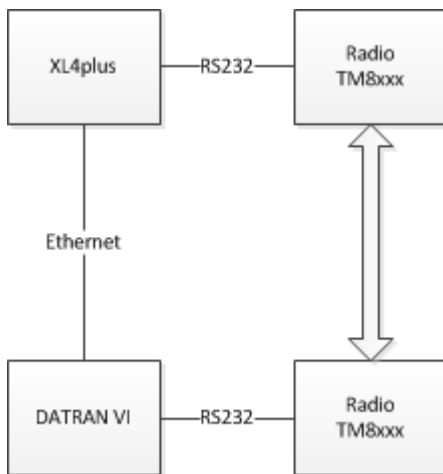


## XL4 *plus* with Dual Failover Comms

### Introduction

An XL4*plus* RTU can communicate with DATRAN via a serial port and via the Ethernet port. It will often be desirable to configure the XL4*plus* such that only the Ethernet port is active yet the serial port takes over if the Ethernet communications fails. Typically the Ethernet port will be connected to a cable modem (ADSL/fibre) or 3G/4G Wi-Fi modem and the serial port to a radio modem.

The physical setup is shown below.



### Ethernet Port Configuration

The XL4*plus* can use DHCP or have a static IP assigned or use NetBIOS. The NetBIOS default name of the RTU is "xl4-serial number", where serial number is the unique RTU serial number. Optionally this can be changed by entering a name in the Configuration → Name field.

In this scenario we are using a fixed IP address of 169.254.226.207

Ethernet Port Settings	
[-] <b>Configuration</b>	
DHCP	<input type="checkbox"/> False
IP Address	169.254.226.207
Netmask	255.255.255.0
Gateway	127.0.0.1
Name	xl4-1674
[-] <b>Current State</b>	
IP Address	169.254.226.207
Netmask	255.255.0.0
Gateway	0.0.0.0
Name	xl4-1674

Next the communication with Datran needs to be configured

- Assign the RTU address as used in Datran (e.g. 13)
- Assign the TCP Port used by Datran (e.g. 4001)

The screenshot shows the configuration interface for Q04 Plus. On the left is a tree view with 'Comms Options' expanded to 'Datran', where 'Datran - Base Station (ethernet)' is selected. On the right, the 'QComms Server' configuration panel is displayed with the following settings:

QComms Server	
<b>Configuration</b>	
Enabled	<input checked="" type="checkbox"/> True
Address	13
TCP Port	4001
<b>Timing Settings</b>	
COS Enabled	<input checked="" type="checkbox"/> True
TCP Keep-Alive Interval	300 s

## Serial Port Configuration

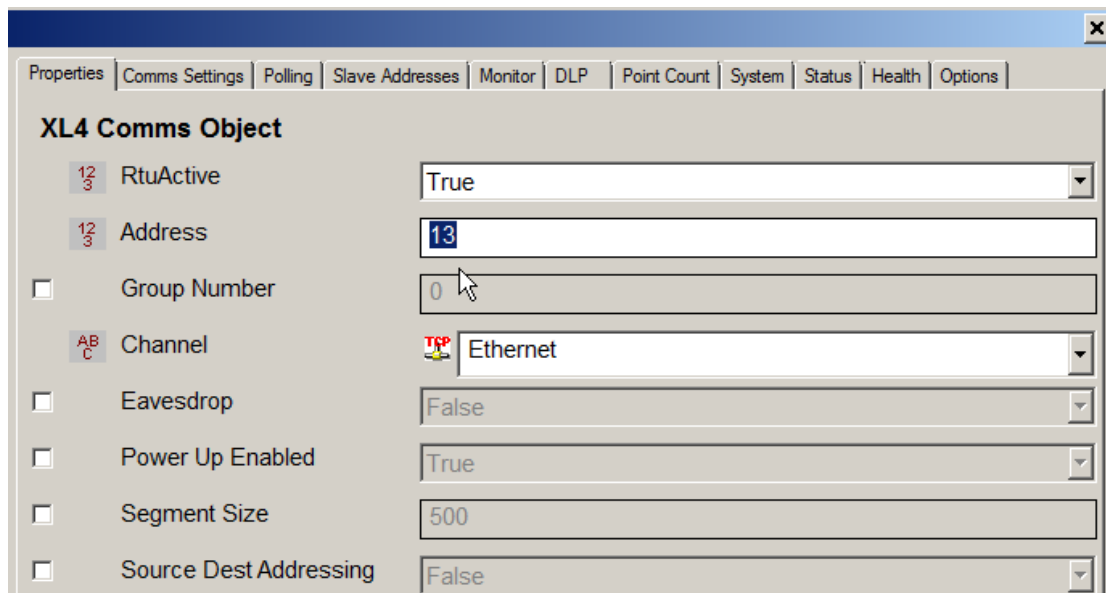
In this example the serial port is configured for use with a Tait radio. The important step is to configure it to use the same address as that used for the Ethernet port. All defaults should be accepted unless there is a specific reason to change them.

