Reliable RPC for .NET Wrapper

Writing a Client

All methods for reliable RPC are available on the service class object. See description of class `Service` for details. The methods are:

- Service.SetReliableState
- Service.getReliableState
- Service.ReliableCommit
- Service.ReliableRollback
- Service.GetReliableId
- Service.GetReliableStatus

Example (this example is included as source in folder *examples\ReliableRPC\NetClient*)

Create Broker object and interface object.

```
Mail mail = new Mail();
mail.service.broker.logon();
```

Enable reliable RPC with CLIENT_COMMIT:

```
mail.SetReliableState(Service.ReliableState.RELIABLE_AUTO_COMMIT);
```

The first RPC message.

```
mail.Sendmail("mail receiver", "subject 1", "Text 1");
```

Check the status: get the message ID first and use it to retrieve the status.

```
StringBuilder reliableID = new StringBuilder();
StringBuilder reliableStatus = new StringBuilder();

mail.service.GetReliableID(ref reliableID);
mail.service.GetReliableStatus(reliableID, ref reliableStatus);
Console.Out.WriteLine("Reliable ID = " + reliableID.ToString());
Console.Out.WriteLine("Reliable Status = " + reliableStatus.ToString());
```

The second RPC message.

```
mail.Sendmail("mail receiver", "subject 2", "Text 2");
```

Commit the two messages.

```
mail.service.ReliableCommit();
```

Check the status again for the same message ID.

```
mail.service.GetReliableStatus(reliableID, ref reliableStatus);
Console.Out.WriteLine("Reliable ID = " + reliableID.ToString());
Console.Out.WriteLine("Reliable Status = " + reliableStatus.ToString());
```

The third RPC message.

```
mail.Sendmail("mail receiver", "subject 3", "Text 3");
```

Check the status: get the new message ID and use it to retrieve the status.

```
mail.service.GetReliableID(ref reliableID);
mail.service.GetReliableStatus(reliableID, ref reliableStatus);
Console.Out.WriteLine("Reliable ID = " + reliableID.ToString());
Console.Out.WriteLine("Reliable Status = " + reliableStatus.ToString());
```

Roll back the third message and check status.

```
mail.service.ReliableRollback();
mail.service.GetReliableStatus(reliableID, ref reliableStatus);

Console.Out.WriteLine("Reliable ID = " + reliableID.ToString());
Console.Out.WriteLine("Reliable Status = " + reliableStatus.ToString());

mail.service.broker.logoff();
```

Limitations

1. All program calls that are called in the same transaction (CLIENT_COMMIT) must be in the same IDL library.
2. It is not allowed to switch from CLIENT_COMMIT to AUTO_COMMIT in a transaction.
3. Messages (IDL programs) must have IN parameters only.

Writing a Server

There are no server-side methods for reliable RPC. The server does not send back a message to the client. The server can run deferred, thus client and server do not necessarily run at the same time. If the server fails, it throws an exception. This causes the transaction (unit of work inside the broker) to be cancelled, and the error code is written to the user status field of the unit of work.