

Experiment No. 11

Aim: Case Study: CORBA.

Theory:

Common Object Request Broker Architecture (CORBA)

The Common Object Request Broker Architecture (CORBA) is a standard defined by the Object Management Group (OMG) that enables software components written in multiple computer languages and running on multiple computers to work together.

CORBA is a standard for distributing objects across networks so that operations on those objects can be called remotely. CORBA is not associated with a particular programming language, and any language with a CORBA binding can be used to call and implement CORBA objects. Objects are described in a syntax called Interface Definition Language (IDL).

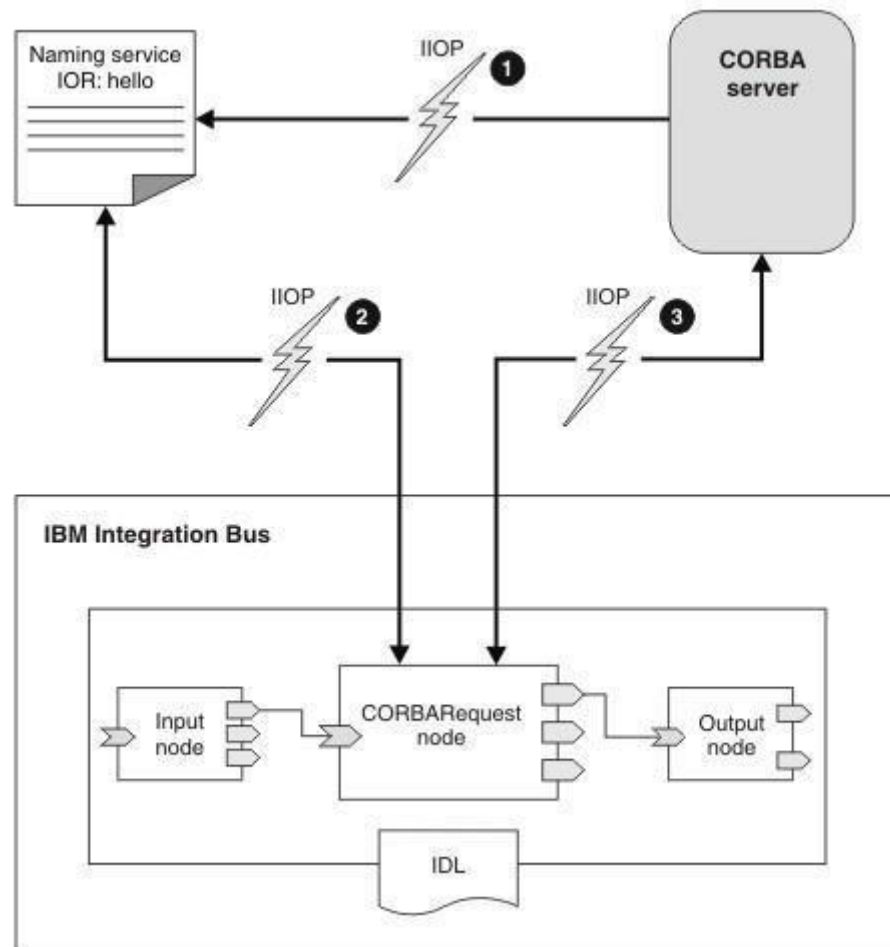
CORBA includes four components:

- **Object Request Broker (ORB)**
The Object Request Broker (ORB) handles the communication, marshaling, and unmarshaling of parameters so that the parameter handling is transparent for a CORBA server and client applications.
- **CORBA server**
The CORBA server creates CORBA objects and initializes them with an ORB. The server places references to the CORBA objects inside a naming service so that clients can access them.
- **Naming service**
The naming service holds references to CORBA objects.
- **CORBARequest node**
The CORBARequest node acts as a CORBA client.

The following diagram shows the layers of communication between IBM® Integration Bus and CORBA.