The screenshot shows the configuration interface for Q04 Plus. On the left, the tree view shows 'Datran - Base Station (serial)' selected. On the right, the 'Datran - Base Station (serial)' configuration panel is displayed with the following settings:

Datran - Base Station (serial)	
<b>Configuration</b>	
Enabled	<input checked="" type="checkbox"/> True
Address	13
Communication Port	RS232B
Equipment	Tait Radio
<b>Timing Settings</b>	
COS Enabled	<input checked="" type="checkbox"/> True
Lead-In Bytes	4
Lead-Out Bytes	2
Max Attempts	5
Short Retry Delay	3 s
Long Retry Delay	300 s
<b>Serial Port Settings</b>	
Baud Rate	2400
Data Bits	8
Parity	None
Stop Bits	1

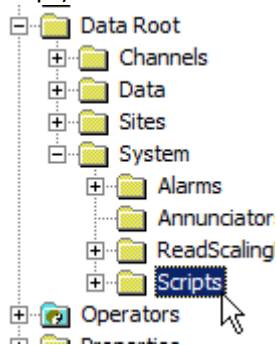
## DATRAN Configuration

Configure the Comms Object to have the site number assigned to the Ethernet & Serial ports of the XL4plus.



## Failover Script

The failover of the comms is controlled by a script. Most of the parameters are configurable. The script is added to the DATRAN TBD file at the location used for general purpose scripts. This varies from system to system and is customer configurable. Typically it will be found at a location as shown below. Adding the script is only done once. Thereafter templates (of the script) are added to each site that requires dual failover comms.

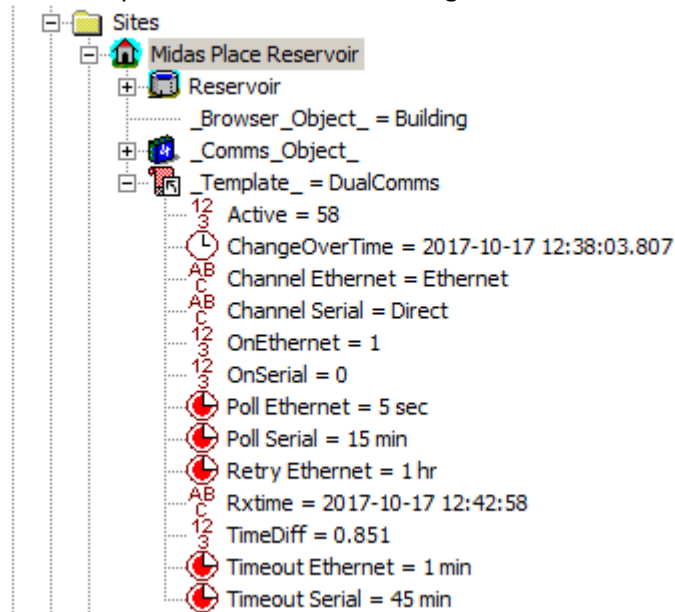


Create a node to hold the script

- At the Scripts node press the "Insert" key
- At "Enter the name of the node..." enter DualComms, press OK
- With DualComms node selected, press insert again, select the Scripting tab, select \_script\_
- Copy and paste the contents of Appendix 1 into the script.

- Select the site that is to have dual failover comms enabled, in this example select Midas Place Reservoir.
- Press the insert key, select the Scripting tab, select `_template_`
- In the Script drop down box either type DualComms or select it from the drop down box list. Click Apply Template.

Your template should look something similar to that shown below



The script will generate all the parameters with some default values. These can be edited to suit your system. You can edit them either in the script to make a generic change or at the individual `_template_` node to make site specific changes. The values you can (and should) change are

- Channel Ethernet: This is the name of the "TCP RTU Channel" comms object
- Channel Serial: This is the name of the Serial channel comms object
- Poll Ethernet: Defines the "Poll Interval" copied to the comms object, used when running on Ethernet
- Poll Serial: Defines the "Poll Interval" copied to the comms object, used when running on Serial
- Retry Ethernet: How often we will try to switch back to running on Ethernet (even when running successfully on serial) 0=never try
- Timeout Ethernet: Time delay before switching to Serial if no data being received
- Timeout Serial: Time delay before switching to Ethernet if no data being received

