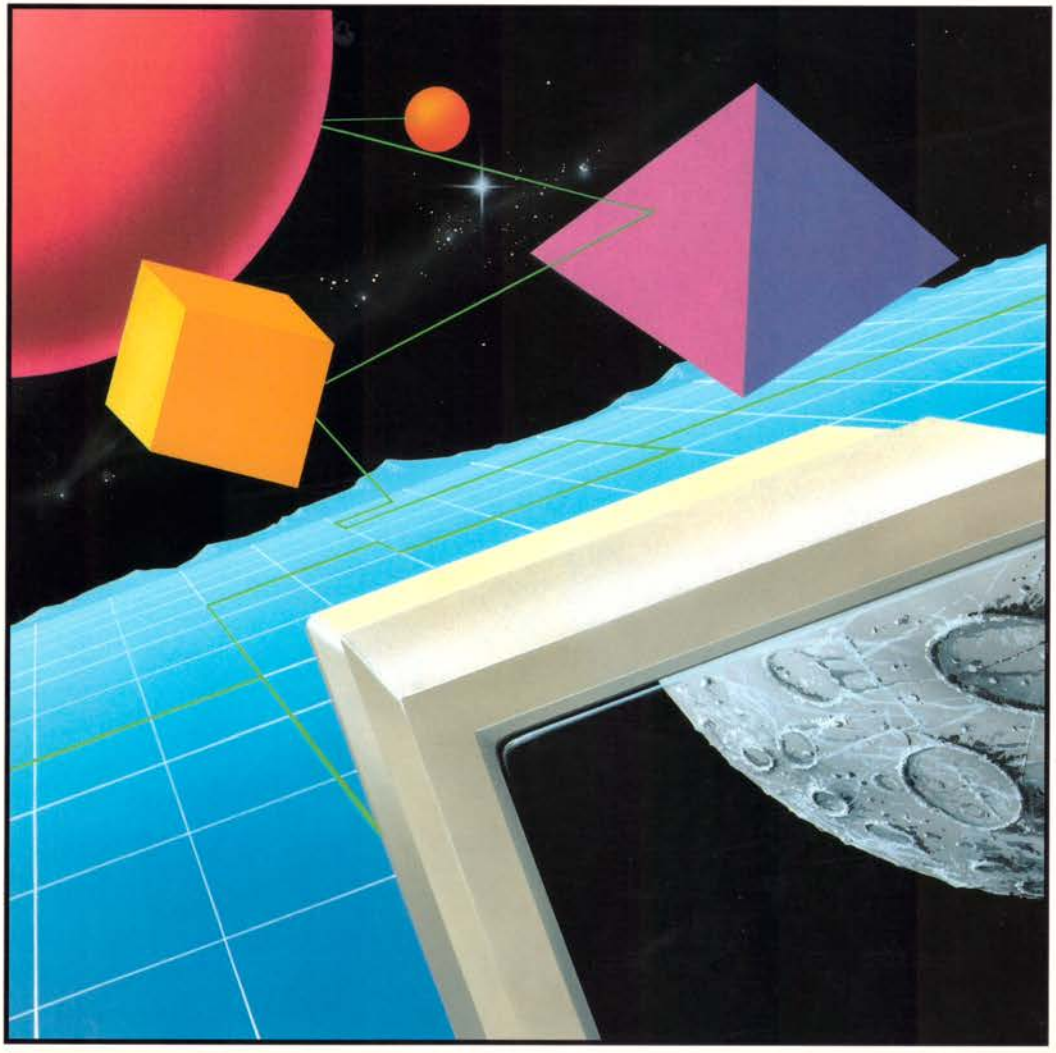
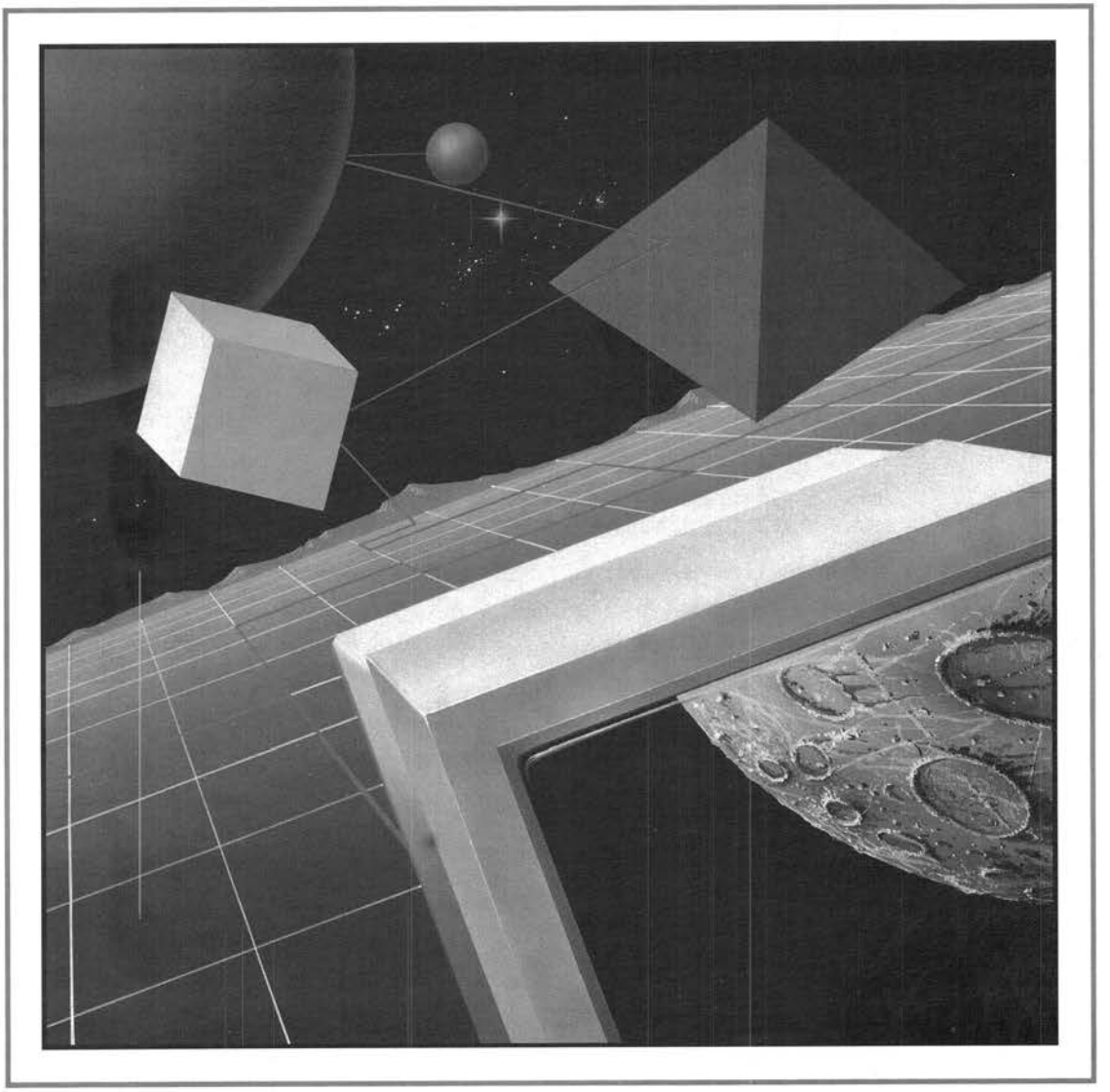
Acorn OmniClient II
User Guide



Acorn

Acorn OmniClient II
User Guide



Acorn

Published by Acorn Computers Limited.

The product described in this manual is not intended for use as a critical component in life support devices or any system in which failure could be expected to result in personal injury.

The product described in this manual is subject to continuous development and improvement. All information of a technical nature and particulars of the product and its use (including the information and particulars in this manual) are given by Acorn Computers Limited in good faith. However, Acorn Computers Limited cannot accept any liability for any loss or damage arising from the use of any information or particulars in this manual.

If you have any comments on this manual, please complete the form at the back of the manual and send it to the address given there.

Acorn supplies its products through an international distribution network. Your supplier is available to help resolve any queries you might have.

Within this publication, the term 'BBC' is used as an abbreviation for 'British Broadcasting Corporation'.

Published by Acorn Computers Limited
ISBN1 85250 169 3
Part number 0484,952
Issue 3, February 1997

Contents

1 Introduction

Acorn OmniClient is a universal desktop filer for network-based filing systems running on Acorn RISC OS computers. It allows Acorn platform users to store and retrieve files in conjunction with 'alternative' file servers, and makes the most of hardware and software located on local area networks (LANs) that use Acorn machines.

It embraces the following network filing systems:

- Acorn AUN Level 4
- Acorn Access and Acorn Access+
- NFS (TCP/IP Protocol Suite)
- Lan Manager (NT workstation 3.1 and 3.5, NT Server 3.5, NT Advanced Server 3.1 and Windows for Workgroups).

