The omniNames CORBA Naming Service

December 2012

1 Introduction

The OMNI Naming Service (*omniNames*) is an omniORB implementation of the OMG's COS Naming Service Specification. It offers a way for a client to turn a human-readable name into an object reference, on which the client can subsequently invoke operations in the normal way. See the OMG specification for full details of the functionality provided by the Naming Service.

The Naming Service stores a set of bindings of names to objects. These bindings can be arranged as an arbitrary directed graph, although they are often arranged in a tree hierarchy. Each node in the graph is a *naming context*. There is a "root" context at which name lookups usually start. This is the object returned by the call to `CORBA::ORB::resolve_initial_references("NameService")`.

2 'Log' file

The Naming Service is often part of the bootstrapping process of other CORBA programs. For this reason, if an instance of omniNames crashes (or the machine on which it runs is rebooted), it is important that certain aspects of its operation persist upon restarting. Firstly the root context of the Naming Service should always be accessible through the same object reference. This helps the ORB to implement the `resolve_initial_references` call by allowing the object reference to be stored in a configuration file, for example. Secondly, the naming graph with all its bindings should persist between invocations.

To achieve this, omniNames generates a redo log file, to which it writes out an entry every time a change is made to the naming graph. The directory in which this log file is written can be specified with the `OMNINAMES_LOGDIR` environment variable. When omniNames is restarted it uses the log file so that it can regenerate the naming graph.

Periodically the log file is checkpointed, removing unnecessary entries from the log file. The idle time between checkpoints can be set with the `OMNINAMES_ITBC` environment variable. It defaults to 15 minutes.

3 Starting omniNames and configuring omniORB

When starting omniNames for the first time, you can either let it start with the default TCP port of 2809, or you can specify the port number on which it should listen. This is written to the log file so that on subsequent invocations it will listen on the same port number and thus can be accessible through the same object reference. When omniNames starts up successfully it writes the stringified object reference for its root context on standard error.

At startup, omniORB reads its configuration from `omniORB.cfg` or from the Windows registry. Amongst other settings, the configuration contains the object reference to the root context of the Naming Service. This object reference is returned by `resolve_initial_references("NameService")`. There are a number of methods of specifying the root naming context—see the omniORB manual for details.

4 Command line parameters

omniNames accepts the following command line parameters.

`-help` Output usage information.

`-start` *[port]* Start omniNames for the first time, listening on *port*.

`-always` In conjunction with `-start`, always start up omniNames, even if its log file already exists.

`-logdir` *directory* Specifies the directory for the redo log file, overriding the default.

`-errlog` *filename* Causes output that would normally be sent to stderr to go to the specified file instead.

`-nohostname` Normally, omniNames includes the host name in the name of the redo log file. This option disables that, meaning the log file can be used on a different host, or if the host name changes.

`-ignoreport` omniNames normally adds its own endpoint, based on the port specification (given with `-start` or stored in the log file). This option causes it to ignore the port. It should be used in conjunction with specific `-ORBendPoint` options to ensure object references are stable.

`-install` *[port]* On Windows, install omniNames service. See below.

`-manual` On Windows, specify that the service should be started manually. See below.

`-remove` On Windows, uninstall omniNames service. See below.

5 Machines with multiple IP addresses

The CORBA naming service is a tree (or graph) of `NamingContext` CORBA objects. For each of those CORBA objects, the object reference contains details about the *endpoint*—i.e. the host address and port—that is used to contact the object.

When the machine running omniNames has multiple IP addresses, omniORB arbitrarily picks one of the addresses to publish in object references. It might pick the ‘wrong’ one, meaning that clients connect to the configured root context successfully, but then fail to connect to the sub-contexts. To force omniORB to publish the correct IP address, use the `-ORBendPointPublish` command line parameter:

```
-ORBendPointPublish giop:tcp:address:
```

where *address* can be an IP address or a host name.

6 Windows service

omniNames can be run as a Windows service.

To install the service, run with the `-install` command line option, with a port to override the default 2809 if necessary. The command line should include any other parameters of relevance, such as the log directory and error log file. It is particularly important to specify an error log file, since the service runs in an environment where stderr goes nowhere. Similarly, if omniORB tracing is requested with the various `-ORBtrace` options, `-ORBtraceFile` should be specified, otherwise the trace messages vanish.

The service is normally configured to start at system startup time. Specifying `-manual` configures it for manual startup. Regardless of that setting, the service is not automatically started at the time it is installed. It should be started manually with the service control manager if it is to run before the next system restart.

Once installed, the service can be uninstalled by running omniNames with the `-remove` option.