBA client
The CORBA client is the native client in an OMG/CORBA programming model. This client, also known as the "pure" CORBA client, uses all of the C++ bindings produced by the idlc compiler to enable remote access to Component Broker managed objects from a client where the Object Request Broker is installed.

CORBA services
Services that comply with the Common Object Request Broker Architecture. OMG has defined a series of common services for use by objects and components in the operational environment. These services are specified by OMG at the API level to provide a consistent environment for object applications.

correlator
An attribute or type of string for both inbound and outbound messages. It is used to identify the inbound message, which is sent as a response to an outbound message. The response message has its correlator set to the correlator of the outbound message. See correlatorKey.

correlatorKey
An attribute or type of string of an inbound message. It is used by the MQAA implementation of the data object to retrieve a message based on a correlator. If the correlatorKey attribute is set, the data object implementation should retrieve the message with this correlator. If the correlatorKey attribute is not set, the data object implementation should retrieve the next message on the queue. See correlator.

credential
An object that represents the authenticity of a given principal. As such, it represents a principal, but only after the principal has been certified as being authentic. When a principal has been authenticated, the Security Service forms a credential for that principal.

CRUD methods
In object-oriented programming, the four standard operations that can be performed on a database: Create (insert), Retrieve, Update, and Delete.

Customer Information Control System (CICS)
An IBM licensed program that provides online transaction processing services and management for critical business applications. CICS runs

on many IBM and non-IBM platforms (from the desktop to the mainframe) and is used in various types of networks that range in size from a few terminals to many thousands of terminals. The CICS application programming interface (API) enables programmers to port applications among the hardware and software platforms on which CICS is available. Each product in the CICS family can interface with the other products in the CICS family, thus enabling interproduct communication.

D

daemon
A program that runs unattended to perform a service for other programs.

database
(1) A collection of data with a given structure for accepting, storing, and providing, on demand, data for multiple users. (2) A collection of interrelated data organized according to a database schema to serve one or more applications. (3) A collection of data fundamental to a system. (4) A collection of data fundamental to an enterprise.

database (DB) key
Uniquely identifies a row in a database. It enables data retrieval, data updates, and data creation. The database ensures that the column that the key represents holds only unique values. The database also creates an internal index for the column selected as primary key, for faster execution.

database management system (DBMS)
A suite of programs that typically manage large structured sets of persistent data, offering ad hoc query facilities to many users. They are widely used in business applications.

database (DB) persistent object
A type of persistent object that represents a record of a table or a view in a relational database. The state data of the component that is stored in the relational database by means of a persistent object lasts longer than the execution time of the application that calls the component. Database persistent objects have the following implementation types:

- Those that use embedded SQL to access and update the data they represent in a database.
- Those that use the caching capabilities of the server for accessing and updating data from the data stores they represent.

database (DB) schema
The object that is created when an SQL file is imported into Object Builder. Multiple persistent objects can be associated with a single database schema.

database 2
Obsolete term for DB2.

data definition language (DDE)
A scripting language that defines the structure of an application on both client and server. Object Builder can generate for each application family a DDE script that defines the structure of the applications in the family.

data description language
Synonym for data definition language.

data object
An object that manages the persistence of a business object's essential state data information by providing a business object with an interface for getting and setting all essential state information. A data object has two parts: the data object interface, which derives from the business object implementation, and the data object implementation, which derives from the data object interface. A data object isolates a business object from the following tasks:

- Knowing which of many data stores to use to make its state persistent.
- Knowing how to access the data store.
- Managing the access.

data structure
In FlowMark, the description of any data that is used as either input or output, or that is referenced in either exit or transition conditions.

data store
A database management system. It is a system for defining, creating, manipulating, controlling, managing and using databases. It is an implementation that stores the persistent data for an object independently of the address space containing the object. The software for using a database of the data store can either be a part of the data store, or it can be a stand-alone database system. Examples of data stores are OODBMS and RDBMS.

data structure member
One of the variables of which the data structure is composed.

data type
(1) A generic description of an elementary unit of information in a particular software system. (2) In Oracle and SQL, an attribute of columns, constants, and host variables. (3) In DB2, an attribute used for defining data. It describes the types of values accepted and imposes certain limits inherent to that type. (4) In C and Pascal, a set

of values, together with a set of permitted operations. A data type determines the kind of values that a variable can assume, or that a function can return.

DBCS See double-byte character set.

DBMS
See database management system.

DB persistent object
See database persistent object.

DB schema
See database schema.

DB2 An IBM object-relational database management system that supports the SQL database language. As an object-relational database management system, DB2 includes the concept of relational database management systems (RDBMS) and object-oriented database management systems (OODBMS).

DCE See distributed computing environment.

DDL See data definition language.

deferred synchronous request
See one-way request and synchronous request.

deployment descriptor
A serialized object that contains run-time settings for an enterprise bean and passes information to the EJB container about how to manage and control the enterprise bean.

derived class
A class that inherits from a base class. You can add new data members and member functions to the derived class. You can manipulate a derived class object as if it were a base class object. The derived class can override virtual functions of the base class. Synonym for child class and subclass. Compare with base class.

DII See dynamic invocation interface.

DIR See dynamic implementation repository.

DIS See dynamic implementation server.

dirty object
A persistent object that has been modified since it was last written to persistent storage.

disconnect
An OMG persistence service semantic, defined as follows: When the

connection ends, the data is the same in the persistent object and the data store, and the relationship between them no longer exists.

distributed computing environment (DCE)
The Open Software Foundation (OSF) specification consisting of standard programming interfaces, conventions, and server functionality (such as naming, distributed file system, and remote procedure call) for distributing applications transparently across networks of heterogeneous computers.

distributed system
A collection of heterogeneous automata whose distribution is transparent to the user so that the system appears as one local machine. This is in contrast to a network, where the user is aware that there are several machines, and their location, storage replication, load balancing and functionality is not transparent. Distributed systems usually use some kind of client/server organization.

DLL See dynamic link library and dynamic loadable library.

document type definition (DTD)
The syntax of markup constructs in markup languages such as SGML, HTML, or XML (DTD files are optional with XML documents). DTD mainly consists of the declarations of element types (which are structures such as paragraphs, or behavior such as hypertext links) and their properties called attributes. It usually begins with a series of parameter entity definitions, each of which defines a kind of macro that can be referenced and expanded elsewhere in the DTD. These macros do not have to appear in markup language documents, only in the DTD. The DTD can include additional definitions such as character entity references, which are numeric or symbolic names for rarely used characters (such as "<" for the < sign) that can also be included in a markup language document.

domain
(1) A distinct scope, within which common characteristics are exhibited, common rules observed, and over which a distribution transparency is preserved. (2) In Open Systems Interconnection (OSI), a part of a distributed system or a set of managed objects to which a common policy applies.

double-byte character set (DBCS)
A set of characters in which each character is represented by 2 bytes. Languages such as Japanese, Chinese, and Korean, which contain more symbols than can be represented by 256 code points, require double-byte character sets. Because each character requires 2 bytes, the typing, display, and printing of DBCS characters requires hardware and programs that support DBCS. Contrast with single-byte character set.

DSI See dynamic skeleton interface.

DTD See document type definition.

durability
(1) The attribute of an application in which the effects of a completed transaction are persistent. (2) The attribute of an object being able to preserve its internal states across system or software failure. Persistent objects that are the result of durability are always recoverable.

dynamic implementation repository (DIR)
Contains dynamic inheritance information. It is initially populated through the omf_diredit utility with information from the IDLX compiler. Changes are instigated by the omf_diredit utility or the DIR application programming interfaces.

dynamic implementation server (DIS or DI server)
An Object Request Broker (ORB) server that implements two basic features needed for the OMF. Multiple Dynamic Implementation servers can communicate with each other through the ORB.

dynamic invocation interface (DII)
The CORBA-specified interface used to build requests on remote objects. Object Request Broker applications can use the `dispatch` method for dynamic method calls when the object is remote.

dynamic link library (DLL)
A piece of code that can be loaded (activated) dynamically. This code is physically separate from its callers. Dynamic link libraries can be loaded at load time or at run time. Widely used term on OS/2, Windows, and, to some extent on AIX.

dynamic loadable library (DLL)
A separately compiled shared library module that is loadable at run time by AIX processes. The Object Request Broker (ORB) uses dynamic loadable libraries as vehicles for loading method implementations into the ORB server process.

dynamic skeleton
An interface-independent type of skeleton used by CB servers to handle requests whose signatures are not known until run time.

dynamic skeleton interface (DSI)
An interface that allows a client to invoke a method on an object it had no knowledge of at compile time.