OmniClient provides a simple, consistent view of network file services to RISC OS users, irrespective of the server type or protocol used.

Packing list

The Acorn OmniClient pack consists of the following components:

- Envelope containing a software disc; the disc contains the following software:
  - !Omni
  - !BootNet
  - !InetSetup
  - !Internet
  - !OmniSetup
  - !LMPasswd
  - !System and !Sysmerge
  - Example boot files.
- Release note
- Owner registration form
- This guide.

If any of these is missing or damaged, please contact your supplier without delay.

User licences

Where user licences are required to use file server software it is still a requirement of the vendors of this software that you purchase the relevant user licences.

Examples of software configurations requiring this are:

- Lan Manager
- NT Server
- Windows for Workgroups.

Further Reading

- TCP/IP *Protocol Suite* (*Release* 2)
- AUN *Manager's Guide* (Release 3)
- *Ethernet Design and Installation Guide*
- *Network Acorn* by Geoff Preston; Northwest SEMERC, 1994.

DCI4

The software modules that make up the protocol stack, allowing network communication to take place between machines, have been updated from DCI2 to DCI4 (DCI stands for *Device Control Interface*) The new version of DCI allows much greater flexibility in mixing and matching device drivers and different protocols. OmniClient requires DCI4 to operate, and a complete set of DCI4 software is supplied as part of this product (within !System).

Ethernet cards will be supplied with DCI4-compatible modules on them, but you may find that old cards may contain DCI2 versions, which you'll need to update. You may either get updated ROMs from your supplier, or use the software provided.

The use of !BootNet is described in detail in the AUN *Manager's Guide* that is part of the Level4 Fileserver product. Specific hints and tips that cover the various combination of card and software are included in the Release Note for this product.

2 Using Acorn OmniClient

This chapter describes how to load Acorn OmniClient, display and mount file servers, and configure OmniClient for your own use.

OmniClient can either be soft-loaded from a hard disc, floppy disc or network mount, or loaded from your Ethernet card.

Loading Acorn OmniClient from a disc or network

1 Copy !Omni, !Internet! and !InetSetup to your hard disc, or, if you don't have one, to the network hard disc where your boot files are kept.

Note that for discless machines, you will only be able to load this software from servers for which there are protocol modules resident in the machine. All machines support Level 4, those fitted with Acorn Access may additionally load from shared Access discs. The Appendix *OmniClient and discless workstations* on page 87 gives you information on how to do this.

2 Run the application !OmniSetup to configure the protocols you wish to use. Alternatively, ensure that the OmniClient Startup file (by default located in `!Omni.Files`) is configured so that only the network filing systems available on your machine are enabled (comments in this file will help you do this).

3 Return your computer to its power-on configuration, by resetting it. Find the !Omni application and double-click on it to load it onto your icon bar. Save a Desktop boot file: if you have a hard disc, save it to your root directory, or to !Boot with RISC OS 3 versions 3.5 and later; if not, to the network hard disc where your Desktop boot files are kept.

OmniClient will now be loaded on your icon bar when you switch on your computer.

Loading OmniClient from ROM

You may wish to upgrade the ROM on your Ethernet card to one that includes the OmniClient software, especially if you have a site that only has Lan Manager style servers. Please refer to your network card supplier for details on how to make this upgrade.

When LanManFS is present in ROM, it is possible to boot resources and the 'front end' of the OmniClient application from this server.

To do this, you need to direct the client station to the required 'boot' server. You will need to configure the client by using the following commands:

```
*Configure fs myfileserver
*Configure filesystem LanMan
*Configure boot
```

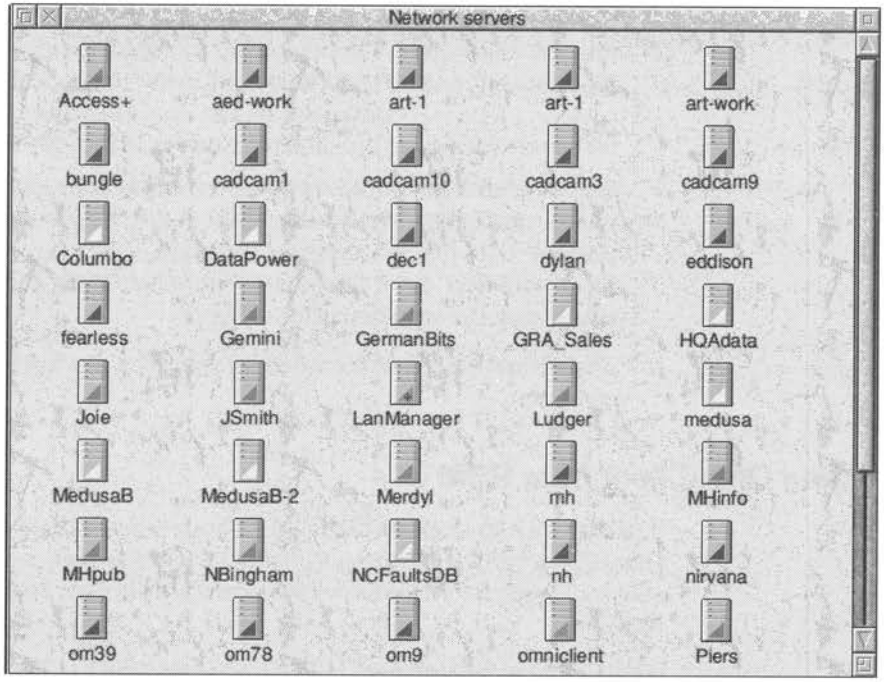
This will tell the client to authenticate as user Armboot in domain 'myfileserver', then mount the Armboot share and run the !Armboot file that it may find there.

Using Acorn OmniClient

Displaying available servers

Click on the OmniClient icon displayed on the lefthand side of your icon bar (or choose **FS list** from the OmniClient icon bar icon menu).

A window will be displayed, listing all the file servers and shared discs which you can access:

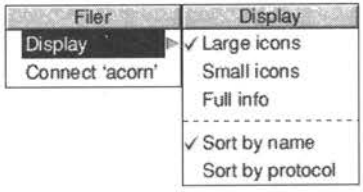


The network filing system to which each server belongs is indicated by a triangle on the server icon, with the following colour-coding:

Filing system	Colour code
Acorn NFS	dark blue
Lan Manager/NT Server (and all other Microsoft options)	light blue
Lan Manager (Domain Logon)	light blue with + sign
Level 4	yellow
Acorn Access	green
Acorn Access+	green (with a plus sign)

Displaying the list of file servers by name or by protocol

The Filer menu on the file server window allows you the normal display choices of **Large icons** and **Small icons**, as well as a **Full Info** display which contains more detailed information about each type of server. It also allows you to sort the servers by server name or protocol.



Mounting different server types

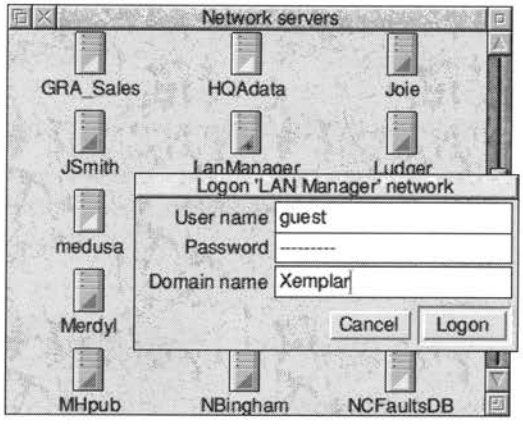
Some network fileservers combine the act of authentication and mounting a particular directory; for example Level4 and NFS. Others, such as Access+ and Lan Manager servers keep these two operations separate – users are invited to authenticate themselves to the network system before mounting any directories to which they are allowed access.

If you are using Access+ or LanMan, you will see 'dummy' fileservers called Access+ and LanManager in the list of fileservers window. You may authenticate yourself by clicking on these icons and entering the log-on details in the dialogue box displayed. Once authenticated, network mounts can be selected simply by selecting them from the list of fileservers window.

If a 'home' directory is defined for you on an NT/Lan Manager server, this will be mounted with the name 'Home'.

Primary domain controllers

If you are using NT Servers, such as the Acorn School Server or Class Server products, you should have set up one server (if you have more than one) as a primary domain controller. This means that you need only authenticate once to have all the resources allocated to you in that domain available for your use. This is a natural consequence of the security model implemented in a Microsoft domain environment.



