NAME

orbd – Enables clients to locate and call persistent objects on servers in the CORBA environment.

SYNOPSIS

orbd [*options*]

options Command-line options. See Options.

DESCRIPTION

The **orbd** command enables clients to transparently locate and call persistent objects on servers in the CORBA environment. The Server Manager included with the orbd tool is used to enable clients to transparently locate and call persistent objects on servers in the CORBA environment. The persistent servers, while publishing the persistent object references in the naming service, include the port number of the ORBD in the object reference instead of the port number of the server. The inclusion of an ORBD port number in the object reference for persistent object references has the following advantages:

- The object reference in the naming service remains independent of the server life cycle. For example, the object reference could be published by the server in the Naming Service when it is first installed, and then, independent of how many times the server is started or shut down, the ORBD returns the correct object reference to the calling client.

- The client needs to look up the object reference in the naming service only once, and can keep reusing this reference independent of the changes introduced due to server life cycle.

To access the ORBD Server Manager, the server must be started using **servertool**, which is a command-line interface for application programmers to register, unregister, start up, and shut down a persistent server. For more information on the Server Manager, see Server Manager.

When **orbd** starts, it also starts a naming service. For more information about the naming service. See Start and Stop the Naming Service.

OPTIONS

-ORBInitialPort *nameserverport*

Required. Specifies the port on which the name server should be started. After it is started, **orbd** listens for incoming requests on this port. On Oracle Solaris software, you must become the root user to start a process on a port below 1024. For this reason, Oracle recommends that you use a port number above or equal to 1024.

NONREQUIRED OPTIONS

-port *port*

Specifies the activation port where ORBD should be started, and where ORBD will be accepting requests for persistent objects. The default value for this port is 1049. This port number is added to the port field of the persistent Interoperable Object References (IOR).

-defaultdb *directory*

Specifies the base where the ORBD persistent storage directory, **orb.db**, is created. If this option is not specified, then the default value is **./orb.db**.

-serverPollingTime *milliseconds*

Specifies how often ORBD checks for the health of persistent servers registered through **servertool**. The default value is 1000 ms. The value specified for **milliseconds** must be a valid positive integer.

-serverStartupDelay milliseconds

Specifies how long ORBD waits before sending a location forward exception after a persistent server that is registered through **servertool** is restarted. The default value is 1000 ms. The value specified for **milliseconds** must be a valid positive integer.

-J*option*

Passes **option** to the Java Virtual Machine, where **option** is one of the options described on the reference page for the Java application launcher. For example, **-J-Xms48m** sets the startup

memory to 48 MB. See java(1).

START AND STOP THE NAMING SERVICE

A naming service is a CORBA service that allows CORBA objects to be named by means of binding a name to an object reference. The name binding can be stored in the naming service, and a client can supply the name to obtain the desired object reference.

Before running a client or a server, you will start ORBD. ORBD includes a persistent naming service and a transient naming service, both of which are an implementation of the COS Naming Service.

The Persistent Naming Service provides persistence for naming contexts. This means that this information is persistent across service shutdowns and startups, and is recoverable in the event of a service failure. If ORBD is restarted, then the Persistent Naming Service restores the naming context graph, so that the binding of all clients' and servers' names remains intact (persistent).

For backward compatibility, **tnameserv**, a Transient Naming Service that shipped with earlier releases of the JDK, is also included in this release of Java SE. A transient naming service retains naming contexts as long as it is running. If there is a service interruption, then the naming context graph is lost.

The **-ORBInitialPort** argument is a required command-line argument for **orbd**, and is used to set the port number on which the naming service runs. The following instructions assume you can use port 1050 for the Java IDL Object Request Broker Daemon. When using Oracle Solaris software, you must become a root user to start a process on a port lower than 1024. For this reason, it is recommended that you use a port number above or equal to 1024. You can substitute a different port when necessary.

