YARP RPC ports

RPC stands for "Remote Procedure Call", and in YARP is used to refer to communication that consists of a message and a reply to that message.

The timing of processes is going to become tightly coupled, and a network of processes written this way is much less robust and malleable than with streaming communication.

RPC client expects to connect to a single server, and receive replies on the same connection.

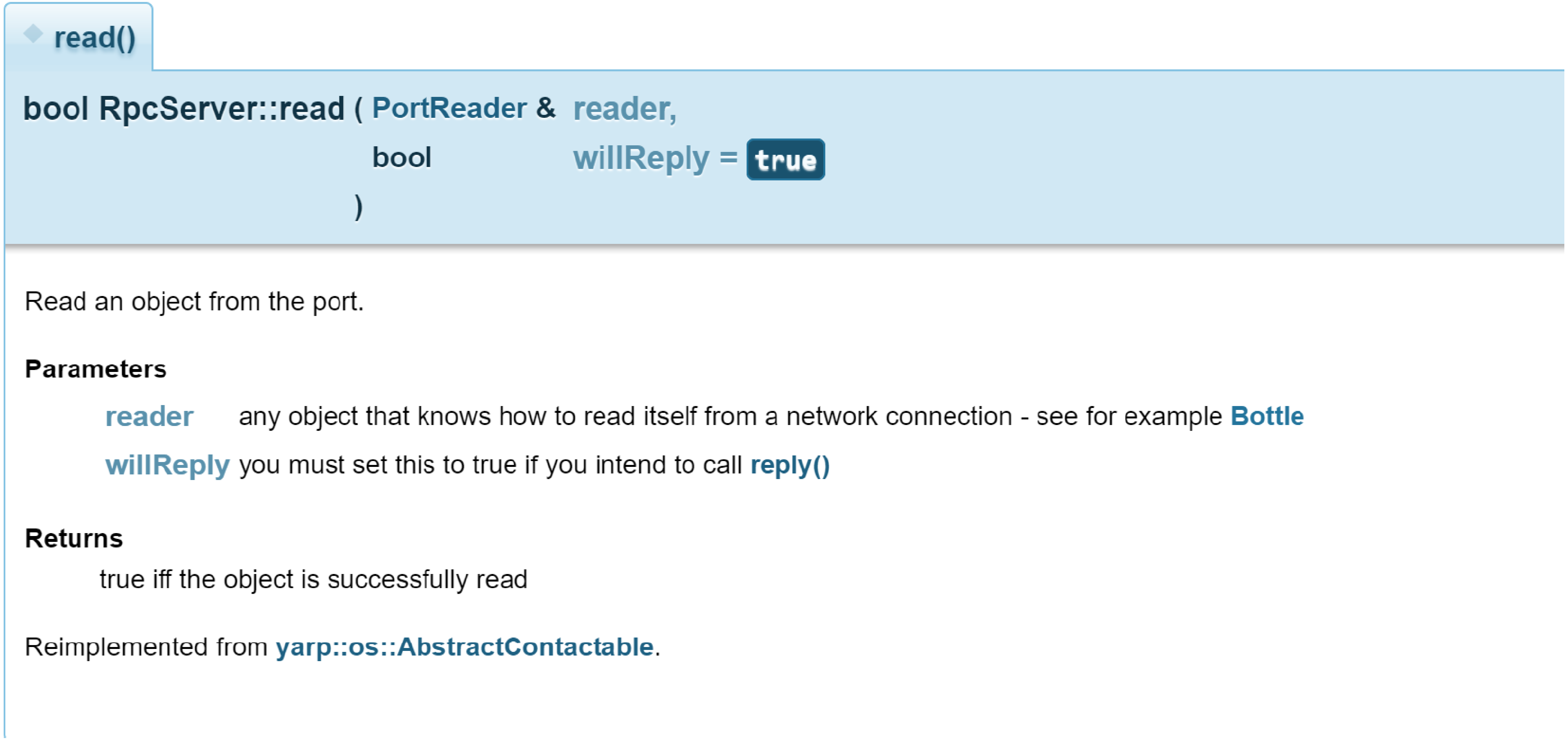
RPC server expects to receive connections from multiple clients, and to reply to them via those connections.

YARP RpcServer

RpcServer is a port that expects to receive connections from multiple clients, and to reply to them via those connections.

```
RpcServer rpcServer;
```

YARP RPCServer::read()



◆ read()

```
bool RpcServer::read ( PortReader & reader,
                       bool         willReply = true
                     )
```

Read an object from the port.