E

EAB See Enterprise Access Builder.

ECI See external call interface.

EJB See Enterprise JavaBeans architecture.

EJB container
A run-time environment that manages one or more enterprise beans. The EJB container manages the life cycles of enterprise bean objects, coordinates distributed transactions, and implements object security. Generally, each EJB container is provided by an EJB server and contains a set of enterprise beans that run on the server.

EJB home
An object that implements the home interface of an enterprise bean. It provides life cycle operations for the bean. It is created by the EJB container's deployment tools, such as the Jetace tool.

EJB JAR file
A JAR file that contains one or more enterprise beans. See enterprise bean.

EJB server
A high-level process or application that provides a run-time environment to support the execution of server applications that use enterprise beans. An EJB server provides a JNDI-accessible naming service, manages and coordinates the allocation of resources to client applications, provides access to system resources, and provides a transaction service. An EJB server could be a database or an application server.

embedded aggregation
A form of aggregation where the sub-objects remain visible from outside of the aggregate.

embedded SQL
In DB2, the Structured Query Language (SQL) statements that are embedded within a program and are prepared during the program preparation process before the program is executed. Also called static SQL.

ENC See extended naming context.

encapsulation
(1) In object-oriented programming, the technique that is used to hide the inherent details of an object. The implementation details of a class are hidden from client programs that are only required to know the interface of a class in order to use the class's methods and attributes. (2) In object-oriented programming, a software technique in which data is packaged with corresponding procedures. In CORBA, the object is the mechanism for encapsulation.

enterprise
A computer system that is to be managed by system management.

The enterprise comprises servers, clients, applications, and other objects on one or more host computers.

Enterprise Access Builder (EAB)
In VisualAge for Java, a set of frameworks and tools for creating Java applications that access existing host applications and data, for example using CICS, IMS, MQSeries, or Encina.

enterprise bean
A nonvisual software component that conforms to Sun's Enterprise JavaBeans architecture, designed to be installed on a server and accessed remotely from a client. It realizes the standard component architecture for building distributed object-oriented business applications in the Java programming language.

Enterprise JavaBeans (EJB) architecture
A portable, platform-independent reusable component architecture defined by Sun Microsystems. EJB-compliant applications are deployed into EJB containers and run on an EJB server.

entity bean
Enterprise beans that contain persistent data, and that can be saved in various persistent data stores. Each entity bean carries its own identity. See bean-managed persistence (BMP) entity bean and container-managed persistence (CMP) entity bean.

error log
The log file used to store information when the Component Broker run time detects an unexpected failure. For example, error log entries are made for assertion failures, irrecoverable error conditions, and failures in memory and other vital resources.

essential state
See state data.

event (1) Any user action (such as a mouse click) or system activity (such as a screen update) that provokes a response from the application. (2) In Windows, a synchronization kernel object used to signal that an operation has completed.

event channel
The function used by the Component Broker Event Service to represent an event that has occurred during the processing of a method or application. An event channel can be viewed as both a consumer and a supplier of events. It consumes events provided by suppliers that detect the occurrence of events and notify the Event Service. It supplies the event to consumers that are processes or objects that have registered an interest in the event. Suppliers and consumers must have registered with the event channel to supply and consume occurrences of the event and any data related to it.

Event Service
The Component Broker service that enables processes and objects to be notified when events occur while a method or application is being processed.

execution environment
The platform that serves as the environment in which the process of carrying out an instruction or a computer program by a computer is done.

extended naming context (ENC)
A specialization of a naming context with extensions for properties, query, identity, administration, and name strings.

extensible markup language (XML)
A restricted form of the Standard Generalized Markup Language (SGML) which enables generic SGML documents to be served, received, and processed on the Web as though they were Hypertext Markup Language (HTML) documents. XML is designed for ease of implementation, and for interoperability with both SGML and HTML.

external call interface (ECI)
The principal communication mechanism used by CICS client classes to access a CICS server.

externalized object reference
An object reference expressed as an ORB-specific string, suitable for storage in files or other external media.

F

factory
An object responsible for creating other objects. In the Component Broker life cycle model, the factory object is responsible for encoding not only the specific type and implementation class for the kinds of objects that it supports, but also details on how to create an object of that type through the use of CORBA methods. Besides managed object factories, which are also called homes, other kinds of factories like application factories and abstract factories exist, depending on the type of objects that the factory creates. See home.

factory object
An object that is capable of creating another object. A factory object will create and initialize a business object and any other helper objects required by the business object so that it can be managed by an instance manager.

factory finder
An object that is used to find homes. It does this by looking through a set of locations to find a home (factory) that creates objects of a certain

type. This search capability of the factory finder relieves client programmers from involvement in complicated location issues in a dynamic server topology.

FDL See FlowMark Definition Language.

filter (1) In System Management, the function that can be used to focus the view on subsets of the objects displayed by an Information Controller window. Each Information Controller window has its own filter that is applied to all navigation actions. You can use the Edit Filter Details window to display and change your filters. (2) In Object Builder, the function used to display different subsets of objects in the Tasks and Objects pane.

FlowMark Definition Language (FDL)
An external format for defining programs, data structures, and workflow models in a flat, text file. Definitions in a FDL file can be imported into a FlowMark database.

foreign key
A column or set of columns in a table whose values must match at least one primary key or unique key value of a row in the parent table. It is a key specified in the definition of a referential constraint. A referential constraint limits the validity of foreign key values by ensuring that they appear as values of a parent key.

framework
A set of object classes that provide a collection of related functions for a user or piece of software. A framework partitions a design into abstract classes and defines their responsibilities and collaborations.

G

GDS See Global Directory Service.

generic security service API (GSSAPI)
An application programming interface (API) that allows client/server programs to request authentication services from an underlying security service, such as DCE, without regard for the security or communication mechanisms used by that service. This API is used with the object security service to make use of DCE.

Global Directory Service
A component of a Distributed Computing Environment (DCE) that provides X.500 directory services and can be used to obtain resource information from outside a DCE cell or to publish resource information to nodes outside a DCE cell.

GSSAPI
See generic security service API.

H

handle
A class that encapsulates an object reference. An object handle enables a single reference collection implementation to use any possible swizzling pattern for its persistent references.

helper server
An alternative server that supports a method implementation as determined by the Dynamic Implementation Repository.

history
The record of all objects that are acted on through the System Management User Interface. When an Information Controller window is opened, a session lasts until it is closed. All objects that are acted on during this session are recorded in the history for that window. The History window is used to display the session history and to revisit any object in the history list.

HOD See host on-demand.

home A run-time instance of an object, which is either an instance of the IHome class, or inherits from it. It is a specific form of managed object that is used to create and find instances of other managed objects that are configured with it. In fact, a home is a factory for a managed object. Each managed object instance can exist in only one home. A home exists in a container, which provides object services for the instances in the home. Homes are also collections. See factory and specialized home.

home interface
An interface that specifies the available methods for locating, creating, and removing instances of enterprise bean classes.

host model
The type of model object that defines an instance of a host computer. This enables system management to relate other model objects (for example, server models) to a particular host. A host model is created automatically (and its name assigned) by the installation of a System Management agent or a System Manager on that host. Host models can be grouped arbitrarily into cell and workgroup models.

host on-demand (HOD)
An eNetwork, Java-based software solution that incorporates industry-standard Telnet 3270 protocols.

hotlist A permanent list of objects that a user can access directly during all sessions with the System Management User Interface. Each user can create his or her own hotlist of significant objects that is preserved

across sessions with the System Management User Interface. The window is used to display the hotlist and to navigate to a selected object.

HTML
See hypertext markup language.

hypertext markup language (HTML)
A file format, based on SGML, for hypertext documents on the Internet. Enables embedding of images, sounds, video streams, form fields, and simple text formatting. References to other objects are embedded using URLs, enabling readers to jump directly to the referenced document.