The diagram illustrates the following steps.

1. CORBA server applications create CORBA objects and put object references in a naming service so that clients can call them.
2. At deployment time, the node contacts a naming service to get an object reference.
3. When a message arrives, the node uses the object reference to call an operation on an object in the CORBA server.



CORBA nodes

CORBA is a standard for distributing objects across networks so that operations on those objects can be called remotely. CORBA objects are described in Interface Definition Language (IDL) files, and these IDL files are used to configure the CORBA message flow nodes. The IDL file is stored in a message set project, in a folder called CORBA IDLs.

An IDL importer imports the IDL file into the message set project and creates the message definition file (.mxsd) in the message set. This message definition file is used for mid-flow validation, ESQL content assist, and the Mapping node.

For each IDL file, a single message definition is created. In the message definition, two messages are created for each operation in the IDL file: one message for the request, and one for the response. The request has a child element for each in and inout parameter; the response has a child element for each input and out parameter, and a child element named `_return` for the return type of the operation.

The name of these elements is based on the interface name and operation name; for example, for the operation `sayHello` in the Interface `Hello`, the request element is called `Hello.sayHello`, and the response element is called `Hello.sayHelloResponse`. If the interface is contained in a module, the request and response element names are qualified with the

names of the modules. For example, if the operation `sayHello` in the Interface `Hello` is contained in `ModuleB`, which in turn is contained in `ModuleA`, the response element would be called `ModuleA.ModuleB.Hello.sayHelloResponse`.

When you add a message flow that contains CORBA nodes to a BAR file, all the IDL files that are used by the nodes are added to the BAR file automatically.

CORBA naming service

A CORBA naming service holds CORBA object references.

A CORBA server puts references to CORBA objects inside a naming service so that clients can query the naming service and obtain the object reference, then call operations on the CORBA objects. Typically, a client queries the naming service once, then caches the object reference.

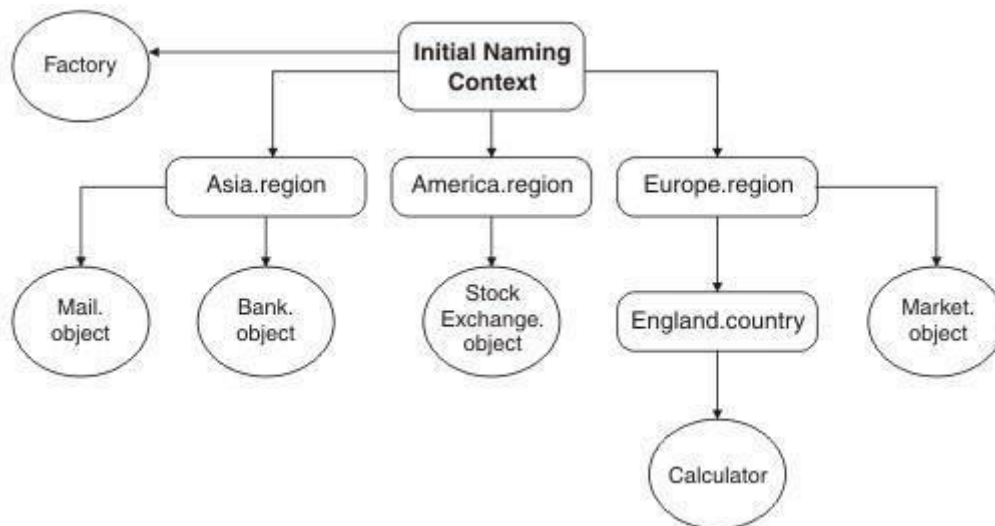
A `CORBARequest` node is a CORBA client; therefore, when it is deployed, the node contacts a naming service to obtain an object reference. If the object reference is not in the naming service at deployment time, or the naming service that is configured on the node is unavailable, the `CORBARequest` node issues a warning, and attempts to contact the naming service to get the object reference when it receives a message. If an object reference cannot be acquired from the naming service when the node receives a message, an error is issued. You can specify the location of an object reference by using the properties on the `CORBARequest` node, or by using the CORBA configurable service. For more information, see `CORBARequest` node and `Defining where the CORBARequest node gets the object reference`.