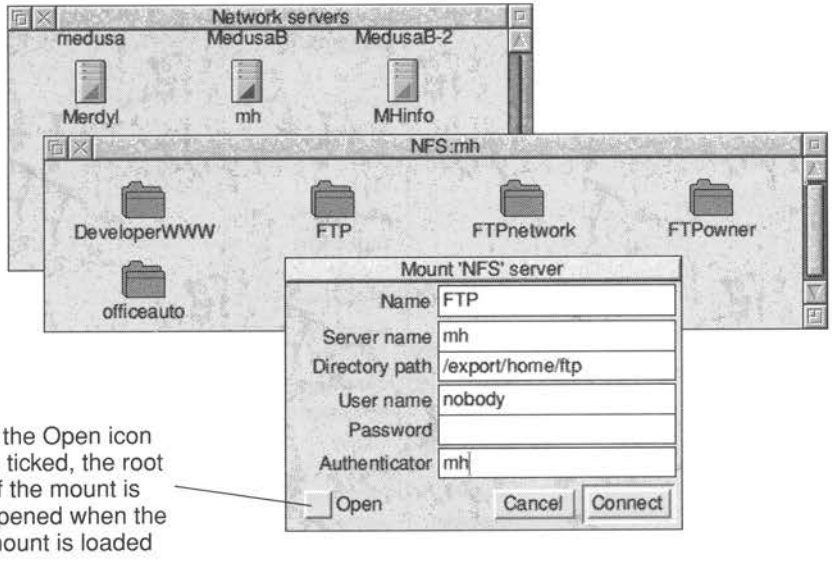
Mounting a file server

There are several different ways to mount a file server:

- by clicking on a file server icon in the list of file servers
- by choosing a pre-saved file server from the icon bar icon menu
- by filling in a Mount dialogue box from the icon bar Mounts menu (this is for file servers which do not appear on the list of file servers)

Mounting a file server from the list of file servers

To connect to a particular server, double-click on its icon; you will be connected directly with the server if all relevant information is available, otherwise a Mount dialogue box will be displayed:

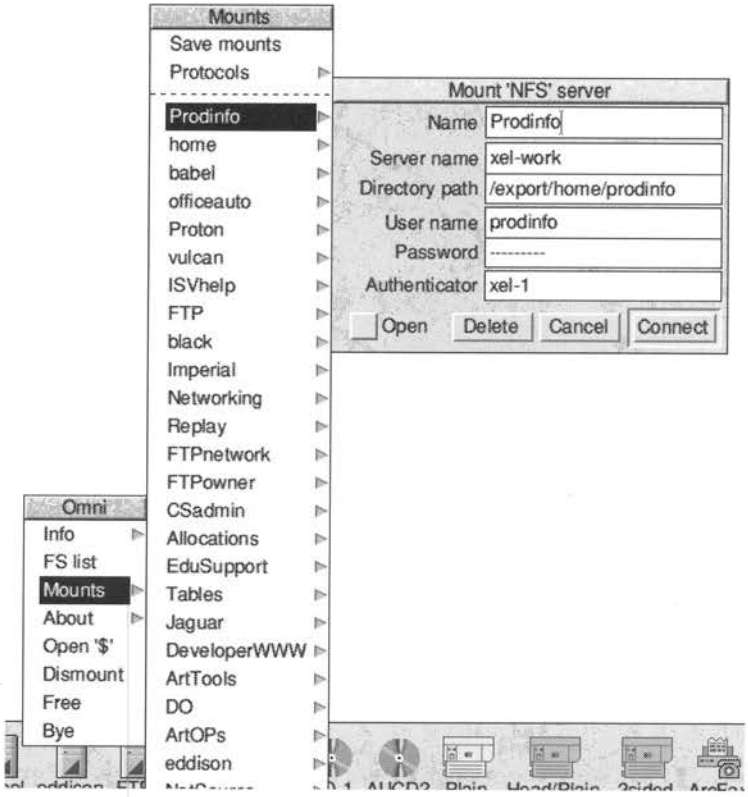


Where mounts have previously been saved (see *Mounts file* on page 14) their details will already be entered in the Mount box, so all you will have to enter is the password.

Mounting a pre-saved file server from the icon bar icon menu

If you have already mounted a file server, or saved the details of a mount in the Mounts file (see *Mounts file* on page 14) then you can go to the **Mounts** submenu to display the (partially) completed dialogue box for that mount. This enables easy access to regularly accessed fileservers.

For example:

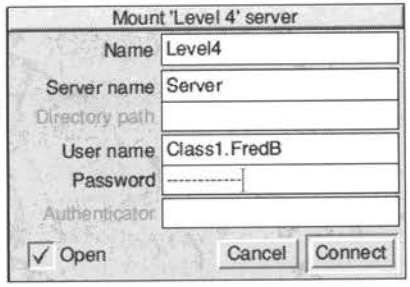


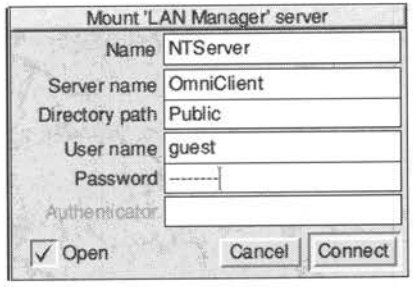
If you choose the **Delete** option, this mount is deleted from the mounts file if you then choose **Save mounts** from the **Mounts** sub-menu.

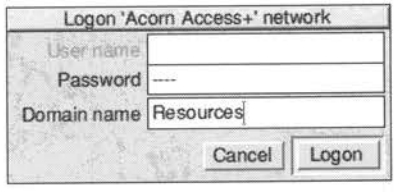
Mounting a file server which is not displayed on the list of file servers

If you need to mount a file server that is not displayed in the list of file servers, you can go to the **Protocols** submenu and click on one of the available protocols displayed. A dialogue box will then appear (with a default user) enabling you to enter all the necessary details before attempting to mount the file server.

For example:







Mounting a file server automatically

You can mount file servers automatically from the OmniClient command tail in your Desktop boot file. For example:

```
Run ADFS::4.$.!Omni mount1 mount2 mount3
```

See *Running OmniClient from your Desktop boot file* on page 12.

When the OmniClient application is started each mount name is looked for in the mounts file (see *Mounts file* on page 14). If the mount is with a filing system (or server) that needs no password (e.g. Acorn Access), or if the password is in the mounts file, the mount will be accessed immediately and an icon will appear on the icon bar.

If the mount requires a password a Mount dialogue box is displayed for you to fill in.

Using mount aliases

A mount alias is the name that appears under a mounted server icon on your icon bar. It is defined in the **Name** field of the **Mount server** dialogue box.

You can either use the default alias, if applicable, or choose your own alias for a given mount (to aid you, OmniClient suggests a default alias name where it can, by choosing the 'short form' of the server name, as shown in the list of file servers). If a mount (either connected or in the saved mounts file) already exists with this name, the mount alias field is left empty for you to fill in yourself.

Each mount must have a unique alias, for all protocols. If you don't choose a unique mount name when you log on to an OmniClient supported server, an error will be generated asking you to choose another name.

After you have mounted a file server

When you have successfully logged on to a file server, an OmniClient icon with the name (i.e. the mount alias) of the new mount will appear on the icon bar, and the directory display for the root of the mount will be opened on the desktop, if you have requested it by choosing **Open**.

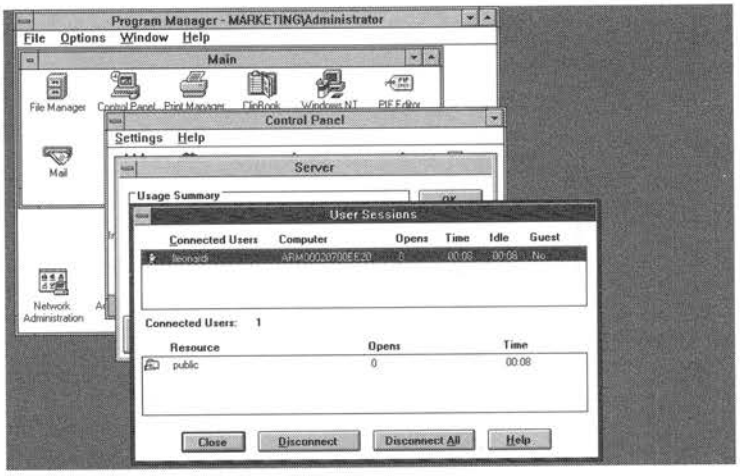
Displaying available servers

After you have established a mount, and an OmniClient icon with the name of the new mount has appeared on the icon bar, clicking Select on this icon will not display the list of file servers, but instead will display the directory mounted by that icon.