I

IBM FlowMark
A tool that has a workflow manager that enables users to automate their business processes. The workflow manager usually runs as a distributed application on local area networks that consist of several workstations, but users can also have the FlowMark clients and servers on a single, stand-alone workstation.

IDE See integrated development environment.

IDL See interface definition language.

IIOP See internet inter-ORB protocol.

image object
The type of system management object used to represent things that exist in the real world. For example, a Server Image represents a server that exists on a host computer. The image objects make up the image world that is used to monitor enterprises through Component Broker System Management.

image world
The part of the common data model that contains image objects. The image world is generated automatically by system management from the model world and install world. It permits system administrators to work with image objects - to stop and start servers, for example, and to change some of their attributes - but restricts the deletion and creation of image objects to administrators of the model world. The System Management agent on each host maintains a separate image world.

implementation
(1) A definition that provides the information needed to create an object and enable 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 also typically include information about the intended interface of the object. (2) In the Object Request Broker, a remote object implementation is also characterized by its server implementation (a program).

implementation definition language
A notation for describing implementations, that may be vendor specific or adaptor specific.

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 interface
An interface introduced as a derivative of the most specialized interface of the object. The implementation interface is intended to designate the type of a specific implementation - the Type ID of the implementation interface can be used within CORBA to differentiate implementations of the same interface. The implementation interface is also used to bring together the operational interface with the specific administrative interface relevant to that particular implementation.

implementation object
An object that serves as an implementation definition. Implementation objects reside in an implementation repository.

implementation repository
A database used by the Object Request Broker (ORB) to store the implementation definitions of ORB servers.

IMS See information management system.

information management system (IMS)
A database or data communication system that can manage complex databases and networks.

inbound message object
In Object Builder, a business object that represents a message retrieved from an MQSeries queue.

inheritance
In object-oriented programming, the ability to derive new classes from existing classes. A derived class inherits the instance variables and methods of the base class, and can add new instance variables and

methods. New methods can be defined with the same names as those in the base class, in which case they override the original one. See base class and derived class.

inheritance hierarchy
The sequential relationship from a root class to a derived class, through which the derived class inherits instance methods, attributes, and instance variables from all of its ancestors, either directly or indirectly.

initializer
A value used as the default until explicitly changed.

installation object
The type of system management object used to define applications (for example, their application programs, classes, dynamic link libraries, homes, and containers). The install objects make up the install world that defines applications managed by Component Broker System Management.

install world
The part of the common data model that defines the topology of applications in terms of the classes and dynamic link libraries of which applications consist, and the homes and containers that they need to run. Install objects are created during Component Broker installation, and are used to facilitate system management. The install world is created by application installation tools. Although the install world can be viewed through the Component Broker System Management User Interface, it cannot be changed by system administrators. This data is stored in both the System Manager and System Management agent, but is primarily intended for use by System Management agent.

instance
An object that is a member of that class. An object created according to the definition of that class.

instance manager
See application adaptor.

instance manager mixin
See application adaptor mixin.

integrated development environment (IDE)
A system for supporting the process of writing software. Such a system can include a syntax-directed editor, graphical tools for program entry, and integrated support for compiling and running the program and relating compilation errors back to the source.

interface

An interface describes the set of available operations that a client can request from an object. That is, an object publishes how other objects or applications can interact with it through an interface. In CORBA, interfaces are defined using the Object Management Group (OMG) Interface Definition Language (IDL). A CORBA interface definition written in IDL defines the operations available in the interface, including each operation's parameters. IDL follows the same lexical rules as C++, with some added keywords to support distribution concepts. A Java interface is defined like a Java class, but with only the declarations of its methods. A Java interface is implemented by Java classes. The interface specifies what methods must be implemented by these classes but not what these methods do.

Interface Definition Language (IDL)

Defines the types of objects, their attributes, the methods they export, and the method parameters. It is a language by which objects tell their potential clients what operations are available and how they should be invoked. The CORBA IDL is a subset of ANSI C++ with additional constructs to support distribution. The IDL is a purely declarative language that uses the C++ syntax for constant, type, and operation definitions, and it does not include any control structures or variables. CORBA IDL can be used to specify component attributes (or public variables), the parent class it inherits from, the exceptions it raises, typed events, pragmas for generating globally unique identifiers for the interfaces, and the methods an interface supports, including the input and output parameters and their data types.

interface inheritance

The construction of an interface by incremental modification of other interfaces. IDL provides interface inheritance.

interface repository (IR)

The database that Component Broker optionally creates, providing persistent storage of objects that represent the major elements of interface definitions. Creation and maintenance of the IR is based on information supplied in the Interface Definition Language source file.

interface repository framework

A set of classes that provide methods whereby executing programs can access the persistent objects of the interface repository to discover everything known about the programming interfaces of Component Broker classes.

interface type

A data type that has the description of its interface as its data. It can be used as a type for other attributes, method return types, and parameter types.

interlanguage object model (IOM)
The name of the language interoperability technology used by Component Broker to permit objects written in C++ and objects written in Java to interact cooperatively within a single process. IOM is the component that makes local interactions between objects written in different languages efficient enough to be practical.

internet inter-ORB protocol (IIOP)
A protocol that is mandatory for all CORBA 2.0-compliant platforms. The initial phase of the project is an infrastructure consisting of:

- An IIOP to HTTP gateway that allows CORBA clients to access WWW resources.
- An HTTP to IIOP gateway to enable WWW clients to access CORBA resources.
- A web server that makes resources available by both IIOP and HTTP.
- Web browsers that can use IIOP as their native protocol.

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.

interoperable object reference (IOR)
Maintains information about the type and key of an object and the communications profiles needed to contact the CB Server and locate the object.

inter-shell language (ISL)
A language used to communicate method parameters to and from script methods. Parameters are converted to the ISL by the dynamic implementation server and delivered to the command interpreted language by the stdin data stream.

IOM See interlanguage object model.

IOR See interoperable object reference.

IR See interface repository.

ISL See Inter-Shell Language.

isolation
Intermediate states of a transaction are not visible to other transactions. Transactions appear to execute serially, even when they are executing concurrently. There is complete independence of tasks to ensure that two transactions cannot operate on the same data at the same time; one transaction must be delayed until the other one completes its tasks, or it must be canceled. When changes to resources

are committed, no change is dependent on uncommitted changes by other concurrent transactions. A locking mechanism must be available to prevent simultaneous updates of the same data. Isolation can be provided by the Concurrency Service.

J

JAR file
A Java ARchive file. At its simplest, a JAR file is a collection of uncompiled .java files, though a JAR file can contain .class files, images, sounds, and other files. See EJB JAR file.

Java An object-oriented programming language for portable interpretive code that supports interaction among remote objects. Java was developed and specified by Sun Microsystems, Incorporated.

JavaBeans
A portable, platform-independent reusable component model.

Java client
An Object Request Broker client written entirely in Java and associated development tools. In a manner similar to the other clients, the idl2java tool enables client developers to process an Interface Definition Language file to create managed object proxies that can be used from Java. (This is integrated by Object Builder). Given that the development work is done and deployed in Java, developers can use the managed object proxy from an applet that is downloaded at run time from their server or from an application that is installed on the client.

Java virtual machine (JVM)
A software implementation of a central processing unit (CPU) that interprets byte codes (compiled Java code) in the Java Runtime Environment (JRE).

junction
A junction represents a transition in a name tree federation between different implementations of a naming context. If database based naming context is bound into a Cell Directory Service based naming context, that binding forms a junction.

JVM See Java virtual machine.

K

key The attributes that are to be used to find a particular instance of the component on the server. It consists of one or more of the business object attributes, which must contain enough information to uniquely identify an instance.

key assistant
A new key helper class. It is a concrete subclass that is associated with a component and has knowledge of the primary key that is configured for the component.

key class
The key class contains methods that are used by managed objects and containers to assemble a key for an object at the time an IOR is being constructed. It contains methods that are used by containers to pick apart a key at the time a 1-1 relationship is being mapped back to the actual object pointer that is providing the implementation.