## Appendix 1. Dual Comms Script

```
//
// Enable Dual Comms on an Ethernet channel and a Serial Channel.
// The script initialises to Ethernet on startup, it then checks for the comms object rxtime changing. If it
// exceeds a parameter it switches to the Serial port and adjusts the polling rate.
// When running on Serial it will periodically retry going back to the Ethernet channel
// (this assumes you want this to be the default channel).
//
// Parameters
// Channel Ethernet: This is the name of the "TCP RTU Channel" comms object
// Channel Serial: This is the name of the Serial channel comms object
// Poll Ethernet: Defines the "Poll Interval" copied to the comms object used when running on Ethernet
// Poll Serial: Defines the "Poll Interval" copied to the comms object used when running on Serial
// Retry Ethernet: How often we will try to switch back to running on Ethernet (if running on Serial) 0=never try
// Timeout Ethernet: Time delay before switching to Serial if no data being received
// Timeout Serial: Time delay before switching to Ethernet if no data being received

// Information
// Active: Current seconds, only updates when script executes - just to let you know we're running
// ChangeOverTime: Time we switched modes
// OnEthernet: Comms set to Ethernet
// OnSerial: Comms set to Serial
// RxTime: Last received data
// TimeDiff: Difference of current time from last received time

// 17 October 2017
// Guy Halliburton, QTech

// Initialise to using the ethernet port, setup sensible values if not existing
OnStart OnDo:
  If Not Exists ([_template_\Poll Ethernet])
  {
    [_template_\Poll Ethernet] := MakeTimeSpan(1 Min);
  }
  If Not Exists ([_template_\Poll Serial])
  {
    [_template_\Poll Serial] := MakeTimeSpan(15 Min);
  }

  If Not Exists ([_template_\Channel Ethernet])
  {
    [_template_\Channel Ethernet] := "Ethernet" ;
  }
  If Not Exists ([_template_\Channel Serial])
  {
    [_template_\Channel Serial] := "Serial" ;
  }

  If Not Exists ([_template_\Retry Ethernet])
  {
    [_template_\Retry Ethernet] := MakeTimeSpan(2 Hr) ;
  }
}
```

```
If Not Exists ([_template_ \Timeout Ethernet])
{
  [_template_ \Timeout Ethernet] := MakeTimeSpan(5 Min) ;
}
If Not Exists ([_template_ \Timeout Serial])
{
  [_template_ \Timeout Serial] := MakeTimeSpan(45 Min) ;
}

[_template_ \Rxtime] := [_Comms_Object_ \RxTime];
[_template_ \OnSerial] := 0;
[_template_ \OnEthernet] := 1;
[_Comms_Object_ \Poll Interval] := [_template_ \Poll Ethernet];
[_Comms_Object_ \Channel] := [_template_ \Channel Ethernet];
[_Comms_Object_] := Null;
[_template_ \Active] := 0;

Poll 30 Sec;
```

#### OnPoll:

```
[_template_ \Active] := Second(Time.Local);
[_template_ \Rxtime] := [_Comms_Object_ \RxTime];
[_template_ \TimeDiff] := (Time.Local - [_template_ \Rxtime])/1000;

If ([_template_ \OnEthernet])
{
  If ((Time.Local - [_template_ \Rxtime]) > [_template_ \Timeout Ethernet])
  {
    [_template_ \ChangeOverTime] := Time.Local;
    [_template_ \OnSerial] := 1;
    [_template_ \OnEthernet] := 0;
    [_Comms_Object_ \Poll Interval] := [_template_ \Poll Serial];
    [_Comms_Object_ \Channel] := [_template_ \Channel Serial];
    [_Comms_Object_] := Null;
    Poll 60 Sec; //give time for serial comms object to reset
  }
  Else
  {
    Poll 1 Sec;
  }
}

ElseIf ([_template_ \OnSerial])
{
  If ((Time.Local - [_template_ \Rxtime]) > [_template_ \Timeout Serial])
  {
    [_template_ \OnSerial] := 0;
    [_template_ \OnEthernet] := 1;
    [_Comms_Object_ \Poll Interval] := [_template_ \Poll Ethernet];
    [_Comms_Object_ \Channel] := [_template_ \Channel Ethernet];
    [_Comms_Object_] := Null;
    Poll 30 Sec; //give time for ethernet comms object to reset
  }
}
```

```
}
Else
{
    Poll 1 Sec;
}
}

// When on Serial, periodically try resetting back to ethernet
If ([_template_\OnSerial] And [_template_\Retry Ethernet] != 0)
{
    If ((Time.Local - [_template_\ChangeOverTime]) > [_template_\Retry Ethernet])
    {
        [_template_\OnSerial] := 0;
        [_template_\OnEthernet] := 1;
        [_Comms_Object_\Poll Interval] := [_template_\Poll Ethernet];
        [_Comms_Object_\Channel] := [_template_\Channel Ethernet];
        [_Comms_Object_] := Null;
        Poll 30 Sec; //give time for comms object to reset
    }
}
```