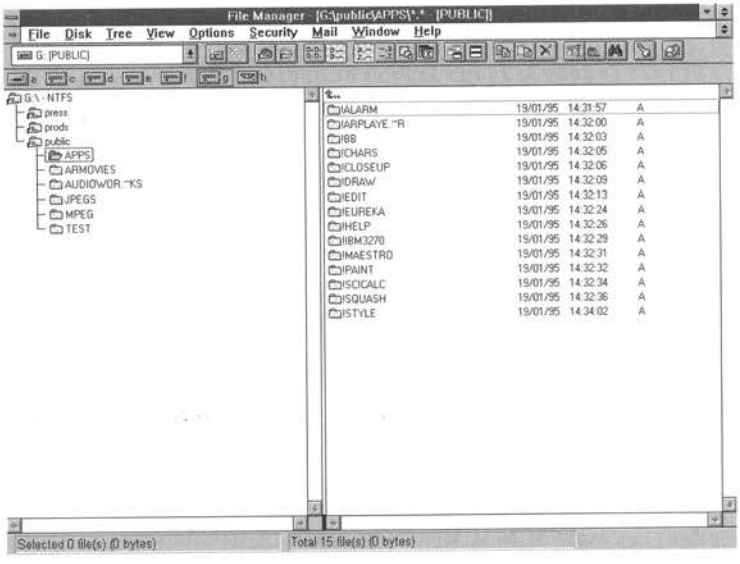
To display the list of available servers you must either choose **FS list** from the OmniClient icon bar icon menu, or click Adjust (not Select) on the OmniClient icon.

Windows™ NT Server

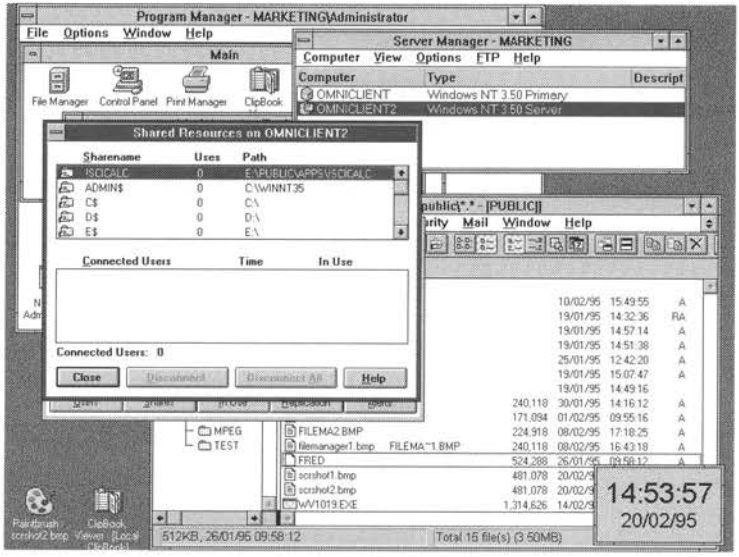
OmniClient allows Acorn platforms to achieve high performance connectivity in conjunction with the Windows™ NT Server product. Acorn clients will appear to the administrator in a very similar fashion to Windows or MS DOS users. The example below shows an active connection from an Acorn Risc PC to the a server:



The following example of File Manager running on a Windows™ NT Server, shows how RISC OS files can be stored and retrieved as users on the Local Area Network require them.



The Windows™ NT Server environment also allows the OmniClient application to choose file servers when storing and retrieving remote files. This means that where more than one file server exists the system can be made totally flexible.



Running OmniClient from your Desktop boot file

If you set up OmniClient to run from your Desktop boot file, you can add one or more mount aliases to the command tail (the command tail is any additional text you add after the command to run !OmniClient) these will then be mounted automatically (although you may have to supply passwords for the mounts if these are not specified in the mounts file).

Instead of adding a list of mount aliases to the OmniClient command tail, you can just add `-ALL`, in which case OmniClient will behave as if the command tail listed all the mounts in the mounts file.

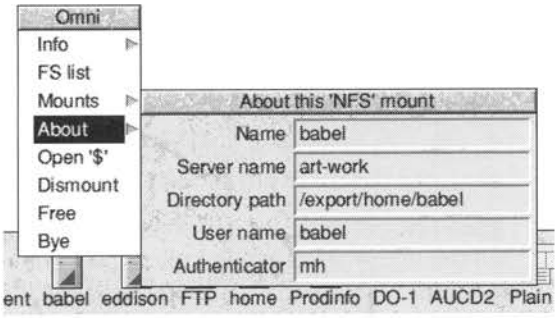
By specifying the fileserver type in the command tail (for example `Share+`, `LanMan+` etc.) all the fileservers of this type will be automatically mounted. A dialogue box will be displayed if more information (for example a password) is required.

To support multiple saved mounts files (e.g. master network file for booting, and a writable per-user mounts file), a *command `OmniLoadMounts` lets a named mounts file be loaded (merged with those already in memory). Mounts will always be saved to the mounts file Omni$Path `Mounts` unless a mount has the LOCKED flag set (see *Flags in the mount file* on page 15).

Other options on the icon bar icon menu

Getting information about a mount

About displays the mounts name, server name, directory path, user name and authenticator:



The 'authenticator' is required by NFS servers who may have a separate name server machine.

Opening a root directory

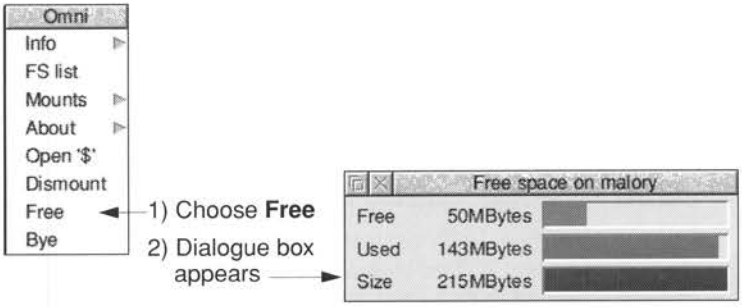
Open '$' opens the user root directory of the file server you are on.

Dismounting a mount

Dismount dismounts the file server and closes any associated directory displays.

Finding the free space on a mount

Free opens a dialogue box that shows you the amount of free space on a mount:



Bye dismounts all active connections and log off all network domains. A confirm box is displayed before this takes place.

Quitting OmniClient

To quit OmniClient, click on the Task manager icon on the icon bar to display the Tasks window, click Menu over the OmniClient entry and choose **Task/Quit**.

Mounts file

Although it is not necessary to access this file under normal circumstances, the mounts file contains the details of any mounts you have specified. The file is in `!Omni.Files.Mounts`.

To save one or more mounts choose **Save mounts** from the **Mounts** submenu. This saves the details of any mounts you have specified to the mounts file.

Each line of the mounts file contains information for a mount. Usually this will be all the information needed to connect a mount with the exception of the user's password, although this can be supplied if the user feels it is not a security problem.

The fields on each line are separated by a comma and white space of any sort other than a new line. The trailing fields on a line may be left off if the null string is to be used. Any line beginning with a '#' character is ignored (and will be lost if the file is rewritten).

Below is an example mounts file:

```
# > <Omni$Path>Mounts
# Mount list saved by OmniClient at 16:14:54 on 8 Nov 1994
#
# Protocol, Alias, Flags, Server, Mount path, User ID,
# Password, Authentication
#
LanMan,IanC,L,PC1,C:,ian
LanMan,PC2,L,PC2,,ian
NFS,Apps,P,zebedee,/extra/apps,apps,letmein,zebedee
```

In the previous example the mount called Apps could be mounted without typing a password (it has the one it needs). Also, note that Access does not require a password or authenticator to gain access to a hard disc or CD ROM.

A null password can be represented in the mounts file as the string `""` (i.e. two double-quotes without any spaces). This could be used where a guest account has no password, and the user doesn't wish to be prompted for a password to be entered. However, in the case of ShareFS, they are automatically assumed.

The mounts file is automatically updated when you choose **Save mounts** from the **Mounts** submenu. This also remembers the state of the **Open** button.

Flags in the mount file

Extra flags are set from the flags field:

L	LOCKED
H	HIDDEN
P	PRESET
O	OPEN

- Mounts with the LOCKED flag do not get saved into the user mounts file, and so can be used in read-only, master mounts files used at boot time.
- Mounts with the HIDDEN flag set do not appear in the mounts list submenu, in the 'servers' window, or on the icon bar.

  The HIDDEN flag is provided to allow for 'system' mounts which are used in a networked boot sequence (e.g. scrap area) but which remain hidden from the user (not appearing on the icon bar, etc). In conjunction with the LOCKED flag for master mounts file(s), you can setup a very flexible boot sequence that has its complexity hidden from the end user. Note that the HIDDEN flag will not prevent servers/mounts appearing in the 'servers' window that have been auto-located by a network scan (if this is enabled see *Application configuration* on page 16).
- The PRESET flag allows extra servers/mounts to be presented in the network servers window – perhaps non-local servers which protocols cannot automatically scan for. Mount file entries with the PRESET flag set do not appear in the mounts list submenu unless they are currently connected. This flag has a higher priority than HIDDEN, and mounts file entries that mistakenly have both set will be treated as PRESET.
- Mounts with the OPEN flag set do not open directory displays when they are mounted – i.e. if the flag is not set the action is to open the window automatically, thus making this version compatible with Release 1. The default action is to set the flag for new mounts.

