

APS TCPIP Service Provider

User Guide

Version: 1.0.0

Author: Tommy Svensson

Copyright © 2012 Natusoft AB

Table of Contents

1 APSTCPIPService	1
1.1 Security	1
1.2 How it works	1
1.3 Examples	1
1.3.1 TCP	1
1.3.1.1 Write	1
1.3.1.2 Read	1
1.3.2 UDP / Multicast	1
1.3.2.1 Write	2
1.3.2.2 READ	2

1 APSTCPIPService

This service provides, in ways of communication, plain simple TCP/IP communication. Users of this service will however have very little contact with the java.net classes.

The following are the points of this service:

- Simple TCP/IP usage.
- Remove all host, port, and partly protocol from the client code by only referencing a named configuration provided by the service.
- Being able to transparently provide different implementations, like a plain non secure implementation as this is, or an SSL:ed version for TCP. A Test implementation that opens no real sockets nor sends any real packets that can be used by tests are also a possibility.

1.1 Security

This implementation is non secure! It sets the following property on the registered service:

```
aps.props.security=nonsecure
```

1.2 How it works

The service registers an APSConfigService configuration with that service where configurations for TCP, UDP or Multicast connections can be defined. Each configuration entry basically specifies host, port and protocol in addition to a unique name for the entry. Do note that in most cases there needs to be separate entries for clients and services.

The client code should have a configuration of itself that specifies the named entry to use. This name is then passed to the service which then only reads or writes data without having to care where from or to.

1.3 Examples

1.3.1 TCP

1.3.1.1 Write

```
APSTCPIPService tcpipSvc;
...
tcpipSvc.sendTCPRequest("somesvc", new TCPRequest() {
    void tcpRequest(OutputStream requestStream, InputStream responseStream) throws
IOException {
        // write to requestStream ...

        // read from response stream ...
    }
})
```

1.3.1.2 Read

```
APSTCPIPService tcpipSvc;
...
tcpipSvc.setTCPRequestListener("remotesvc", this);
...
void tcpRequestReceived(String name, InetAddress address, InputStream reqStreamn,
OutputStream respStream) throws IOException {
    // Read request from reqStream ...

    // Write response to respStream ...
}
```

1.3.2 UDP / Multicast

Since Multicast uses UDP packets there is no difference between host and port connected UDP or Multicast. The only difference is in the configuration where "UDP" is specified for point to point UDP packets and "Multicast" is specified for multicast packets.

1.3.2.1 Write

```
APSTCPIPService tcpipSvc;  
...  
bytes[] bytes = "Some data".getBytes();  
tcpipSvc.sendUDP("myudptarget", bytes);
```

1.3.2.2 READ

```
APSTCPIPService tcpipSvc;  
...  
byte[] packetBuff = new byte[4000];  
DatagramPacket packet = tcpipSvc.readUDP("myudpsomething", packetBuff);  
byte[] data = packet.getData(); // This is actually packetBuff being returned!
```