Parameters

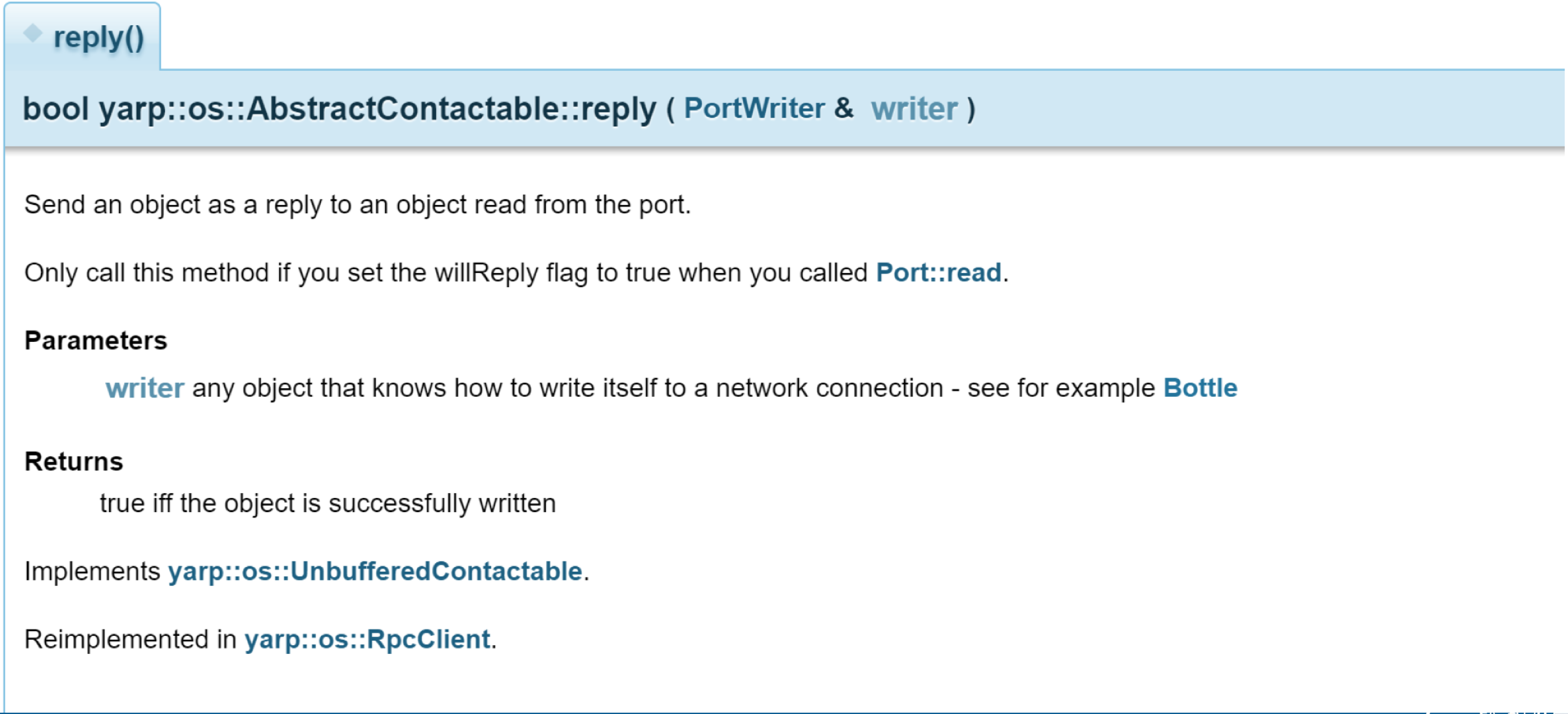
reader	any object that knows how to read itself from a network connection - see for example **Bottle**
willReply	you must set this to true if you intend to call **reply()**

Returns

true iff the object is successfully read

Reimplemented from **yarp::os::AbstractContactable**.

YARP RPCServer::reply()



◆ reply()

bool yarp::os::AbstractContactable::reply (PortWriter & writer)

Send an object as a reply to an object read from the port.

Only call this method if you set the willReply flag to true when you called **Port::read**.

Parameters

writer any object that knows how to write itself to a network connection - see for example **Bottle**

Returns

true iff the object is successfully written

Implements **yarp::os::UnbufferedContactable**.

Reimplemented in **yarp::os::RpcClient**.