To set these flags, you must edit the mounts file to include the strings `L`, `P`, `O` or `H` flags field.

Note that PRESET and HIDDEN are mutually exclusive, since the purpose of PRESET is to have mounts that cannot be located automatically (for whatever reason) appear in the servers window, and the purpose of HIDDEN is to stop them appearing.

OmniSetup

This application gives the network manager an easy way of selecting which network protocols are in use. Note that a choice has to be made between NetBEUI and IP for Lan Manager access. The option to switch on the display of all network servers is also available, as well as the options given below. See also **Configure LanManTransport* on page 76 and **Configure* FS on page 77.

Application configuration

To allow configuration to be stored on a per-user basis, all configuration files used by the OmniClient application are accessed through the system variable Omni$Path (by default set to `!Omni.Files`). This directory contains three files:

- `Mounts` (see *Mounts file* on page 14)
- `Extensions` (contains the Filetype extension mappings list)
- `Startup` (obey file).

The Startup obey file is executed when the application is loaded, and may contain commands to load protocol modules, set Omni$User variables, etc. It can also set the system variable Omni$Options which allows the default configuration of the application in a similar manner to that of Acorn !Edit and Edit$Options. Currently defined options are:

Option	Action	
`sN`	Sort type	N = 0 (None) N = 1 (Name) N = 2 (Protocol/Name)
`dN`	Display type	N = 0 (Large icons) N = 1 (Small icons) N = 2 (Full info)
`a`	Auto-location of network servers ON	

This is primarily for further reading because the Sort type, Display type and Auto-location of network servers options can be modified in `!Omni.Files.Startup`. However running !OmniSetup will display these options in a dialogue box permitting a more efficient selection procedure, but the option is still there to modify your choices as you wish.

The `a` option allows the OmniClient software to scan the network to enable the display of all network servers. If the `a` option is not present, only those mounts that are specified in the Mounts file will be displayed.

For example:

```
Set Omni$Options s0 d1
Set Omni$Options a s0 d1
```

You may wish to edit the Startup file to comment out protocol modules that are not required, or alter the Omni$Options settings, but it is recommended that you use !OmniSetup to do this.

Setting the user name

You can use the system variable Omni$User*FilingSystem* to set the value of the user name. If the variable is not set but Omni$User is set then the value of this will be used as the default user name. This should ensure that in general on a user's private machine only the password need be typed.

For example:

```
Omni$UserNFS JSmith
```

Printing using OmniPrint

You can use Acorn's standard version of !Printers to access remote printers located or controlled by NT or other Lan Manager-type network printers, or any other network printer type supported.

1 Click Menu over the Printer control window and choose the **Connection...** option. This displays the Connections window.

2 Click on the **File** option.

3 Enter a filename of the following form:

`OmniPrint#`*`FSName`*`;`*`Server`*`;`*`Printer`*`;`
`User``;`*`Password`*`;`*`Options`*`;`*`Size`*`:`

4 Save the new !Printers configuration.

The fields you must fill in are the *`FSName`*, *`Server`* and *`Printer`* names, although the latter two may be the same for some situations (e.g. a stand-alone AUN printer). Note that *`User`*, *`Password`*, *`Options`* and *`Size`* are all optional fields. The file name must be terminated by a colon.

Any filename given is ignored – the critical data is contained in the special field. Fields are semi-colon separated, and illegal (RISC OS) characters should be encoded as follows:

RISC OS character	encoding characters
SPACE	~_ (underscore)
"	~'

RISC OS character	encoding characters
| (solidus)	~l (lowercase L)
:	~1 (numeric one)
;	~!
,	~.
~	~~

Examples:

```
OmniPrint#NFS;ant;lp;nas:

OmniPrint#LanMan;NTserver;NTprint;guest;guest:
```

For protocols that require the file size prior to the transmission of any data, an optional field *Size* can be included in the special field (none of the protocols in this release need the size field).

Alternative printing using NFS

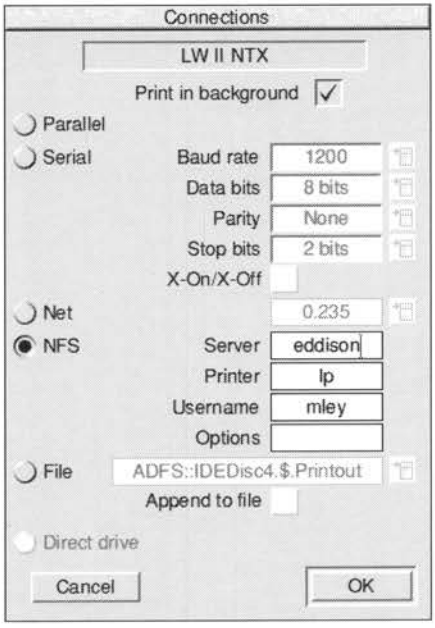
You can also use NFS software in conjunction with the standard RISC OS Printers application to send print jobs to printers connected to remote NFS servers.

Loading the NFS software

Obviously the Printers application needs to use NFS software to send the print job to an NFS server. You should ensure that OmniClient is set up to use NFS before you run the Printers application.

Setting up the Printers application

Set up the printer in the same way as any other (see the chapter on setting up printers in the RISC OS 3 *User Guide*) except for the **Connections** window, where you should choose **NFS** and type in some information:



- The **Server** must be running a utility known as the `pcnfsd` daemon. You need this same utility to use the server as a name server, so:

  Any server you can use as a name server you can also use as a print server, and vice versa.

- The **Printer** must be the name of a printer that the server can access. If you don't give a printer name, the server will use the printer named `lp` – the default name for a UNIX printer.

- The **Username** gets passed to the program that does the printing (`lpr` in the case of UNIX), and is used to establish your access rights to the printer. If you leave this out the server uses a blank user name; you can do so if there are no restrictions on printer use at your site.

- The **Options** also get passed to the program that does the printing. If you omit them, no options are forwarded.

Don't forget to choose **Save choices** from the Printers icon bar menu if you want the new setup to be saved.

Using the Printers application

Once you've got everything correctly set up and running, you can use the Printers application just as you always have. Any files you drop onto its icon get converted to the correct format for the remote printer (e.g. PostScript), and then forwarded to it via the NFS server.

3 !BootNet and SetStation

BootNet

BootNet is an application that loads the software modules required to allow Level4-type access over Ethernet and help establish the IP address of the station for use with AUN.

!BootNet may be run

- on start-up to allow Ethernet access to Level 4 file servers
- after !Internet is run, when Level 4 file servers are used on a configured TCP/IP network.

Normally this software is loaded automatically from the Ethernet card ROM or from system ROM, except when it is run after !Internet where the !BootNet application is required.

Where the software is resident on Ethernet card ROM, then the modules are loaded if the configuration requires it: `*Configure BootNet on` loads the modules from Ethernet Card ROM, but on an Acorn Access card some of the modules are loaded even if BootNet is configured off.

If you wish to access Level 4 and your Ethernet cards contain DCI2 modules, configure BootNet off and use the copy of !BootNet provided.

If you are running a full TCP/IP network, communicating with NFS servers, for example, then you should ensure that the file `!BootNet.!Configure` does not re-load the driver. See the comments inside that file for full information. The Release Note contains hints and tips about this.

SetStation

Specifically designed to assist discless client start-up, SetStation sets the full four byte IP address in CMOS RAM. This is required if a client is connected to a fully-configured TCP/IP network, as would be the case if you were connected to the Internet via an ISDN Router. The new Bootnet module checks for a valid AUN (if BootNet is configured on) or full TCP/IP address (if configured off) if it finds valid numbers.

Note that a Power-On Delete will wipe the top three bytes of the IP address, which will have to be reset if this occurs.

Discless client stations

If you have discless client stations on your network, you can configure them to boot from a suitable server. For Level 4 and Acorn Access network protocols, this procedure is described in the Level 4 and Access documentation. See also the Appendix *OmniClient and discless workstations* on page 87.

Briefly, any client machine containing an Ethernet card with AUN software is able to boot from a Level 4 server. The following commands:

```
*Configure boot
*Configure filesystem net
*Configure fs Server1
```

will make a machine attempt to log on to a fileserver named 'Server1" as user 'Boot' and attempt to run a file or application entitled !ArmBoot. Ensure the user boot has the option set to 'Run'.

Similarly, for Acorn Access the commands would be:

```
*Configure boot
*Configure filesystem share
```

and the leftmost of the shared discs on the icon bar is used. The client machine looks for a file or application called !Shareboot and attempts to run it. To boot from a shared disc, you will of course need to have an Acorn Access card fitted to your machine.

You can use a shared disc to load a copy of OmniClient and then establish a link to an NT server. However, if you buy an Ethernet card with OmniClient on board, you may boot from an NT Server directly.

To boot from an NT Server, you must first decide whether you wish to use NetBEUI or IP as the underlying protocol – both are supported by NT. NetBEUI is lightweight and fast, but IP is more robust and is suitable for networks with gateways and routers in them.

