Technical Support Guide

v1.0

Telnet Connections May Require a Handshake

13-Feb-20

Most devices communicate in a raw telnet session without any initial handshake involved. Other devices such as Cisco Codecs and BiAmp Tesira require a handshake to open a telnet session. NetLinx controllers establish a raw connection and therefore the telnet handshake must be handled in NetLinx code. Most telnet clients quietly perform a handshake in the background. With Putty you can establish a raw connection as a test to determine if a handshake is required.

In a handshake, there are all types of negotiation options (i.e Echo, Binary Transmission, Reconnection). In most cases the client (in our case a NetLinx Controller) needs to simply respond with either a "DON'T" or "WON'T" based on each request. When a "WILL" ($FB) is received the response would be "DON'T" ($FE). When a "DO" ($FD) is received the response would be "WON'T" ($FC).

For example, the following is received upon the initial telnet connection:

```
$FF$FD$18  $FF$FD$20  $FF$FD$23  $FF$FD$27
```

The response would be:

```
$FF$FC$18  $FF$FC$20  $FF$FC$23  $FF$FC$27
```

Byte 1: $FF is the IAC (Interpret as Command)

Byte 2: $FB (WILL), $FD (DO), $FE (DON’T), or $FC (WON’T)

Byte 3: is the option. (echo, binary, reconnect, etc.)

Here are some great resources:

support.biamp.com/Tesira/Control/Telnet_session_negotiation_in_Tesira

www.iana.org/assignments/telnet-options

Example code to handle this:

```
DEFINE_DEVICE
dvDevice= 0:3:0

DEFINE_VARIABLE

volatile integer telnet_complete
volatile integer login_complete

DEFINE_EVENT

DATA_EVENT[dvDevice]
{
STRING:
{
 STACK_VAR CHAR         sBuild[100]
 STACK_VAR INTEGER     index

IF(!telnet_complete)
{
        IF(FIND_STRING(data.text,"$FF",1))
        {
                sBuild = data.text
                index = FIND_STRING(sBuild,"$FF",1)

                WHILE(index)
                {
                        IF(sBuild[index+1] = "$FD") sBuild[index+1] =type_cast("$FC") //FD > FC (DO > WON'T)
                        IF(sBuild[index+1] = "$FB") sBuild[index+1] = type_cast("$FE") //FB > FE (WILL > DON'T)
                        index = FIND_STRING(sBuild,"$FF",index+1)
                }
                SEND_STRING data.device,sBuild
        }
        ELSE ON[telnet_complete]
}

IF(telnet_complete AND !login_complete)  //example login
{
        IF(FIND_STRING(data.text,'ogin successful',1)) ON[login_complete]
        ELSE IF(FIND_STRING(data.text,'ogin:',1)) SEND_STRING data.device,"'admin',$0d"
        ELSE IF(FIND_STRING(data.text,'assword:',1)) SEND_STRING data.device,"'1988',$0d"
}}}
```

******
Be sure to manage "telnet_complete" and "login_complete" full circle.
One Example:

```
{
IP_CLIENT_OPEN(dvDevice.Port,'10.35.93.102',23,IP_TCP)
OFF[telnet_complete]
OFF[login_complete]
}
```