

Distributed Development Using CORBA

Duration:	3 days
Type:	intermediate

Description

This course introduces Java programmers to CORBA development. Delegates create distributed applications written in Java and then access them through both Java and C++ clients. The process of writing IDL is explained in detail, as are the mappings from IDL to Java and C++.

By default the course uses the ORB supplied with the Sun JDK and the Eclipse IDE. Other configurations can be supported on request.

Prerequisites

Delegates should have several years Java coding experience, understand the principles of good OO design and be familiar with UML class and sequence diagrams. Previous experience in distributed computing (RPC, DCOM, RMI or Remoting) is advantageous but not essential.

List of Modules

Introduction

- The need for distributed architectures
- How CORBA emerged from UNIX RPC
- The role of an Object Request Broker (ORB)
- Network protocols used by ORB's (GIOP and IIOP)
- Common terminology used in distributed systems
- How IDL supports clients written in many languages
- Introducing CORBA Services (Naming and Trading)

CORBA and Java

- Java as the ideal middleware language
- Examples of ORB's written in Java
- Support for CORBA within the JVM
- How Enterprise JavaBeans use CORBA

Working with Interface Definition Language (IDL)

- Creating and reopening modules
- Adding interfaces to modules
- Data types defined within IDL
- Forward declarations of interfaces
- Adding attributes to interfaces
- Adding operations to interfaces
- Deriving one interface from another
- Marking operations as asynchronous
- Declaring exceptions thrown by operations
- Supplying extra context information
- Mappings between IDL and Java
- Mappings between IDL and C++

Creating a Basic CORBA Server

- Creating and initializing an ORB
- Instantiating server side objects
- Connecting objects to the ORB
- Starting a request handling loop

Creating a CORBA Client

- Creating a client side ORB
- Understanding client side proxies
- Operations supported by all proxies
- Recovering references from strings
- Making copies of references
- Creating and using nil references
- Invoking operations on the server
- Coping with system and user exceptions

The Portable Object Adapter

- The responsibilities of a POA
- References, identifiers and servants
- Services offered by the root POA
- Creating a hierarchy of portable adapters
- Dynamically creating a portable adapter
- Policies that can be used with a POA
- Persistent versus transient objects
- Creating CORBA objects without servers
- Explicitly registering servants with the POA
- Dynamically supplying servants via managers
- Providing a default servant for detached objects
- Deactivating objects and closing the ORB

The OMG Naming Service

- The purpose of the Naming Service
- Finding an initial naming context
- Adding bindings to the naming tree
- Creating trees of contexts and bindings
- Iterating over a naming context