Then configure the client station, for example

```
*Configure boot
*Configure filesystem Lanman
*Configure fs Server2
```

and ensure there is an application or file !ArmBoot available on the server which then runs a copy of !Omni, to set up the user environment.

4 Introduction to TCP/IP

OmniClient enables you to connect Acorn RISC OS computers to a TCP/IP network, and to access computers on that network in a variety of ways.

You can use Internet and OmniClient NFS functionality to connect RISC OS computers directly to an existing Ethernet-based TCP/IP network. To do so, your RISC OS computers will need to have an Acorn Ethernet interface fitted.

Finding out more

For general information on the use of a RISC OS computer and its desktop interface, see the *Welcome Guide* and RISC OS 3 *User Guide* supplied with it.

For details of how to use the TCP/IP Protocol Suite (Release 2), see the TCP/IP *Protocol Suite (Release 2) User Guide*.

For details of how to use the programming interfaces provided by the TCP/IP Protocol Suite (Release 2), see the TCP/IP *Protocol Suite (Release 2) Programmer's Guide*, available separately from Acorn Developer Support and on the Acorn ftp site (although unsupported). This includes a disc of useful C libraries.

You should also see any relevant documentation supplied with other computers you plan to be on your TCP/IP network.

Finally, you can get more detailed information from *Internetworking with* TCP/IP. Douglas Comer (1988) Prentice-Hall, Englewood Cliffs, NJ, USA.

TCP/IP concepts

When you install OmniClient, you will have to assign certain names and numbers to the computers on your TCP/IP network, and to their network interfaces. This section explains those names and numbers.

If you've already got a TCP/IP network running...

If you've already got a TCP/IP network running on your site, you should already have naming and numbering schemes set up. Make sure that any names and numbers you assign conform to this scheme, and that you first contact the person who administrates their allocation.

Host names

Each computer on your network must have a *principal host name*, or *host name* for short. Your users will use this name to refer to the computer. The name must be unique on your site – you can't have two computers with the same name.

It helps your users if each host name is easy for them to remember. One way to do this is to use a theme, such as planets (e.g. `saturn`, `uranus`); another way is to give names that have some relationship to the computer's function on your network (eg `accounts1`, `accounts2`). You can combine these ideas – so you might name the graphics department's computers after famous artists (eg `turner`, `vangogh`).

Interface names

Each network interface in each computer – whether it be an Ethernet or Econet interface – must also have an *interface name*. Again, this name must be unique on your site – you can't have two interfaces with the same name.

If there's only a single interface in a computer it's normal to use just the principal host name as the interface name. If there are two interfaces in a name it's normal to refer to the principal host name in each interface name: so a machine named `saturn` may have interfaces named `saturn_eco` and `saturn_ether`.

Internet addresses, netmasks and subnets

Furthermore, each interface must also have a unique numerical address, known as its *Internet address*. It is this address that the TCP/IP protocol uses to communicate; if a user specifies a host name or interface name, the software automatically converts it to an Internet address.

An Internet address is four bytes long. These four bytes are split into fields:

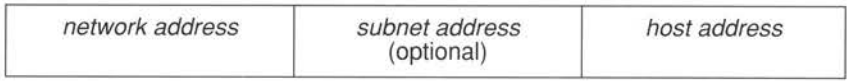


Figure 4.1 Fields within an Internet address

The *network address* identifies an entire network (which is typically a whole site). The *subnet address* is optional, and identifies a local network that forms part of the main network. The *host address* identifies a host on that network.

A *netmask* specifies the portion of the address used by the network and subnet addresses. For example, if the network address is held in the top byte, and no subnets are used, the netmask would be 0xFF000000 (i.e. FF000000 hexadecimal) or 255.0.0.0 decimal.

Unlike the interface name, the Internet address must be unique on all networks with which the interface will ever communicate.

If you plan to connect to other sites...

If you plan to connect to other sites over the Internet, you need to ensure not only that Internet addresses are unique to your site, but also that they are unique to the entire Internet. The Internet already connects together thousands of sites, each with many hosts. Clearly it's impossible to keep so many Internet addresses unique on an informal basis. Consequently there is an administrative body responsible for allocating network addresses. You must contact them before you use the Internet to connect to other sites; write or send email to:

DDN Network Information Center
SRI International
Room EJ217
333 Ravenswood Avenue
Menlo Park, CA94025
USA email: HOSTMASTER@SRI-NIC.ARPA

Depending on the size of your network, you will be allocated a Class A, B or C address: these use respectively the top one, two or three bytes for the network address. It is your responsibility how you use the remaining unallocated bytes to specify subnets and hosts. For example, let's say you've been allocated a Class B network address, and so have two bytes free for your own use:

- If all your site's computers are connected to a single local network, you won't need to use subnets, and so might use all two bytes for the host address (allowing 64k hosts). In this case, you'd use a netmask of 0xFFFF0000.

  Note that an Ethernet generally behaves as a single network, even if it is made up of multiple segments of cable (unless divided by routers).

- However, if your site's computers are connected to different local networks (such an Ethernet and some Econets) you'll need to use subnets. You might decide to use 5 bits for the subnet address (allowing 32 subnets), and the remaining 11 bits for the host address (allowing 2000 hosts). In this case, you'd therefore use a netmask of 0xFFFFF800.

  Note that separate Econets (i.e. those not connected together by Econet bridges) form separate subnets.

If you don't plan to connect to other sites...

If you don't plan to connect to other sites over the Internet, all you need to do is to ensure that the interface's Internet address is unique on your own site. We suggest you use the following scheme:

Figure 4.2 Suggested local TCP/IP numbering scheme

Number your local networks from ten: for example, you might number your Ethernet as net 10, and an Econet as net 20. Likewise, number your hosts (not your interfaces) from 10. Your available Internet address and their meanings would be as follows:

Ethernet	Meaning	Econet	Meaning
10.0.0.1	host 1 on Ethernet	20.0.0.1	host 1 on Econet
10.0.0.2	host 2 on Ethernet	20.0.0.2	host 2 on Econet
10.0.0.3	host 3 on Ethernet	20.0.0.3	*and so on up to...*
10.0.0.255	host 255 on Ethernet	20.0.0.255	host 255 on Econet
10.0.1.0	host 256 on Ethernet	20.0.1.0	host 256 on Econet
10.0.1.1	host 257 on Ethernet	20.0.1.1	*and so on...*

Of course if a machine has only got one interface fitted, you'll only use one of the addresses assigned to it; one of the addresses will be wasted. But if you later upgrade the machine to add a second interface, you already have a meaningful Internet address reserved for it.

Physical Addresses

Each interface also has a six byte physical address (alternatively known as its MAC address). You shouldn't need to do anything to set this up, because

- an Ethernet interface's physical address is unique worldwide, and is set in the hardware at the time of manufacture.
- an Econet interface's physical address is based on its network and station numbers. So long as you've correctly installed the Econet, these should be unique to your site.

Further Information

If you require any further information, please contact Acorn Customer Services' ftp site to obtain Application Note 283.

5 The Internet application

This chapter tells you how to set an IP address. It also gives you information about a variety of modules and * commands that duplicate UNIX commands. If you're a more experienced Internet user you may well find these useful.

Setting an IP address

The Internet application allows you to communicate with other computers on a fully-configured TCP/IP network. For this to work you need to use !InetSetup to reflect the state of your network, and your machine.

Changing the setup

Firstly you must run !InetSetup and edit the displayed fields so that it specifies the machine (host) on the network to which it is connected.

A number of options are then displayed which you need to modify; the first one is **Enable TCP/IP Protocol Suite**. Choose this option and all the other options can now be accessed.

Choosing **Interfaces** should identify the network card in your machine and ticking the box will allow you to configure the options – for example, setting your IP address and the appropriate Netmask.

Routing allows you to select which host you will be using as a gateway, so that the computer knows how to access machines on other networks.

Host names requires your unique Host name to be entered, as well as server names and a choice of resolver modules if required, so that Internet names can be converted to IP numbers by a remote name server machine.

Editing the hosts file

You will also need to edit the Hosts file – this can be done in **Host names**. This forms the link between the host name and the IP address (see *If you plan to connect to other sites...* on page 25 for details of obtaining a unique IP address). The file has the following format:

```
127.0.0.1 loopback localhost loghost
1.0.0.1               ahost
2.0.0.1               anotherhost
2.0.0.2               agateway
3.0.0.1               remotehost
```

You need to specify computers in this list to which software may need to refer to the remote computer by name. This may include server and gateway machines. Do **not** delete the loop-back entry.

Running the Internet application

To use any of these commands you need to run the Internet application. There are three ways you can do this:

- double-clicking on the Internet icon from the desktop
- typing a command at the command line
- including a command in a boot file.

