
TCP/IP Protocol Suite (Release 2)

Release Note

This note clarifies some operational issues not covered in the product documentation, and describes any deviations from the product documentation.

Starting Ethernet if not connected to a network

If you start the Ethernet interface when not physically connected to a network you may notice that your machine slows down for a few seconds every 30 seconds or so. This is caused by the Ethernet device driver attempting to start the Ethernet controller, which is detecting the line error and failing to respond, thereby causing the driver to time out and retry the initialisation procedure.

Configuring an R140 Econet IP interface

When configuring an R140 RISC iX host to enable communication with a RISC OS client via TCP/IP over Econet, you should pass the parameters `-trailers` and `arp` to the R140 `ifconfig` program when starting the Econet interface. This program is normally invoked from the system startup file `/etc/rc.net`. For example:

```
ifconfig ec0 inet arp -trailers eco_ifname
```

You don't need to pass these parameters to the equivalent `ifconfig` program running under RISC OS.

Use of *RMTidy

You should use *RMTidy with great care if the TCP/IP system is active.

For example, the Internet module will be left in a state equivalent to that existing after it has been *RMLoaded. Network interfaces will not be initialised and routes will not be established. To recover properly you should run !Internet . !Run, to restart the interfaces and configure routing tables according to local requirements.

Rebooting an NFS server while Mounts are active

Rebooting an NFS server host while a RISC OS client has mounts active on that host may cause subsequent communication attempts from RISC OS to timeout and fail. This is because the RISC OS client has cached certain port numbers required for communication, and these port numbers may have changed on the host across the reboot.

Currently, the only way to recover from this situation, in order that the host can be accessed once more, is either to *RMReInit the NFS module or to reset the RISC OS machine.

Sorting large directories for the NFS Filer

Mounted NFS directories can contain more than 255 files, but the Filer that displays directories in the RISC OS desktop will only sort directories of up to 255 files. If a directory display contains more than 255 files, they will appear in the order they are stored in the mounted directory.

Top bit set characters in mount names

Although your mount names for the NFS Filer can contain *top bit set* characters, you should not save such a mount. If you do so, next time you start the NFS Filer you'll get the error 'Invalid mounts file'.

and any mounts saved after the offending mount name will be ignored. You can fix this problem by deleting the offending line from the `mounts` file.

Insecure directory access via NFS

It has been observed that some NFS servers allow a client to list a directory which has execute access but no read access (an operation which is not allowed using the UNIX `ls` command when logged in). From RISC OS this results in a directory view of the files within the directory being displayed on the desktop.

This is a fault at the NFS server.

The NFS Cache

RISC OS NFS has an internal cache of UNIX file attributes and lookups of a file in a directory. Each entry in these caches is assigned a time for which it is assumed to be valid – after this time RISC OS NFS will access the information from the server. When a cache entry is filled in, the time when it becomes invalid is stored, which is calculated as *CacheTime* centiseconds from now; the cache entry is assumed valid until that time is reached. As a consequence of this algorithm, changing the *CacheTime* won't affect entries already cached; it will only affect newly-cached information.

The default *CacheTime* is 6000 centiseconds (1 minute). The UNIX file attribute cache has 23 entries. The file lookup cache has 23 entries.

When will the cache become out of date? Updates caused on the local machine will update the cache immediately, but updates caused on a remote machine may not be seen on the local machine until *CacheTime* later.

NFS File Types

If you are starting NFS from the command line rather than the desktop, you should set the following file types:

```
*Set File$Type_FE6 "UNIX Ex"  
*Set File$Type_FDC "SoftLink"
```

Effect of VDU 21 in *Telnet output stream

If *Telnet is run from the CLI, and ^U appears in the output stream from the remote host, then this will be interpreted by the VDU handler as a VDU 21 command and screen display will be disabled. You should type ^F, which when echoed will be interpreted as a VDU 6 command, to re-enable screen output.

Note that in some host terminal modes typing ^U (to delete a line) may result in the ^U character being reflected back to the terminal. You should choose a host terminal mode (eg stty dec on RISC iX) which does not cause this to occur.

Use of *Telnet in a task window

If *Telnet is run within a task window, you should always remember to close any open connection and quit *Telnet before closing the task window. Failure to do so will cause your machine to crash.

VT220 Line Width

When a connection is opened over the Telnet protocol the ID given by !VT220 to the remote host is VT220, so a UNIX host will use the 'termcap' entry for VT220. This entry will define the line width for a VT220 terminal. Changing the line width (say to 132 chars) on the local terminal will not effect the line width as perceived by the remote host. This can only be achieved by changing the termcap entry on the remote host.

Telnet protocol uses 7 bit characters

The Telnet protocol uses 7 bit characters. Therefore if VT220 is used over the Telnet protocol, it should not be set to the VT200 8bit mode.

Tab stops on RISC iX

The default tty setting for RISC iX machines is `stty -tabs`

This makes the host convert tabs to a sequence of space characters, and prevents the terminal's tab settings from working. Type

```
stty tabs
```

when logged into a RISC iX host to make the host send tab characters instead of spaces.

VT220 Wrap mode

Some programs (such as `vi`) expect the user terminal to be in no-wrap mode, but do not set the mode when they start. This can result in apparently erroneous screen updates in !VT220.

If a program expects no-wrap mode, set it before starting the program.

VT100 vs VT200

Many host applications will expect to be talking to a VT100, rather than a VT200 type terminal. To make sure that the emulator application presents itself as a VT100, you should both set the terminal mode to VT100 **and** set the terminal id to VT100 ID.

VT200 Function Keys

If the remote application you are using uses the VT200 function keys, make sure that !VT220 is **not** in Local Fn keys mode **before** you start the application. Also note that the function keys only send codes when the terminal is in VT200 mode, and not when it is in VT100 mode.

VT220 Errors File

The file !VT220.Errors is not mentioned in the *User Guide*. This file contains explanations of the various error messages which may be generated by the VT220 application.

Use of Pause and Abort in Ftp

When performing a multiple file transfer, or a directory transfer, clicking on either Abort or Pause/Continue will have effect only if actual data transfer is in progress. At other phases in the transfer operation (such as setting up data connections) these clicks will be ignored.



Ethernet is a trademark of Xerox Corporation
UNIX is a trademark of AT&T
VT is a trademark of Digital Equipment Corporation

Copyright © Acorn Computers Limited 1993
Published by Acorn Computers Limited
September 1993
Part number 0484,190
Issue 1