

The IP Network Driver Router

The IP Network Driver Router program acts as a conduit between an internal IP network (multiple QL emulators running on one PC) and an external IP network (single QL emulators running on multiple PC's) using the IP Network Device Driver. It allows network stations on one network to access network stations on the other network.

The IP Network Driver Router sits between the two QL networks on the PC that contains the internal (IPLocalNet) network. To each network, the IP Network Driver appears to be just another network station with it's own station number.

For the IP Network Driver Router to operate, the PC that is running the internal (IPLocalNet) network must also be on the same Local Area Network as the PC's that are running the external (IPNet) network.

While normal operation of the IPNet and IPLocalNet networks are unaffected. To access a network station on the other (secondary) network from your (primary) network, Entails you having to through the IP Network Driver Router station, using an extended network specifier. This extended specifier comprises of the IPNetwork Driver Router station number on your side of the network, then the network station number required on the other network.

Do not try to run the IP Network Driver Router and IP Network Device Driver on the same emulator at the same time. However if you try it, you may get limited functionality in certain directions.

The IP Network Device Router program requires IPNet & IPLocalNet versions 1.08 and above to operate. (IPNet & IPLocalNet were combined into one program, the IP Network Device Driver at version 1.10)

The IP Network Driver Router was developed for use with QPC2. And is also compatible with SMSQemulator version 2.21 or above.

However it is not suitable for use on Qemulator, due to one of the IP device driver commands (IP_ACCEPT) working differently from QPC2 & SMSQemulator.

IP Network Driver Router should not cause any problems, damage, or loss of data. However you use it at your own risk, and I do not accept any responsibility for any damage, or loss of data.

Installing the Program

IPNet Router should be installed on it's own copy of QPC2 or SMSQemulator on the same PC that is running the QL emulators that makes up the internal (IPLocalNet) network.

To load the program, load into memory and call it.

example: i. **LRESPR flp1_Router_cde**

ii. **x=RESPR(2422)** {if you don't have Toolkit 2, or equivalent}
LBYTES flp1_Router_cde,x
CALL x

An installation message, and a version number will be displayed in #0

Note: IP Network Driver Router should be installed on it's own QL emulator and not alongside the IP Network Device Driver. You may encounter problems if you try to run the two programs on the same QL emulator, at the same time.

Starting the Router

The router is started with the **ROUTER** command. This starts a background job that constantly scans for connection requests from the IP Network Device Driver networks. When a connection request is received, another background job is created to handle the connection, while the Router continues to scan for more connection requests.

ROUTER FROUTER

The **ROUTER** command, or the **FROUTER** function starts the Router.

The Router will appear to the IP Network Device Driver networks as just another network station on each network.

ROUTER requires two parameters. The first parameter is the network station number that the Router will be on the internal (IPLocalNet) network. The second parameter is an IP Address string of the network and station number of the Router on the external (IPNet) network. This takes the same form as the **NET_START** command found in the IP Network Device Driver. IP Addresses ending in .0 e.g. 172.16.0.0 will not be accepted. As this would make the Network station number 0, which is not allowed in the IP Network Device Driver.

FROUTER is a function version of **ROUTER** which will return 0, or an error code.

ROUTER must be started with parameters the first time it is used. But if the Router job is killed. It may be restarted without parameters, in which case it will try to use the last parameters that were supplied to it.

syntax: `station_number := Integer` {1 to 8}
 `IPAddress:= String`

`ROUTER [station_number, IPAddress]`
 `FROUTER ([station_number, IPAddress])`

example: i. **ROUTER 2,"172.16.0.6"**
 ii. **result=FROUTER (2,"172.16.0.6")**

comment: In the above example the Router will be network station number 2, on the internal (IPLocalNet) network. And network station number 6 on the "172.16.0.xx" external (IPNet) network.

note: Internal (IPLocalNet) station numbers could be higher than 8, but you would then have to use the **MAP_N** command in the IP Network Device Driver.

Using the IP Network Driver Router

The use of all of the normal IP Network Driver commands remain unchanged. To access a network station on the other 'secondary' network is done via the router station. By using an extended network specifier of the routers station number, and then the station number required on the secondary network.

For the most part this double network station number definition works, but the operating system does get in the way occasionally, see below.

Accessing File Servers on the other network

To enable you to access a file server on a network station that is on the other network, an extended compound device name is used:

Nrouter station number_Nsecondary station number_IO device

For example, If the router's station number on your network is 2 then

LOAD n2_n4_flp1_fred

Will load the file 'fred' from floppy 1 on network station 4 on the secondary network.

OPEN_IN #3,n2_n1_flp2_myfile

Opens 'myfile' on floppy 2 on network station 1 on the secondary network.

However if you try

DIR n2_n4_win1_

You will not get the expected directory of WIN1_ on the secondary networks station number 4. You will get the drive statistics, but no list of files. This is not a problem with IP Network Driver Router, or the IP Network Device Driver. But SMSQ/E getting confused, and instead of giving a list of all files in WIN1_, it tries to give a list of all files in the win1_ subdirectory of win1_ i.e. win1_win1_

This is to do with the 'wildcard' handling, and other wildcard commands like **WCOPY** are liable to give problems.