Identifying an object reference in a naming service

Each object in a naming service has a unique name. You must use this name when you configure the `Object reference name` property on the `CORBARequest` node.

Naming services are typically arranged in a hierarchy so that names can be given context or scope. The initial naming context is at the top of the hierarchy. Object references can be added to the initial naming context, and additional contexts can exist below it. The number of levels in the hierarchy is unlimited.

Object references and contexts can be assigned a kind to facilitate grouping. The kind is appended to the context in the format `context.kind`. If you are using IBM® Integration Bus to access an external CORBA application, you need to know the location of the naming service and the name of the object reference in the naming service. The following example shows how to determine the exact string representation of the name.



In the diagram, contexts are represented by squares, and object references are represented by circles. • An object called Factory is directly attached to the initial naming context.

- Three contexts, with kind region, are also attached to the initial naming context.
- These three contexts each have one or more object references attached to them.
- The Europe context has an England context attached to it, of kind country, which has an object attached to it (Calculator).

The name that you specify when you configure the Object reference name property on the CORBARequest node reflects the position of the object in the hierarchy. The following table shows how to refer to the specific objects in the diagram.

Object	Object reference name
Factory	Factory
Bank	Asia.region/Bank.object
Mail	Asia.region/Mail.object
StockExchange	America.region/StrockExchange.object
Market	Europe.region/Market.object
Calculator	Europe.region/England.country/Calculator

All objects in the naming service can be connected directly to the initial naming context; in which case, their names would be in the same format as the Factory object in this example.

CORBA Request node

Use the CORBA Request node to call an external CORBA application over Internet Inter-Orb Protocol (IIOP).

This topic contains the following

sections: • Purpose

- Using this node in a message flow
- Configuring the CORBARequest node • Terminals and properties

Purpose

You can use the CORBA Request to connect IBM® Integration Bus to CORBA applications. CORBA is a standard for distributing objects across networks so that operations on those objects can be called remotely. CORBA objects are described in Interface Definition Language (IDL) files. You can create a message flow that contains a CORBA Request node, which calls a CORBA server. The message flow uses the IDL file to configure which operation is called on which interface. By using a message flow that includes a CORBA Request node, you can give existing CORBA applications a new external interface; for example, a SOAP interface. The IDL file is stored in a message set project inside a folder called CORBA IDLs, and is used to configure the CORBA Request node in the message flow.

Using this node in a message flow

One possible use of a CORBA Request node is to connect a SOAP-based Web service application to an existing CORBA IIOP application by using a synchronous style of message flow. You can achieve this connection by creating the following message flow:



In this example, the SOAPInput node receives a Web service request, the Mapping node transforms the data in the SOAP message to a CORBA request, and a request is made to the CORBA server. The second Mapping node transforms the response message back into a SOAP reply, which is propagated by the SOAPReply node.

The CORBA Request node is not transactional. After the node has made a request, it cannot roll back the request. The CORBA nodes use the DataObject domain.

The CORBA Request node is contained in the CORBA drawer of the message flow node palette, and is represented in the IBM Integration Toolkit by the following icon:



Look at the following sample to see how to use this node:

- CORBA nodes

You can view information about samples only when you use the product documentation that is integrated with the IBM Integration Toolkit or the online product documentation.

You can run samples only when you use the product documentation that is integrated with the IBM Integration Toolkit.

Configuring the CORBA Request node

When you have put an instance of the CORBA Request node into a message flow, you can configure it; see [Configuring a message flow node](#). The properties of the node are displayed in the Properties view.

All mandatory properties for which you must enter a value (properties that do not have a default value defined) are marked with an asterisk.