L

language binding
The conventions by which a programmer writing in a specific programming language accesses ORB capabilities.

language mapping
See language binding.

legacy method
A method that is implemented as an executable in the file system. Parameters are passed to and from a legacy method using UNIX I/O through the argv parameter and the stdin, stdout, and stderr data streams. Legacy methods typically are executables or shell scripts that already exist.

localized CORBA client
The localized "on-server" CORBA client is a process running on the same machine as its server processes. Although the localized CORBA client runs on the server in proximity to the server processes, it is considered a client type because it uses different code packages to access the services on the server, than the server processes.

location object
A special object that is used to relate objects that are near each other. Nearness can mean that the objects are close physically, organizationally, or even temporarily (based on time of creation). Component Broker locations provide a mechanism for defining policies that recognize objects that are near each other.

location services daemon
A process whose primary purpose is to give Object Request Broker clients the communications information they need to connect with an implementation server.

lock (1) The means by which integrity of data is ensured by preventing more than one user from accessing or changing the same data or

object at the same time. (2) In Communications Manager/2, a password-protection system that can be used to prevent access to some advanced functions.

logical resource mapping
In OS/390 Component Broker, an object used by containers that locates logical resources on the system. There may be one or more logical resource mappings for each type of subsystem in the sysplex, for example, DB2 or CICS. Logical resource mappings are connected to containers with logical resource mapping connections.

logical resource mapping connection
In OS/390 Component Broker, links a container and one or more logical resource mappings.

logical resource mapping instance
In OS/390 Component Broker, an instance of a logical resource mapping that exists on a specific system. It provides information that allows a server instance to connect to a subsystem on that system.

LRM See logical resource mapping.

LRM connection
See logical resource mapping connection.

LRM instance
See logical resource mapping instance.

M

macro An alias for executing a sequence of hidden instructions on a method activation policy. Methods require method activation policies to specify how the method implementation is executed.

manageable object
Any object that inherits from the IManagedClient::IManageable interface class. Manageable objects include business object interfaces and business object implementations.

managed object
(1) A component of a system that can be managed by a management application. (2) In Component Broker, a class of objects that defines the set of methods that must be implemented by the business object to work with the appropriate application adaptor. The managed object is part of a component, and is managed by the application adaptor. It handles communication with other classes, as well as initialization, de-initialization, activation, and passivation of the business object. By default, Object Builder names managed objects with the *MO* suffix.

managed object framework (MOFW)
A framework that provides an organized environment for running a

collection of objects. It contains managed objects, which define the set of methods that must be implemented by the business object to work with the appropriate application adaptor, and makes available a common set of interfaces that are available to users of all managed objects. The MOFW defines the rules for resource and service usage for all managed objects in the CB environment.

managed object mixin
A means by which an application adaptor can provide a greater quality of service than the underlying operating system provides. By composing the mixin with the business object, the application adaptor can provide implementations for some of the methods of the business object, and provide added behavior to all of the methods of the business object.

management zone
The definition of all or part of an enterprise that is to be managed as a unit. Where several management zones are used, each one represents a separate section of the whole enterprise and does not overlap any other sections represented by other management zones. A management zone comprises one or more alternative configurations. Dividing an enterprise into management zones enables administrators to alter only the characteristics of their own management zones without affecting the functioning of others.

markup
All forms of codes inserted into electronic texts to govern formatting, printing, or other processing.

marshaling
The process of packing one or more items of data into a message buffer, before transmitting that message buffer over a communication channel. The packing process not only collects together values that can be stored in non-consecutive memory locations but also converts data of different types into a standard representation agreed with the recipient of the message.

master container
The container owned by the root application adaptor.

message queue
See queue.

message queuing
A programming technique in which each program within an application communicates with the other programs by putting messages on queues.

metadata
The self-descriptive information that can describe both services and information. With metadata, new services can be added to a system and discovered at run time.

meta model
The set of descriptions of the types that the instances in a model conform to.

method
A procedure or routine associated with one or more classes. Different classes can define methods with the same name (that is, methods can be polymorphs). The term "method" is used both for a named operation (such as PRINT), and also for the code that a specific class provides to perform that operation.

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

mixin object
See application adaptor mixin.

model A collection of objects that are accompanied by their state and behavior; the collection can be described by a set of instance data.

model object
A type of system management object used to define the things to be managed (for example, hosts, servers, and applications), the relationships between those things, and to create logical groups of objects for better management. The model objects make up the model world that defines an enterprise managed by Component Broker System Management.

model world
The part of the common data model that defines the topology of an enterprise in high-level terms. For this world, you use model objects to define the things to be managed (for example, hosts, servers, and applications), define the relationships among managed objects, and create logical groups of objects for better management. Model objects are normally created by a system administrator through the System Management User Interface. The data is stored in the System Manager and can be backed up and restored independently of any other data.

module
The organizational structure required within an Interface Definition Language source file that contains interface declarations for one (or more) classes that are not a class-superclass pair. Such interfaces must be grouped within a module declaration.

MOFW client programming model
A model that shows how application developers create objects that are clients of Component Broker applications and business objects. Application developers can use the Component Broker MOFW client programming model and development tools to build client and server applications whenever they use a managed object to implement a new object. See managed object framework.

MOFW server programming model
A model that is essentially an interface model of the Component Broker business object. This model can be used by client application developers to build business objects based on the managed object framework. The business objects that application developers create can be used and reused to create client/server applications.

MQSeries
A family of IBM-licensed programs that provide message queuing services.

multiple inheritance
(1) An object-oriented programming technique implemented in C++ through derivation, in which the derived class inherits members from more than one base class. (2) The structuring of inheritance relationships among classes so a derived class can use the attributes, relationships, and functions used by more than one base class.

N

name binding
The procedure used to associate the name tree for a host for a remote name tree.

naming context
A naming context is a container of name bindings that associates a human readable name to an object reference. Naming contexts support the CosNaming::Naming Context interface.

naming scope
See scope.

Naming Service
A part of Object Services that is composed of artifacts that help in the creation and maintenance of object names. The principal artifact of the Naming Service is a naming context. See naming context.

navigation
See transaction object.

network
Hardware and software data communication systems.

non-essential state
Transient data that is associated with an object, and can be recreated as required. The non-essential state is usually derived from other states and complements the essential state.

O

object (1) In Object Builder, an IDL interface along with one or more classes in C++ or Java that work together to provide part of the function of a component. For example, business object, data object, managed object. This term is not used in the sense of a class instance. See component. (2) A computer representation of something that a user can work with to perform a task. An object can appear as text or as an icon. See instance. (3) A collection of data and member functions that operate on that data, which together represent a logical entity in the system. In object-oriented programming, objects are grouped into classes that share common data definitions and member functions. Each object in the class is said to be an instance of the class. (4) In Windows, any item that is or can be linked into another Windows application, such as a sound, graphic, piece of text, or portion of a spreadsheet. An object must be from an application that supports OLE. See object linking and embedding.

object adaptor
The ORB component which provides object reference, activation, and state related services to an object implementation.

Object Builder
The development environment for Component Broker. Object Builder is used to develop applications from start to finish, or you can start by designing in Rational Rose and then import the design into Object Builder, where you add the final objects and program logic. Object Builder supports the CORBA programming model using IDL, C++, and Java. Complete working applications can be generated, including unit test versions and full client/server packages complete with server setup scripts.

Object Concurrency Service
The CORBA services specification for managing concurrent access to an object or resource.

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

object class libraries
These provide foundations for creating, assembling, and reusing objects. Class libraries package objects for reusability.

object definition
See class.

object destruction
A physical deletion of an object from main memory and permanent storage.

object frameworks
These provide foundations for creating, assembling, and reusing objects. Object frameworks are pre-assembled and packaged class libraries that provide many specific functions such as snapshots of database data to implement concurrency control mechanisms, and buffering mechanisms that determine when changes in the user interface are actually applied to enterprise objects.

object implementation
See implementation.

object identifier (OID)
An identifier that is uniquely assigned to an object. Object identifiers are never reused to identify another object.

object invocation access policy
The mechanisms within the systems that transparently control access to objects. The object invocation access policy is enforced by the system without application involvement. See application access policy.

object level trace
A tool for testing distributed applications. It includes a graphical trace facility and remote debugger.

object linking and embedding
(1) An API that supports compound documents, cross-application macro control, and common object registration. OLE defines protocols for in-place editing, drag-and-drop data transfers, structured storage, custom controls, and more. (2) A data-sharing scheme that allows dissimilar applications to create single complex documents cooperatively. The documents can consist of material that a single application could not have created on its own.