There are a couple of ways around this problem. One is to use the **NFS_USE** command.

NFS_USE mdv,n2_n4_win1_

Then **DIR mdv1_** will give you the required directory of n2_n4_win1_

Alternatively use the **DEV_USE** command.

DEV_USE 4,n2_n4_win1_

Then **DIR DEV4_** will give you the required directory of n2_n4_win1_

RENAME

When renaming a file on a server through the router you should not specify the router station number in the **TO** side of the command. e.g.

RENAME n1_n4_win1_file1 TO n4_win1_file2

NFS_USE mdv,n1_n4_win1_
RENAME mdv1_file1 TO mdv1_file2 {the n1_ will be removed for you}

Accessing network stations on the other network

To send to, or receive from network stations on the secondary network, The **NET** device has been extended:

NETO_ *router station number_secondary station number*

NETI_ *router station number_secondary station number_timeout*

Where the secondary station number is the final destination station in a NETO, and the originator station in a NETI.

For example, If you are network station number 1 on your network, and the router's station number is 2 then

OPEN#4,NETO_2_4

Will open a channel to network station 4, on the secondary network via the router.

And if the router is seen as network station 6 on the secondary network, then on the network station 4 on the secondary network, the following would be entered

OPEN#4,NETI_6_1

This will open a channel to network station 1, on the secondary network via the router.

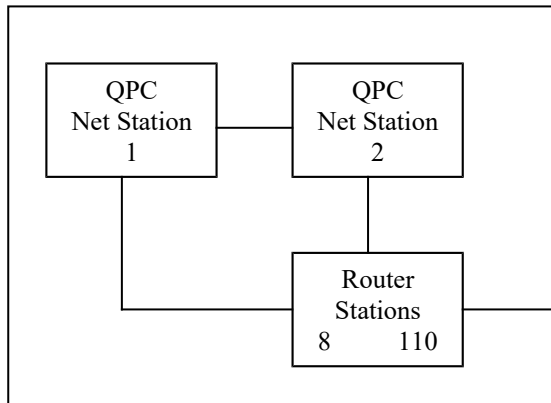
Note: The originating station number (1 in the above case) is not checked during the **OPEN**, but as soon as a data packet is received, the packet is checked to see that it comes from the desired originator. If they do not match then a 'Transmission error' is generated.

Acknowledgements

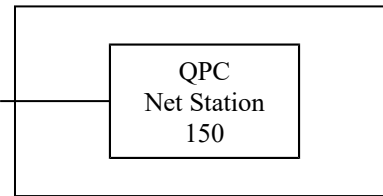
I would like to thank Giorgio Garabello, Urs Koenig and Wolfgang Lenerz for their suggestions, advice, and help in the development and testing of the IP Network Device Driver, and the IP Network Driver Router.

Example Set up

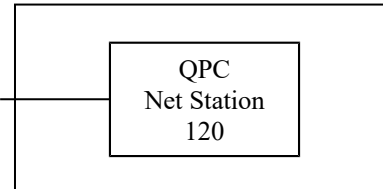
PC 'A' IP Address 192.168.0.110



PC 'B' IP Address 192.168.0.150



PC 'C' IP Address 192.168.0.120



You have three PC's connected together on a Local Area Network. PC's 'B' and 'C' are running IPNet. PC 'A' has three copies of QPC2 running. Two copies are running IPLocalNet with station numbers of 1 and 2. The third copy of QPC2 is running IPNet Router as station 8.

On PC 'A' QPC net station 1

NET 1 : LRESPR dos2_LocalNet_cde

FSERVE

DEV_USE 2,n8_n120_win1_

DEV_USE 3,n8_n120_ram1_

DEV_USE 4,n8_n150_win1_

DEV_USE 5,n8_n150_ram1_

DEV 2 & 3 access PC 'C' and DEV 4 & 5 access PC 'B'

On PC 'A' QPC net station 2

NET 2 : LRESPR dos2_LocalNet_cde

FSERVE

NFS_USE mdv,n8_n120_win1_, n8_n120_ram1_, n8_n150_win1_, n8_n150_ram1_

MDV 1 & 2 access PC 'C' and MDV 3 & 4 access PC 'B'

On PC 'A' QPC router station

LRESPR dos2_Router_cde

ROUTER 8,"192.168.0.110"

On PC 'B' QPC net station 150

LRESPR dos2_NetDriver_cde

IP_ADDR "192.168.0.150"

FSERVE

MAP_N 1,110

MAP_N 2,120

NFS_USE mdv,n1_n1_win1_, n1_n1_ram1_, n1_n2_win1_, n1_n2_ram1_

MDV 1 & 2 access PC 'A' QPC net station 1 and

MDV 3 & 4 access PC 'A' QPC net station 2

On PC 'C' QPC net station 120

LRESPR dos2_NetDriver_cde

IP_ADDR "192.168.0.120"

FSERVE

MAP_N 1,110

MAP_N 2,150

NFS_USE nwk,n1_n1_win1_, n1_n1_ram1_, n1_n2_win1_, n1_n2_ram1_

NWK 1 & 2 access PC 'A' QPC net station 1 and

NWK 3 & 4 access PC 'A' QPC net station 2