Subject: Re: RMI
Posted by zsolt on Sun, 22 Apr 2007 10:25:10 GMT
View Forum Message <> Reply to Message

```
class RemotelyCallableClass
{
public:
 RCallback1<int> doSomething;
};

class ClientSide()
{
 RClient<RemotelyCallableClass> client; //some setup needed
 void doSomethingRemotely()
 {
  client.doSomething(33);
 }
};

class ServerSide()
{
public:
 ServerSide()
 {

  server.doSomething <<= THISBACK(OnDoSomething);
  //some setup (IP, port)
 }
 RServer<RemotelyCallableClass> server;
 void OnDoSomething(int v)
 {
  //some processing on server
 }
};
```

Of course this is an idea only
RCallback arguments should implement a serializable interface if they are not basic types.
The return value could be handled as a callback on the client side with something like RGate<int, int>.
I don't prefer blocking calls.

Edit: I changed RClient to RServer on server side.