Object Management Group (OMG)
A non-profit consortium whose purpose is to promote object-oriented technology and the standardization of that technology. OMG was formed to help reduce the complexity, lower the costs, and hasten the introduction of new software applications.

object model
Any description of the structure of the objects in a system including their identity, relationships to other objects, attributes, and operations.

object-oriented analysis (OOA)
The discovery, analysis, and specification requirements in terms of objects with identity that encapsulate properties and operations, message passing, classes, inheritance, polymorphism, and dynamic binding.

object-oriented database management system (OODBMS)
A database management system that integrates database capabilities with object-oriented programming capabilities. It is a data store that stores an object in its binary form. Unlike an RDBMS, it does not require that the object be converted into one or more fixed data types for storage.

object-oriented programming (OOP)
A programming approach based on the concepts of data abstraction and inheritance. Unlike procedural programming techniques, object-oriented programming concentrates on what data objects comprise the problem and how they are manipulated, not on how something is accomplished.

object-oriented SQL (OOSQL)
A query language where the syntax of the query is expressed in the Structured Query Language with a minimal number of extensions for objects. The primary extension is the path expression. This extension generalizes the notion of column for navigation through attributes, methods, structures (for example, the struct construct in CORBA IDL), references to objects, or collections of objects.

1-1 relationship
A relationship between a business object (an attribute) and another business object (a type of that attribute).

1-n relationship
A one-to-many relationship defined between business objects.

Object Request Broker (ORB)
A communications protocol for conveying messages between objects. A CORBA term designating the means by which objects transparently make requests (that is, invoke methods) and receive responses from objects, whether they are local or remote. An ORB is the implementation of an OMG specification which allows the distribution of objects across a system or network.

Object Services
A collection of work-item components you can use to help with the creation, storing, definition, and naming of objects when building Component Broker applications. Component Broker Object Services

provides the following work-item services consistent with CORBA 2.0: Naming, Security, Transaction, Concurrency, LifeCycle, Event, Notification, and Externalization.

Object Transaction Service (OTS)
The CORBA services specification for managing atomic units-of-work over a series of method requests on recoverable objects.

OCS See Object Concurrency Service and lock.

OID See object identifier.

OLE See object linking and embedding.

OMG See Object Management Group.

one-copy serializable
The consistency property of the replication framework which states that the concurrent execution of methods on a replicated object is equivalent to the serial execution of those same methods on a non-replicated object.

one-way request
The client does not wait for completion of the request, nor does it intend to accept results. Contrast with synchronous request.

OOA See object-oriented analysis.

OODBMS
See object-oriented database management systems.

OOP See object-oriented programming.

Open Software Foundation (OSF)
A nonprofit research and development organization whose goals are (a) to develop specifications and software for use in an open software environment and (b) to make the specifications and software available to information technology vendors under fair and equitable licensing terms. For example, OSF developed the Distributed Computing Environment (DCE).

operation
In object-oriented design or programming, a service that can be requested at the boundary of an object. Operations include modifying an object or disclosing information about an object. See operation class.

operation class
A class that defines all required element and key operations required by a specific collection implementation.

operational interface
The interface of an object that defines its operational behavior. Typically the operational interface of an object is used by general business applications.

optimistic caching
A form of caching in which data is read from the database and kept in memory, but the data is not locked in the database. That is, the locks are released soon after the data is read. This results in a higher level of data concurrency. However, the application cannot be guaranteed current data. Optimistic caching applies to objects that are activated by either a primary key or a query. When using procedural application adaptors, optimistic caching is used only with transactions, not with sessions.

ORB See Object Request Broker.

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

ORB daemon
The SOM ORB program. It is an ORB executable program that is responsible for server location and activation. See location services daemon.

ORB object reference
See 1-1 relationship and proxy object reference.

OSF See Open Software Foundation.

OTS See Object Transaction Service.

outbound message object
In Object Builder, a business object that represents a message sent to an MQSeries queue.

override
The technique by which a class replaces (redefines) either the attributes (variables) and methods of the class from which it inherits so that it has its own characteristics.

overridden attribute
An attribute that is defined by the base class (parent class), and is re-implemented (overridden or redefined) in the child class. In Object Builder, you can modify the `get` and `set` methods of attributes that you override.

overridden method
A method that is defined by the base class (parent class) and

re-implemented (redefined or overridden) in the child class. An overriding method can elect to call the parent class' method procedure as part of its own implementation.

P

PAA See procedural application adaptor.

PA bean
See procedural adaptor bean.

package file
A file created in the associated database when code is generated for a database persistent object. This package file is used to resolve any unresolved names such as column names or table names that are specified in the .sqx file, which is generated from the persistent object. The DB2 precompiler takes the .sqx file as input, and the makefile creates a bind file with the same name as the generated .sqx file. See bind file.

packet integrity security
In Component Broker, the Object Request Broker (ORB) provides safeguards that protect all parties from false claims that data was tampered with or not sent or received. To accomplish this, the ORB provides senders with proofs of delivery and receivers with proofs of the sender's identity. Component Broker uses an industry standard roll-your-own (RYO) encryption mechanism that allows the ORB to provide secure communication standards between two or more principals.

PAO See procedural adaptor object.

parameter
(1) A variable that is given a constant value for a specified application and that may denote the application. (2) In basic CUA architecture, a variable used in conjunction with a command to affect its result. (3) An item in a menu for which the user specifies a value or for which the system provides a value when the menu is interpreted. (4) Data passed to a program or procedure by a user or another program, namely as an operand in a language statement, as an item in a menu, or as a shared data structure. (5) In NetView commands, a part of the command object.

parameter passing mode
The direction of information flow for an operation parameter. The parameter passing modes are *in*, *out*, and *inout*.

parent class
A class from which another class inherits instance methods, attributes, and instance variables. A parent class is sometimes called a base class or superclass.

parent method call
A technique where an overriding method calls the method procedure of its parent class as part of its own implementation.

PA persistent object
See procedural adaptor persistent object.

PA schema
See procedural adaptor schema.

pass ticket
In RACF secured sign-on, a dynamically generated, random, one-time-use, password substitute that a workstation or other client can use to sign on to the host rather than sending a RACF password across the network.

passivate
The operation that is performed on every managed object when the object is temporarily removed from storage. This is the managed object framework equivalent of an operating system paging storage out to disk. Resources being used by the managed objects must be released when this operation is performed. Managed objects can be made active again by a reactivation operation.

persistence
See durability.

persistent identifier (PID)
An opaque value that can be used by a persistence mechanism to deduce the data store information needed to locate the persistent state of an object.

persistent object
A C++ object that provides a mechanism for storing the state of a component in a data store. Each persistent object has an identifier or a key that is used for locating its corresponding record within the data store. There can be multiple persistent objects associated with a data object. These objects can be used for reading data and writing data. All of these categories of persistent objects; however, can be encapsulated into the same data object, giving it the appearance of a single readable and writable entity. There are two kinds of persistent objects based on the type of data that they manage. These types are:

- Database persistent objects that store the state data of the component in the relational database.

- Procedural adaptor persistent objects that store the data using transactions against the reusable procedural application.

persistent object (PO) key
The persistent identifier (PID) that is used to uniquely identify the persistent object. The persistent object key is most often the primary key in the database, but does not even have to be a key in the database at all.

pessimistic caching
The traditional form of caching, which maintains data integrity in the model. It makes use of repeatable read isolation in the database manager. When using procedural application adaptors, pessimistic caching is used only with transactions, not with sessions.