To start **orbd** from a UNIX command shell, enter:

orbd −ORBInitialPort 1050&

From an MS-DOS system prompt (Windows), enter:

start orbd −ORBInitialPort 1050

Now that ORBD is running, you can run your server and client applications. When running the client and server applications, they must be made aware of the port number (and machine name, when applicable) where the Naming Service is running. One way to do this is to add the following code to your application:

Properties props = new Properties();
props.put("org.omg.CORBA.ORBInitialPort", "1050");
props.put("org.omg.CORBA.ORBInitialHost", "MyHost");
ORB orb = ORB.init(args, props);

In this example, the naming service is running on port 1050 on host **MyHost**. Another way is to specify the port number and/or machine name when running the server or client application from the command line. For example, you would start your **HelloApplication** with the following command line:

java HelloApplication −ORBInitialPort 1050 −ORBInitialHost MyHost

To stop the naming service, use the relevant operating system command, such as **pkillorbd** on Oracle Solaris, or *Ctrl+C* in the DOS window in which **orbd** is running. Note that names registered with the naming service can disappear when the service is terminated because of a transient naming service. The Java IDL naming service will run until it is explicitly stopped.

For more information about the naming service included with ORBD, see Naming Service at http://docs.oracle.com/javase/8/docs/technotes/guides/idl/jidlNaming.html

SERVER MANAGER

To access the ORBD Server Manager and run a persistent server, the server must be started with **servertool**, which is a command-line interface for application programmers to register, unregister, start up, and shut

down a persistent server. When a server is started using **servertool**, it must be started on the same host and port on which **orbd** is executing. If the server is run on a different port, then the information stored in the database for local contexts will be invalid and the service will not work properly.

See Java IDL: The "Hello World" Example at http://docs.oracle.com/javase/8/docs/technotes/guides/idl/jidlExample.html

In this example, you run the **idlj** compiler and **javac** compiler as shown in the tutorial. To run the ORBD Server Manager, follow these steps for running the application:

Start **orbd**.

UNIX command shell, enter: **orbd -ORBInitialPort 1050**.

MS-DOS system prompt (Windows), enter: **start orbd -ORBInitialPort 1050**.

Port 1050 is the port on which you want the name server to run. The **-ORBInitialPort** option is a required command-line argument. When using Oracle Solaris software, you must become a root user to start a process on a port below 1024. For this reason, it is recommended that you use a port number above or equal to 1024.

Start the **servertool**: **servertool -ORBInitialPort 1050**.

Make sure the name server (**orbd**) port is the same as in the previous step, for example, **-ORBInitialPort 1050.** The **servertool** must be started on the same port as the name server.

In the **servertool** command line interface, start the **Hello** server from the **servertool** prompt:

**servertool > register –server HelloServer –classpath . –applicationName
       HelloServerApName**

The **servertool** registers the server, assigns it the name **HelloServerApName**, and displays its server ID with a listing of all registered servers.Run the client application from another terminal window or prompt:

java HelloClient –ORBInitialPort 1050 –ORBInitialHost localhost

For this example, you can omit **-ORBInitialHost localhost** because the name server is running on the same host as the **Hello** client. If the name server is running on a different host, then use the -**ORBInitialHost nameserverhost** option to specify the host on which the IDL name server is running.Specify the name server (**orbd**) port as done in the previous step, for example, **-ORBInitialPort 1050**. When you finish experimenting with the ORBD Server Manager, be sure to shut down or terminate the name server (**orbd**) and **servertool**. To shut down **orbd** from am MS-DOS prompt, select the window that is running the server and enter *Ctrl+C* to shut it down.

To shut down **orbd** from an Oracle Solaris shell, find the process, and terminate with the **kill** command. The server continues to wait for invocations until it is explicitly stopped. To shut down the **servertool**, type *quit* and press the *Enter* key.

SEE ALSO
- servertool(1)
- Naming Service at http://docs.oracle.com/javase/8/docs/technotes/guides/idl/jidlNaming.html