In both the latter cases you must use a command of the form:

`*Run` *`Internet_pathname`*`.!Internet`

where *`Internet_pathname`* is the rest of the pathname to the Internet application. You **must** start the command with `*Run`; if you don't RISC OS won't know which filing system holds the Internet application.

Getting a command line

If you're using the desktop, there are three ways of getting a command line so you can enter * Commands:

- Open a task window by pressing Ctrl-F12, or by choosing **Task window** from the Task Manager's icon bar menu.
- Temporarily leave the desktop by pressing F12, or by choosing ***Commands** from the Task Manager's icon bar menu.
- Permanently leave the desktop by choosing **Exit** from the Task Manager's icon bar menu.

Internet module * Commands

The core of the Internet application is a module named `Internet`. It is stored as the `!RunImage` file within the application. It provides three * Commands:

Command	Use	Page
*InetGateway	Toggle IP packet forwarding	40
*InetInfo	Display Internet module internal statistics	41

Running the Internet application loads the Internet module; you can then use the above * Commands.

Absolute programs

There are also a set of commands that are provided as Absolute files (i.e. each file is a program that performs one command) in the `bin` subdirectory of the Internet application. These are:

Command	Use	Page
*ARP	Address resolution display and control	31
*IfConfig	Configure network interface parameters	36
*Ping	Send ICMP ECHO_REQUEST packets to network hosts	42
*Route	Manually manipulate the routing tables	45

Running the Internet application adds the `bin` subdirectory to the system variable `Run$Path`. RISC OS then knows where to find the above * Commands, so you can use them directly from the command line.

Although the above are actually programs, in the section that follows we've treated them as * Commands because you're most likely to use them in just the same way.

Ethernet driver module * Commands

The Ethernet driver modules (held in the `drivers` subdirectory of the application) each provide a single * Command to give information.

Ether3 refers to the A5000 style podules and EtherB to the Risc PC network card. Cards produced by different manufacturers may have differences – such as EtherH (i-cubed) – but all are suitable.

The `Ether3` module also provides a configuration command:

Command	Use	Page
*Configure Ether3 *Configure EtherB	Sets the configured state of Acorn Ethernet 3 cards	33
*E*n*Info	Display details of physical interface activity, including their physical addresses	34

Running the Internet application loads any appropriate Ethernet driver modules; you can then use any of the above * Commands that are relevant to your interface(s).

Address resolution display and control

Syntax

```
*ARP host
*ARP -a
*ARP -d host
*ARP -s host phys_addr [temp] [pub]
*ARP -f filename
```

Parameters

host	an Internet host specified either by name (which must be present in the host name data base <InetDBase$Path>hosts) or by address (using the standard Internet dot notation)
phys_addr	the physical address of *host* given in Ethernet format (i.e. six hexadecimal bytes separated by colons)
filename	the full pathname of a file containing multiple entries to be set in the ARP table

Use

The ARP program displays and modifies the Internet-to-Physical-address translation tables located in the Internet module and used by the address resolution protocol ARP.

With no flags, the program displays the current ARP entry for *host*.

The -a flag makes the program display all the ARP entries currently in its table.

The -d flag makes the program delete an entry for *host*.

The -s flag makes the program create an ARP entry for the host called *host* with the physical address *phys_addr*.

- The entry will be permanent unless the word temp is given in the command.
- The entry will be 'published' if the word pub is given. This system will then act as an ARP or Reverse ARP server, responding to requests for *host*'s physical address even though the host address is not its own.

The -f flag causes the file *filename* to be read and multiple entries to be set in the ARP tables. Entries in the file should be of the form:

host phys_addr [temp] [pub]

with argument meanings as given above.

If you don't know the physical address of an interface, you can use the *InetInfo command to find it.

For your reference, the physical address of an Econet interface takes the form:

00.00.00.00.*station_number*.*net_number*

Examples

```
*ARP tpl
*ARP -a
*ARP -d 01.01.01.01
*ARP -s tpl 01.01.01.01.01.01 temp
*ARP -f adfs::HardDisc.$.Internet.ARP_Table
```

Related commands

None

*Configure Ether3 EtherB

Sets the configured state of Acorn Ethernet 3 cards

Syntax

```
*Configure Ether3 Enable|Disable|Default|Terse|Verbose [card]
```

Parameters

card	the expansion card's number, as given by *Podules

Use

*Configure Ether3 sets the configured state of all fitted Acorn Ethernet 3 cards, or of a single *card* specified by its expansion card slot number.

*Configure Ether3 Enable permits the interface to be used – which is the default – whereas *Configure Ether3 Disable prevents the interface from being used.

*Configure Ether3 Default sets the configured state to the default: namely, Enabled.

Example

```
*Configure Ether3 Default 0
```

Related commands

*Status Ether3 (page 47)

*EnInfo

Display details of physical interface activity, including their physical addresses

Syntax

`*E1Info`	Display details for all Acorn Ethernet 1 cards
`*E2Info`	Display details for all Acorn Ethernet 2 cards
`*E3Info`	Display details for all Ethernet 3 cards
`*EBInfo`	Display details for all Atomwide-based Network slot cards
`*EHInfo`	Display details for all i-cubed (I^3) cards

Use

*EnInfo displays details of physical interface activity, including the physical addresses of all Ethernet interfaces of the relevant type that are fitted to the computer.

Most of the information displayed is runic in nature. It is presented mainly as an aid to trouble-shooting, should you require it.

If you are using an interface that is not listed above, you should consult its documentation to see if its manufacturer has provided a similar command; we expect most will do so.

Example

```
*E3Info

Ether3 interface statistics

ea0: 8005 16 bit MEMC1a, slot 2, up, hardware address 00:02:07:00:A3:48
```

```
Interface driver    : ea
Interface unit      : 0
Interface location  : Expansion slot 2
Interface address   : 00:02:07:00:A3:48
Interface controller: 8005 16 bit MEMC1a
Initialise time     : Wed Mar  1 11:32:02 1995
Running time        : 2 minutes, 5 seconds.
Packets sent        : 4362
Packets received    : 4830
Bytes sent          : 2276647
Bytes received      : 2352828
Send errors         :
Receive errors      :
Broadcasts sent     :
Broadcasts received : 298


Multicasts sent     :
Multicasts received : 111
Monitor sends       :
Monitor receives    :
Send interrupts     : 4362
Receive interrupts  : 4830
Delivered packets   : 4835
Undelivered packets :
Sends too large     :
Send max collisions :
Send overflows      :
Sends blocked       :
Receives too small  :
Receives too large  :
Incomplete receives :
Receive CRC errors  :
Lack of mbufs       :
Receive overflows   :

Standard clients:

Frame = &0800, ErrLvl=00, AddrLvl=01
Frame = &0806, ErrLvl=00, AddrLvl=01
Frame = &8035, ErrLvl=00, AddrLvl=01

IEEE 802.3 client:

Frame = &0000, ErrLvl=00, AddrLvl=02

Log: Ether3 messages can appear here
```

Related commands

*InetInfo (page 41)

*IfConfig

Configure network interface parameters

Syntax

```
*IfConfig interface address_family [address
[dest_address]] [parameters]
*IfConfig interface [protocol_family]
*IfConfig -a
*IfConfig -au
*IfConfig -ad
```

Parameters

address	for the DARPA-Internet family, the address is either a host name present in the host name data base, hosts, or a DARPA Internet address expressed in the Internet standard dot notation. For the Xerox Network Systems™ family, addresses are `net:a.b.c.d.e.f`, where `net` is the assigned network number (in decimal), and each of the six bytes of the host number, `a` through `f`, are specified in hexadecimal. The host number may be omitted on 10Mb/s Ethernet interfaces, which use the hardware physical address, and on interfaces other than the first. For the ISO family, addresses are specified as a long hexadecimal string, as in the Xerox family. However, two consecutive dots imply a zero byte, and the dots are optional, if the user wishes to count out long strings of digits in network byte order.
address_family	Specifies the address family which affects interpretation of the remaining parameters. Since an interface can receive transmissions in differing protocols with different naming schemes, specifying the address family is recommended. The address or protocol families currently supported are inet, iso, and ns.
interface	The interface parameter is a string of the form name unit, for example, en0.

Use

IfConfig is used to assign an address to a network interface and/or configure network interface parameters. IfConfig must be used at boot time to define the network address of each interface present on a machine; it may also be used at a later time to redefine an interface's address or other operating parameters.