PID See persistent identifier.

policy group
A set of bind policies to be used for an application. A policy group applies all the selection criteria of its bind policies to give a cumulative effect.

portability
The ability of a programming language to compile successfully on different operating systems without requiring changes to the source code.

precision
The total number of digits in the decimal number. See scale.

principal
A user or system entity that is identified to the security system. Principals can be authenticated by the security system.

private
Pertaining to a class member that is accessible only to member functions and friends of that class.

privilege attribute
Security information associated with a principal that is used during the process of granting that principal access to data or resources.

procedural adaptor object (PAO)
The object that is used in the generated code for the procedural adaptor persistent object. The interface of a procedural adaptor object is also used to define the procedural adaptor schema.

procedural adaptor (PA) bean
In VisualAge for Java, a bean that inherits from the CBProceduralAdapterObject class. Procedural adaptor beans, built using the Enterprise Access Builder (EAB) support, wrap existing

transactions for reuse in Component Broker. Procedural adaptor beans are imported into Object Builder as procedural adaptor schemas and procedural adaptor persistent objects.

procedural adaptor (PA) persistent object
A type of persistent object that encapsulates entities that are accessed through a procedural system. These entities typically have data attributes as well as logic to manipulate these attributes; for example, a CICS application that manages customers. A procedural adaptor persistent object is responsible for storing the data using transactions against the reusable procedural application.

procedural adaptor (PA) schema
The object that is created in Object Builder when a procedural adaptor bean is imported from VisualAge for Java. This schema has an associated persistent object at the time of its creation, but it can be associated with more than one persistent object.

procedural application adaptor (PAA)
An extension of the business object application adaptor (BOIM). This adaptor enables the reuse of pre-existing legacy business logic (transactions) and data. The procedural application adaptor provides the mapping between Component Broker objects (as well as their references) and an existing legacy procedural and transactional business application. For example, a procedural application adaptor can be used for mappings between Component Broker objects and CICS or IMS applications.

procedure
A small section of code that executes a limited, well-understood task when called from another program. See method procedure.

process
(1) A collection of code, data, and other system resources, including at least one thread of execution, that performs a data processing task. (2) A running application, its address space, and its resources. (3) An instance of a running program. A Win32 process owns a 4 GB address space containing the code and data for an application executable file; it does not execute anything. It also owns certain resources, such as files, dynamic memory allocations, and threads. (4) A program running under OS/2, along with the resources associated with it (memory, threads, file system resources, and so on). (5) In FlowMark, a sequence of activities that must be completed to accomplish a task. The process is invoked when the activity is started either automatically or manually.

process activity
In FlowMark, an activity to which a separate process is assigned.

When a user starts this activity, or when it starts automatically, an instance of the referenced process is created and started.

profile
(1) Data that describes the significant characteristics of a user, a group of users, or one or more computer resources. (2) In Object Builder, a shell script that contains initializations of environment variables.

program
In FlowMark, a computer-based application program that supports the work to be done in an activity. Program activities reference executable programs using the logical names associated with the programs in the FlowMark program registrations. Program registrations can contain run-time parameters for the executable programs.

program activity
In FlowMark, an activity to which a registered program is assigned. Starting this activity, either automatically or manually, invokes the program.

program registration
In FlowMark, the identification of a program to a FlowMark database so that the program can be assigned to a program activity in a workflow model.

propagation
A function of the Transaction Service that transfers the transactional context of a client along with a method request to a served object. The Transaction Service supports both implicit and explicit propagation of a transaction context. Implicit propagation occurs automatically as the result of invoking any method on an object whose class inherits from CosTransactions::TransactionalObject. Explicit propagation occurs only when the client obtains a context object and passes that as an explicit argument on a method request.

protected
In C++, pertaining to a class member that is accessible by member functions and friends (classes or functions) of the class, as well as by classes derived from the class.

protocol
The meanings of, and the sequencing rules for, requests and responses used for managing a network, transferring data, and synchronizing the states of network components.

proxy An object that has the same interface as the server object it represents. Instead of having the actual method implementation, its methods communicate with the ORB.

proxy object reference
An object reference used by a client to refer to an object located in a remote server. The proxy is a pointer to a data object, with information encoded in the reference data to refer to the actual object on the server.

proxy push supplier
A proxy object that binds a pull supplier to an event channel.

public In C++, pertaining to a class member that is accessible by any function.

push down method
In Component Broker, a method that can indirectly cause communication with the data store. The method is written on the procedural adaptor bean in VisualAge for Java. When imported to Object Builder, this bean contains the mapping of the method implementations that are transferred from the persistent object through the data object to the business object. The procedural application adaptor handles the logic processing.

Q

query evaluator
An object used by the Component Broker Object Query Service as a search engine to find objects.

queue An MQSeries object. Message queuing applications can put messages on, and get messages from, a queue. A queue is owned and maintained by a queue manager. Local queues can contain a list of messages waiting to be processed. Queues of other types cannot contain messages—they point to other queues, or can be used as models for dynamic queues.

queue manager
(1) A system program that provides queuing services to applications. It provides an application programming interface so that programs can access messages on the queues that the queue manager owns. (2) An MQSeries object that defines the attributes of a particular queue manager.

quiesce
(1) To end a process by allowing operations to complete normally. (2) To request that a node stop sending synchronous-flow messages.

R

RACF See Resource Access Control Facility (RACF).

RACF secured sign-on
In the Resource Access Control Facility (RACF), a function that

enables workstations and other clients to sign on to the host and communicate in a secure way without having to send RACF passwords across the network. See pass ticket.

RAS Reliability, availability, and serviceability. See remote access services.

RDBMS
See relational database management systems.

recoverable object
An object whose data is effected by committing or rolling back a transaction.

recoverable server
A server containing one or more recoverable objects.

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

relational database application adaptor
An extension of the Business Object Application Adaptor (BOIM). It enables object reuse of existing relational databases. This adaptor enables the value of any identified field in a database to be read or to be updated using a single SQL statement. Fields in a database have a one-to-one relationship with attributes of the data object.

relational database management systems (RDBMS)
A collection of hardware and software that organizes and provides access to a relational database.

remote access services
A feature of Windows NT that allows most of the services available on a network to be accessed over a modem link. The services include support for dial-up and logon functions, and then present the same network interface as the normal network drivers (though not the same access time). It is not necessary to run Windows NT on the client; there are client versions for other Windows operating systems.

remote build
A version of a program, typically one that is still being tested that is activated on another computer, which is distant from a central site, usually over a network connection. The remote computer may be stationary and nonportable, or it may be portable.

remote interface
An interface that specifies the remote business methods that a client can call on an enterprise bean.

remote procedure call (RPC)
(1) A facility that a client uses to request the execution of a procedure

call from a server. This facility includes a library of procedures and an external data representation. (2) A client request to a service provider located in another node.

repository
The core of a computer-aided software engineering (CASE) tool, typically a DBMS where all development documents are stored. See interface repository and implementation repository.

request
Any invocation of an operation on an object or class, comprising an operation name, a list of zero or more actual parameters, and the results to be returned to the client.

required rights
A list of access rights associated with a method. The required rights list is compared with the rights associated with object groups for which the object is a member. A match means that access is granted to the method.

Resource Access Control Facility (RACF)
An IBM licensed program that provides for access control by identifying and verifying the users of the system, by authorizing access to protected resources, by logging the detected unauthorized attempts to enter the system, and by logging the detected accesses to protected resources.

restart daemon
A restart facility provided by the Transaction Service that can be used to automatically restart transactional processes.

restart repository
A file that contains information about the transaction programs being run on a machine. The restart repository is used by the restart daemon during restart of failed transaction processing.

results
The information returned to the client by the given message or operation.

resynchronization
The process where objects recovered after a server failure contact one another to resolve the outcome of the transaction.

reusable
Using code developed for one application program in another application. Traditionally achieved using program libraries. Object-oriented programming offers reusable code through its techniques of inheritance. Polymorphic functional languages support reusability while retaining the benefits of strong typing.

rights (or user rights)
Are associated with object groups and compared with the required rights associated with a method. The comparison only takes place for object groups of which the object is a member.

root A node that has no parent. All other nodes of a tree are descendants of the root.

root application adaptor
A container that contains other application adaptor containers. It is a bootstrap mechanism integrated directly into the CB Server environment.

RPC See remote procedure call.

run time
The data structures, objects, and global variables that are created, maintained, and used by the functions, procedures, and methods in the Component Broker run-time library.

S

SBCS See single-byte character set.

scalable
Pertaining to the capability of a system to adapt readily to a greater or lesser intensity of use, volume, or demand. For example, a scalable system can efficiently adapt to work with larger or smaller networks performing tasks of varying complexity.

scale The number of digits in the fractional part of the number. That is, the number of digits that lie to the right of the decimal point. See precision.

schema
A textual description of an object type that is understood and stored in a persistent data store. It is a structural and behavioral abstraction of the real physical data and focuses on information relevant to users of the applications that use an existing database or access an existing procedural system. See database schema and procedural adaptor schema.

schema group
An organization of the different schemas imported from the DDL file. When the generate action is used on a schema group, an SQL file is created. The file contains the subset of the definitions of the schemas that the Object Builder requires to do table to persistent object mapping.

scope That portion of a program within which an identifier name has

visibility and denotes a unique variable. An IDL source file forms a scope. An identifier can only be defined once within a scope.

