

Getting Started with CORBA in R (and S)

August 29, 2000

Abstract

This provides a basic introduction to using the CORBA package in R and S. For more comprehensive descriptions of some of the features, see also **ShortManual** and **Manual**. We first use R as a client which calls methods in a remote CORBA server. Then we show how to find out about the available servers and the operations they support. And finally, we illustrate how to implement a CORBA server from within an R session using R functions as the server's methods.

1 R as a Client

As with any Client/Server framework, we need both a client and a server to effect a transaction. In this example, we will create a server using one R session and call it from another R session¹. To create this server, run the script **RCorbaServer** in the examples directory of the R CORBA package².

To call the *test* method in the server, we must first get a reference to that server. We do this by reading the identifier for the server from a file to which it was written when the server was created.

```
1a <* 1a>≡ 1b>
    server <- ior("/tmp/RTest.ior")
Uses ior and server.
```

At this point, we are ready to call the *test* method.

```
1b <* 1a>+≡ <1a 1c>
    .Corba(server, "test")
```

The return from this function call should be an integer sequence 1, 2, 3, 4, 5. While this is unexciting, it is powerful. Next, we can call the *runif* method in the server, specifying the number of elements to return.

```
1c <* 1a>+≡ <1b>
    .Corba(server, "runif", 10)
```

Again, while this is suggestive of interesting things, we can get the same results from within the local R session. The real pay-off comes when the server is either

1. running on a different machine (and may be faster or capable of larger computations, or part of a distributed cluster),
2. a different application with functionality not available in the local session.

2 Locating Other Servers

Using files to store the identifiers for different servers is a little amateurish given that we are performing remote method invocations in these servers. CORBA provides a more organized and robust mechanism by which to locate servers. This is the naming service and it works much like a hierarchical file system, with directories and regular entries.

The

¹One can use some of the Java and Omegahat servers as examples also which *may* simplify the process.

²For the curious, this will register some IDL code with the interface repository and then start a non-interactive R session with an R CORBA server which gets registered in the naming server

3 Creating CORBA Servers in R

```
j .CorbaServer("IDL:R/Test:1.0", RTest=list(test=function() return(1:5), runif=function(n)runif(n)))  
<* 1a>  
ior: 1a  
server: 1a
```