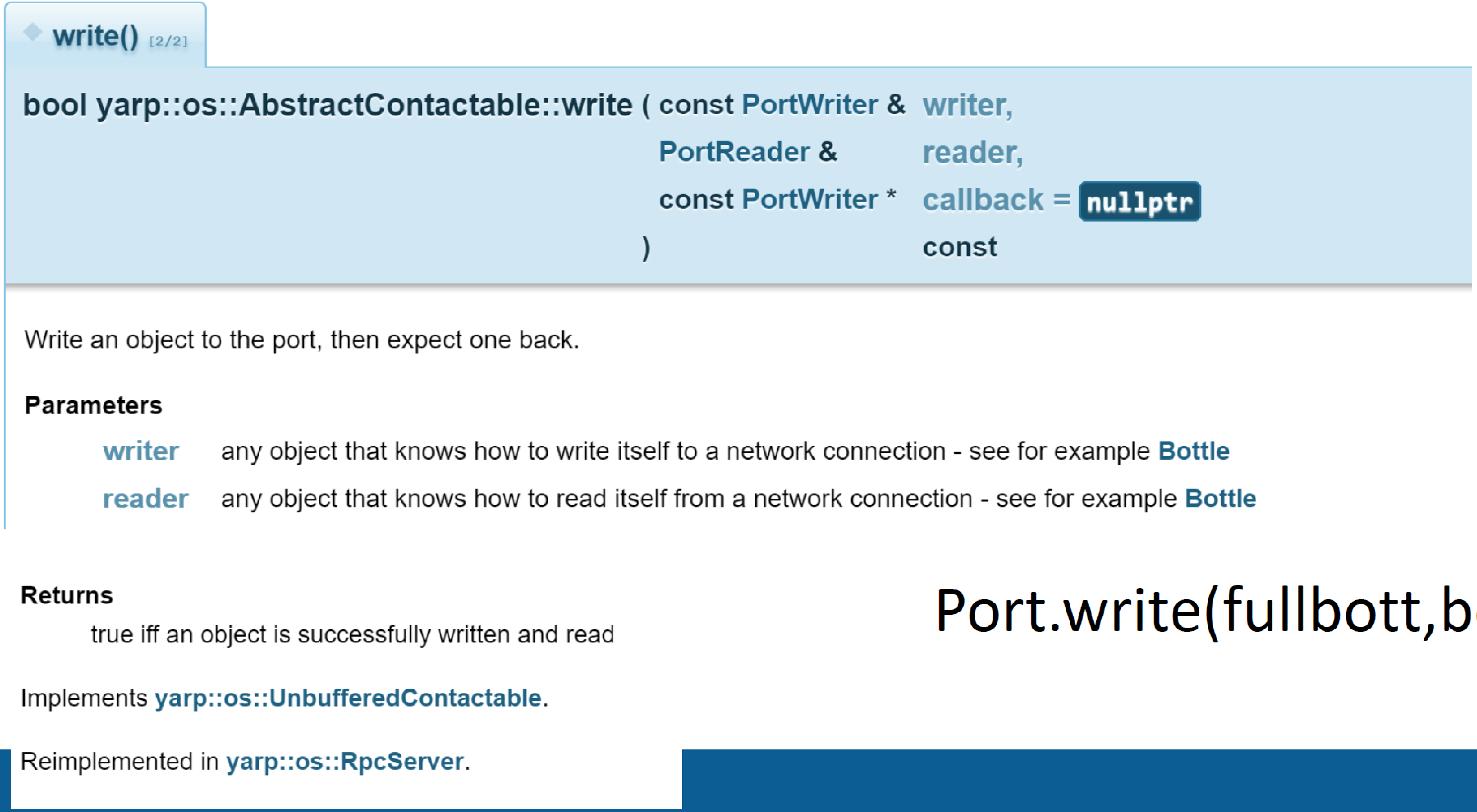
YARP RpcClient

RpcClient is a port that connects to a single server, and receives replies on the same connection.

RpcClient rpcClient;

The connect is -> yarp.connect(client, server)

YARP RpcClient::write()



◆ write() [2/2]

```
bool yarp::os::AbstractContactable::write ( const PortWriter &  writer,
                                            PortReader &        reader,
                                            const PortWriter *  callback = nullptr
                                          )                     const
```

Write an object to the port, then expect one back.

Parameters

writer	any object that knows how to write itself to a network connection - see for example **Bottle**
reader	any object that knows how to read itself from a network connection - see for example **Bottle**

Returns

true iff an object is successfully written and read

Implements **yarp::os::UnbufferedContactable**.

Reimplemented in **yarp::os::RpcServer**.

```
Port.write(fullbott,bott)
```

Excercise

Make a client that connects to the RPC port of the iCub_SIM (/icubSim/head/rpc:i) and send commands to all joints (from 0 to 5) to move the head from -30 to 30 degrees in 60 steps.

```
Bottle req;
req.addString("set");
req.addString(“pos");
req.addInt8(3);
req.addDouble(value));
```