SCS See Security Service.

Secure Sockets Layer (SSL)
A security protocol that allows the client to authenticate the server and all data and requests to be encrypted. The Secure Sockets Layer was developed by Netscape Communications Corp. and RSA Data Security, Inc.

security name
The identity of a principal used to authenticate that principal to the security system. A set of security attributes are associated with a principal through the principal's security name, including the principal's access identity, audit identity, and privileges. The security name is often referred to as a user ID.

Security Service
A collection of security services provided by Component Broker.

server (1) A functional unit that provides services to one or more clients over a network. Examples include a file server, a print server, and a mail server. (2) In the AIX operating system, an application program that usually runs in the background and is controlled by the system program controller. (3) In the Enhanced X-Windows Toolkit, a program that provides the basic window mechanism. It handles interprocess communication (IPC) connections from clients, de-multiplexes graphics requests onto screens, and multiplexes input back to clients. (4) In OS/390 Component Broker, a logical grouping of replicated server instances; that is, all server instances in a server are identical. See server instance.

server adaptor
An adaptor that participates in dispatching methods within the CB Server process. It establishes the environment in which a method request executes. The adaptor determines the threading model that the server process is using.

server control interface
The interface used by agents to start up and close down a Component Broker server and the server start-up daemon.

server group control point (SGCP)
A server found on the managing host of a controlled server group and coordinates the exchange of information between clients and application servers. The SGCP allows clients to know which servers in the group are active.

server group gateway (SGGW)
A gateway that routes requests from clients without the workload management extension to a particular server in a controlled server group. The server group gateway appears to the clients as if it is the target server for a request.

server instance
In OS/390 Component Broker, a functional component on which Component Broker applications run. A server instance has two kinds of address spaces: a control region and one or more server regions. Replicas of server instances are logically grouped into a server. See server.

server main
The main program that executes when the Component Broker server is started.

server object
An object that responds to a request for a service. A given object may be a client for some requests and a server for other requests.

server programming model
An interface model of the Component Broker business object. This interface model for the Component Broker business object can be used by client application developers to build business objects based on the managed object framework. The business objects that application developers create can be used and reused to create client/server applications.

server region
In OS/390 Component Broker, one of two kinds of address spaces that make up a server instance. Application server code runs in a server region. A server region can be replicated based on the workload demands of the system. See control region.

service context
The information that flows from the client to the server or from the server to the client. The information is used to inform the server of the context running on the client; for any service, the appropriate behavior on the server is defined by the context of the client.

service control interface
The interface used by agents to obtain non-definition type information from running servers and services. Definition type information is obtained from the Common Data Store.

session bean
Relatively short-lived enterprise beans. There are two types of session beans: stateful session beans, and stateless session beans. See stateful session bean and stateless session bean.

SGCP See server group control point.

SGGW
See server group gateway.

SGML
See Standard Generalized Markup Language.

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.

simple name
A name in a naming context that identifies a particular name-binding. A simple name is normally composed of an ID field and a kind field. Also referred to as a name-component. Contrast with compound name.

single-byte character set (SBCS)
A character set in which each character is represented by a one-byte code. Contrast with double-byte character set.

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

skeleton
The object-interface-specific ORB component that assists an object adaptor in passing requests to particular methods.

smart proxy
A piece of code that represents a server object on the client side and runs in the same address space as the client. Smart proxies enable you to add filters to a regular proxy so you can customize the behavior of the client to suit your needs.

special framework method
In Object Builder, a del(), insert(), retrieve(), setConnection(), or update() method that is implemented for the data object to act on the underlying application data or procedural system.

specialized home
A customized form of home. You can create a customized home in Object Builder by creating a customized file, business object interface, business object implementation, and managed object.

SQL See Structured Query Language.

SQL DDL
A language containing data definition statements that describe the organization of data and their relationships in a database.

SQL View Editor
A tool provided with Object Builder that enables the user to build SQL views from schemas that are either imported into Object Builder or created using Object Builder. This tool can also be used to edit existing views.

SQX file
A C++ file with embedded SQL. This file is converted into the package file, which is bound to the database being accessed.

SSL See Secure Sockets Layer.

Standard Generalized Markup Language (SGML)
An international standard for the description of marked-up electronic text. SGML is a metalanguage, that is, a means of formally describing a language, in this case, a markup language. See markup.

state An object's characteristic that is manifested in its public and private data members, and can be divided into two categories: essential state and non-essential state. See state data and behavior.

state data
The manifestation of the object's state in its private and public data members. It consists of data for an object that is persistent, and not calculated or derived from other data members.

stateful session bean
A session bean that acts on behalf of a single client and maintains client-specific session information across multiple method calls and transactions. This type of bean exists for the duration of a single client/server session. Contrast with stateless session bean.

stateless session bean
A session bean that does not maintain any conversational state. Stateless session beans are pooled by their container to handle multiple requests from multiple clients. Contrast with stateful session bean.

static invocation
The construction of a request at compile time. Calling an operation through a stub procedure.

static SQL
See embedded SQL.

stored procedures
In DB2, a block of procedural constructs and embedded SQL statements that is stored in a database and can be called by name. Stored procedures allow an application program to be run in two

parts. One part runs on the client; the other on the server. Stored procedures allow one call to produce several accesses to the database. See procedure.

stream
The client interface of the Externalization Service for externalizing and internalizing objects. The stream object uses the StreamIO and Streamable interfaces. It manages the creation of streamable objects and restores references.

Structured Query Language (SQL)
A standardized language for defining and manipulating data in a relational database.

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

superclass
See base class and abstract class.

swizzling (of pointers)
The act of taking an abstract object and making it behave like an in-memory pointer. It also converts an in-memory object pointer to a form suitable for persistent storage. When required, the data object can be converted back into an in-memory pointer. Swizzling of pointers thus ensures that an object can be referenced.

symbol
Any of a set of names that are used as place holders when building a text template to pattern the desired emitter output. When a template is emitted, the symbols are replaced with their corresponding values from the emitter symbol table.

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

sysplex
A set of OS/390 systems communicating and cooperating with each other through certain multi-system hardware components and software services to process customer workloads.

System Management (SM)
Part of Component Broker that provides you with the ability to install, configure, deploy, monitor, and control most aspects of a distributed object solution built using Component Broker development components, frameworks, and tools. System Management consists of several tools and components that provide easy to use management services for Component Broker servers and deployed applications.

System Management agent
The component on a host by which the System Manager communicates with the servers and other objects to be managed on that host. The System Management agent interacts directly with the managed objects, passing data back to the System Manager and acting on the managed objects on behalf of the System Manager. A System Management agent has its own process that runs, as an NT service or AIX resource, separate from the servers that it controls.

System Management DDL
A language used by System Management that defines the structure of an application on both client and server.

System Management User Interface (SMUI)
A graphical user interface that provides front-ends to all of the Component Broker System Management components. Typically the System Management user interface is introduced to object services in the administrable interface. It is intended to be used by system administrators.

System Manager
The component that provides the logic to manage part or all of an enterprise. A System Manager relates the definition of the enterprise (data stored in its central configuration data) to the servers, clients, and applications within the real enterprise (data provided by System Management agents on managed hosts). It composes the data defined by the Component Broker System Management and administers your enterprise with the data that is determined from the System Management agent. It presents that data through the System Management User Interface to which it is connected. Through the user interface to a System Manager, you can operate servers and applications on the managed hosts and can administer those servers and applications in management zones defined in the central configuration data. A System Manager has its own process that runs as an NT service.

T

target object
The object responding to a method call. The target object is always the first formal parameter of a method procedure. For Component Broker C-language bindings, the target object is the first argument provided to the method invocation macro, _methodName().

template
Pseudocode generated by an automated computer-aided software engineering (CASE) system and requiring further hand-coding before compilation.

tier-1 In a logical three-level client/server enterprise system, tier-one is the client level which contains clients and client interfaces. See tier-2 and tier-3.

tier-2 In a logical three-level client/server enterprise system, tier-two is the server level, where the Component Broker object server and development components are housed. This tier is also referred to as the "middle-tier". See tier-1 and tier-3.

tier-3 In a logical three-level client/server enterprise system, tier-three is the database or legacy system level, where enterprise resource managers and historical data are located. See tier-1 and tier-2.

trace log
The log file used to record debug trace information in Component Broker code at run time. This debug information contains execution path and data information. Separate controls are provided for each Component Broker component, and varying levels of detail can be gathered. This service is for use exclusively by or on behalf of IBM Service personnel. Copies of the error log and activity log entries may also be added to the trace logs when those entries are generated while trace is active.

transaction processing facility (TPF)
A high-availability, high-performance system, designed to support real-time, transaction driven applications. The specialized architecture of TPF is intended to optimize system efficiency, reliability, and responsiveness for data communication and database processing. TPF provides real-time inquiry and update to a large, centralized database, where message length is relatively short in both directions, and response time is generally less than three seconds. Formerly known as the Airline Control Program/Transaction Processing Facility (ACP/TPF).

transaction object
(1) An object whose behavior is affected by being invoked within the scope of a transaction. (2) In VisualAge for Java with Enterprise Access Builder support, a container that encapsulates the sequence of screen panels a user would navigate to complete a CICS or IMS transaction. All panel states, input fields, and output fields are modeled in the transaction object.

transaction record
In VisualAge for Java with Enterprise Access Builder support, an element of the transaction object. One transaction record models a single panel state in a CICS or IMS transaction, including all input and output fields on that panel.

transaction server
A server containing one or more transaction objects.

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

tuple In a relational database, a part of a relation that uniquely describes an entity and its attributes. A tuple can be represented by one row of a relation table.

U

UDM See user-defined mapping.

UI meta model
A specialized meta model that describes a user interface.

UI model
The set of elements created by using a user interface. For example, a UI model that has a single entry field would have a UI model that contained a single element, which would be a string with the same name as the field.

UI script
XML instance data that describes how a user interface must render data and relate to a UI model.

universally unique identifier (UUID)
A value constructed with an algorithm that provides a reasonable assurance that the identity value is unique within the known universe. Typically, a universally unique identity is 16 bytes long.

usage bindings
The language-specific binding files for a class that are generated by the Component Broker compiler for inclusion in client programs using the class.

user-defined mapping (UDM)
Any mapping that is defined by the user rather than built into the language.

UUID See universally unique identity.

V

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

view object
A view object is responsible for providing a visual presentation on the

client side for a business object on the server side. The view object interfaces are generated directly from server-side objects.

VisualAge Component Development package
(1) Tools that provide visual and code generation support to help you develop applications using the MOFW client programming model and MOFW server programming model. These tools speed up application development by providing component-based development and object reusability. (2) The package that contains the run-time and development environments.

W

wellness checks
The Component Broker server is responsible for reporting various statistics as it runs. These statistics are called wellness checks and include performance related information such as processor utilization, or incoming work rate. This information is retrieved from the various components in the Component Broker server and communicated with the system management tools.

workflow
The sequence of activities performed in accordance with the business processes of an enterprise.

workflow model
A complete representation, from start to finish, of business processes.

workgroup model
A type of model object that defines a logical grouping of host computers that are represented by host models. This allows Component Broker System Management to manage as a unit a number of host computers in which the hosts are logically related, typically by the business or geographical area that they support. On one of the hosts in the workgroup there is a server that provides a dedicated Naming Service for the workgroup. That naming server resolves all requests by name for hosts, servers, and other objects in the workgroup. Hosts in a workgroup can also be members of a cell. Therefore, you can have an enterprise that comprises a number of cells that group hosts similar to Windows NT or DCE cells for general administration, and several workgroups that group the same hosts into business areas for administration along business lines.

workload management
Enables the operational environment to manage active business object instances across multiple CB servers. It minimizes client complexity by having a single system image and single object image for objects across groups of CB servers.

X

XML See Extensible Markup Language.

XML instance
An element in an XML file, which is an instance of an XML type that is described in a DTD file.

XML instance data
The instance of a particular XML tag in an .xml file.

XML type
The description of an element type in XML, in a document type definition format that is contained in a DTD file.

Notices

This information was developed for products and services offered in the U.S.A. IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

IBM World Trade Asia Corporation Licensing
2-31 Roppongi 3-chome, Minato-ku
Tokyo 106, Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law:

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS DOCUMENT "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OR CONDITIONS OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the document. IBM may make

improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

For Component Broker:
IBM Corporation
Department LZKS
11400 Burnet Road
Austin, TX 78758
U.S.A.

For TXSeries:
IBM Corporation
ATTN: Software Licensing
11 Stanwix Street
Pittsburgh, PA 15222
U.S.A.

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM International Program License Agreement or any equivalent agreement between us.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurements may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available

sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

All statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples may include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

If you are viewing this information softcopy, the photographs and color illustrations may not appear.

Trademarks and service marks

The following terms are trademarks or registered trademarks of the IBM Corporation in the United States, other countries, or both:

AFS
AIX
AS/400
CICS
CICS OS/2
CICS/400
CICS/6000
CICS/ESA
CICS/MVS
CICS/VSE
CICSPlex
DB2
DCE Encina Lightweight Client
DFS
Encina
IBM
IMS
MQSeries
MVS/ESA
OS/2
OS/390
OS/400
PowerPC
RISC System/6000
RS/6000
S/390
Transarc
TXSeries
VSE/ESA
VTAM
VisualAge
WebSphere

Microsoft, Windows, Windows NT, and the Windows 95 logo are trademarks of Microsoft Corporation in the United States and/or other countries.

Oracle and Oracle8 are registered trademarks of the Oracle Corporation in the United States and/or other countries.

UNIX is a registered trademark of The Open Group in the United States and/or other countries licensed exclusively through X/Open Company Limited.

OSF and Open Software Foundation are registered trademarks of the Open Software Foundation, Inc.

* HP-UX is a Hewlett-Packard branded product. HP, Hewlett-Packard, and HP-UX are registered trademarks of Hewlett-Packard Company.

Orbix is a registered trademark and OrbixWeb is a trademark of IONA Technologies Ltd.

Sun, SunLink, Solaris, SunOS, Java, all Java-based trademarks and logos, NFS, and Sun Microsystems are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States and/or other countries.

Some of this documentation is based on material from Object Management Group bearing the following copyright notices:

Copyright 1995, 1996 AT&T/NCR
Copyright 1995, 1996 BNR Europe Ltd.
Copyright 1991, 1992, 1995, 1996 by Digital Equipment Corporation
Copyright 1996 Gradient Technologies, Inc.
Copyright 1995, 1996 Groupe Bull
Copyright 1995, 1996 Expersoft Corporation
Copyright 1996 FUJITSU LIMITED
Copyright 1996 Genesis Development Corporation
Copyright 1989, 1990, 1991, 1992, 1995, 1996 by Hewlett-Packard Company
Copyright 1991, 1992, 1995, 1996 by HyperDesk Corporation
Copyright 1995, 1996 IBM Corporation
Copyright 1995, 1996 ICL, plc
Copyright 1995, 1996 Ing. C. Olivetti &C.Sp
Copyright 1997 International Computers Limited
Copyright 1995, 1996 IONA Technologies, Ltd.
Copyright 1995, 1996 Itasca Systems, Inc.
Copyright 1991, 1992, 1995, 1996 by NCR Corporation
Copyright 1997 Netscape Communications Corporation
Copyright 1997 Northern Telecom Limited
Copyright 1995, 1996 Novell USG
Copyright 1995, 1996 02 Technolgies
Copyright 1991, 1992, 1995, 1996 by Object Design, Inc.
Copyright 1991, 1992, 1995, 1996 Object Management Group, Inc.
Copyright 1995, 1996 Objectivity, Inc.
Copyright 1995, 1996 Oracle Corporation
Copyright 1995, 1996 Persistence Software

This software contains RSA encryption code.

Printed in the United States of America
on recycled paper containing 10%
recovered post-consumer fiber.

SC09-4450-00

Spine information:

WebSphere Glossary Version 3.0 SC09-4450-00