You can set the following parameters with the IfConfig program:

`alias`	Establish an additional network address for this interface. This is sometimes useful when changing network numbers, and one wishes to accept packets addressed to the old interface.
`arp`	Enable the use of the Address Resolution Protocol in mapping between network level addresses and link level addresses (default). This is currently implemented for mapping between DARPA Internet addresses and 10Mb/s Ethernet addresses.
`-arp`	Disable the use of the Address Resolution Protocol.
`broadcast` (Inet only)	Specify the address to use to represent broadcasts to the network. The default broadcast address is the address with a host part of all 1's.
`debug`	Enable driver-dependant debugging code; usually this turns on extra console error logging.
`debug`	Disable driver-dependant debugging code.
`delete`	Remove the network address specified. This would be used if you incorrectly specified an alias, or it was no longer needed. If you have incorrectly set an NS address having the side effect of specifying the host portion, removing all NS addresses will allow you to respecify the host portion.
`dest_address`	Specify the address of the correspondent on the other end of a point to point link.
`down`	Mark an interface 'down'. When an interface is marked 'down', the system will not attempt to transmit messages through that interface. If possible, the interface will be reset to disable reception as well. This action does not automatically disable routes using the interface.
`ipdst`	This is used to specify an Internet host which is willing to receive ip packets encapsulating NS packets bound for a remote network. An apparent point-to-point link is

constructed, and the address specified will be taken as the NS address and network of the destination. IP encapsulation of CLNP packets is done differently.

`metric n` — Set the routing metric of the interface to n, default 0. The routing metric is used by the routing protocol routed. Higher metrics have the effect of making a route less favourable; metrics are counted as addition hops to the destination network or host.

`mtu n` — Set the maximum transmission unit of the interface to n; the default is interface-specific. The mtu is used to limit the size of packets that are transmitted on an interface. Not all interfaces support setting the mtu, and some interfaces have range restrictions.

`netmask mask` — (Inet and ISO): specify how much of the address to reserve for subdividing networks into sub-networks. The mask includes the network part of the local address and the subnet part, which is taken from the host field of the address. The mask can be specified as a single hexadecimal number with a leading 0x, with a dot-notation Internet address, or with a pseudo-network name listed in the network table networks. The mask contains 1's for the bit positions in the 32-bit address which are to be used for the network and subnet parts, and 0's for the host part. The mask should contain at least the standard network portion, and the subnet field should be contiguous with the networkportion. (See eon.)

`nsellength n` — This specifies a trailing number of bytes for a received NSAP used for local identification, the remaining leading part of which is taken to be the NET (Network Entity Title). The default value is 1, which is conformant to US GOSIP. When an ISO address is set in an IfConfig command, it is really the NSAP which is being specified. For example, in US GOSIP, 20 hex digits should be specified in the ISO NSAP to be assigned to the interface. There is some evidence that a number different from 1 may be useful for AFI 37 type addresses.

`trailers` — Request the use of a 'trailer' link level encapsulation when sending (default). If a network interface supports trailers, the system will, when possible, encapsulate outgoing messages in a manner which minimizes the number of memory to memory copy operations

	performed by the receiver. On networks that support the Address Resolution Protocol (see arp; currently, only 10 Mb/s Ethernet), this flag indicates that the system should request that other systems use trailers when sending to this host. Similarly, trailer encapsulations will be sent to other hosts that have made such requests. Currently used by Internet protocols only.
`-trailers`	Disable the use of a 'trailer' link level encapsulation.
`link[0-2]`	Enable special processing of the link level of the interface. These three options are interface specific in actual effect, however, they are in general used to select special modes of operation. An example of this is to enable SLIP compression. Currently, only used by SLIP.
`-link[0-2]`	Disable special processing at the link level with the specified interface.
`up`	Mark an interface 'up'. This may be used to enable an interface after an IfConfig down. It happens automatically when setting the first address on an interface. If the interface was reset when previously marked down, the hardware will be re-initialized.
`IfConfig -a`	This displays information on all interfaces. When followed by a configuration parameter, it will also set the configuration on all interfaces.
`IfConfig -au`	Similar to IfConfig -a, except it only affects interfaces that are currently marked as up. Conversely, IfConfig -ad affects only interfaces that are marked down. IfConfig displays the current configuration for a network interface when no optional parameters are supplied. If a protocol family is specified, IfConfig will report only the details specific to that protocol family. Only the super-user may modify the configuration of a network interface.

Example

```
*IfConfig ea0 inet tp1 up
```

Related Commands

None

*InetGateway

Toggle IP packet forwarding

Syntax

```
*InetGateway 1|0
```

Use

*InetGateway may be used to enable (*InetGateway 1) or to disable (*InetGateway 0) IP layer packet forwarding (i.e. gateway operation) if multiple network interfaces are present.

The default state is off.

Example

```
*InetGateway 1
```

Related commands

None

*InetInfo

Display Internet module internal statistics

Syntax

```
*InetInfo [r][i]
```

Use

*InetInfo displays detailed information about Internet module activity. By default it only gives details of internal resource usage and protocol activity using the options:

`r`	display only internal resource information (the default)
`i`	display only interface information

Most of the information displayed is runic in nature. It is presented mainly as an aid to trouble-shooting, should you require it.

The `i` option used in Release 1 of the TCP/IP Protocol Suite to give information on an interface has been replaced by the *E*n*Info command.

Example

```
*InetInfo rp
```

Related commands

*E*n*Info (page 34)

*Ping

Send ICMP ECHO_REQUEST packets to network hosts

Syntax

```
*Ping [-dfnqrvR] [-c count] [-i wait] [-l preload] [-p
pattern] [-s packetsize]
```

Parameters

`-c count`	stop after sending and receiving count ECHO_RESPONSE packets.
`-i wait`	wait within seconds between sending each packet. The default is to wait for one second between each packet. This option is incompatible with the -f option.
`-l preload`	if preload is specified, ping sends that many packets as fast as possible before falling into its normal mode of behaviour.
`-p pattern`	you may specify up to 16 'pad' bytes to fill out the packet you send. This is useful for diagnosing data-dependent problems in a network. For example `-p ff` will cause the sent packet to be filled with all ones.
`-s packetsize`	specifies the number of data bytes to be sent. The default is 56, which translate into 64 ICMP data bytes when combined with the 8 bytes of ICMP header data.

Use

An Internet can be a large and complex aggregation of network hardware, connected together by gateways. Tracking a single-point hardware or software failure can be difficult. The Ping program utilises the ICMP protocol's mandatory ECHO_REQUEST datagram to elicit an ICMP ECHO_RESPONSE from a host or gateway. ECHO_REQUEST datagrams (or pings have an IP and ICMP header, and then an arbitrary number of padding bytes used to fill out the packet.

The optional -d parameter sets the SO_DEBUG option on the socket being used. The optional -f parameter causes a flood ping. It outputs packets as fast as they come back or one hundred times per second, whichever is more. For every ECHO_REQUEST sent a period '.' is printed, while for every ECHO_REPLY received a backspace is printed. Providing a rapid display of the number of packets dropped, but only a super-user may use this option and can be hard on a network.

The optional -n parameter specifies numeric output only. No attempt will be made to look up symbolic names for host addresses.

The optional -q parameter specifies quiet output. Nothing is displayed except the summary lines at start-up time and when finished.

The optional -r parameter bypasses the normal routing tables and sends directly to a host on an attached network. If the host is not on a directly-attached network, an error is returned. This option can be used to ping a local host through an interface that has no route through it (e.g. after the interface was dropped by routed).

The optional -v parameter specifies verbose output. ICMP packets other than ECHO_RESPONSE that are received are listed. The optional -R parameter specifies the recording of the route. Includes the RECORD_ROUTE option in the ECHO_REQUEST packet and displays the route buffer on returned packets. Note that the IP header is only large enough for nine such routes. Many hosts ignore or discard this option.

When using ping for fault isolation, it should first be run on the local host, to verify that the local network interface is up and running. Then, hosts and gateways further and further away should be 'pinged'.

ICMP packet details

An IP header without options is 20 bytes. An ICMP ECHO_REQUEST packet contains an additional eight bytes worth of ICMP header followed by some data. When a packetsize is given, this indicated the size of this extra piece of data (default is 56). The amount of data received inside of an IP packet of type ICMP ECHO_REPLY will always be eight bytes more than the requested data space (the ICMP header).

If the data space is at least eight bytes large, ping uses the first eight bytes of this space to include a timestamp which it uses in the computation of round trip times. If less than eight bytes of pad are specified, no round trip times are given.

Duplicate and damaged packets

Ping will report duplicate and damaged packets. Duplicate packets should never occur, and seem to be caused by inappropriate link-level retransmissions. Duplicates may occur in many situations and are rarely (if ever) a good sign, although the presence of low levels of duplicates may not always be cause for alarm.

Damaged packets are obviously serious cause for alarm and often indicate broken hardware somewhere in the ping packet's path (in the network or in the hosts).

Trying different data patterns

The (inter)network layer should never treat packets differently depending on the data contained in the data portion. However, data-dependent problems can get into networks and remain undetected. In many cases the pattern that will have problems is something that doesn't have sufficient 'transitions', such as all ones or all zeros. It's not usually enough to specify a data pattern of all zeros on the command line because the pattern of interest is at the data link level, and the relationship between what you type and what the controllers transmit can be complicated.

This means that if you have a data-dependent problem you will probably have to do a lot of testing to find it. If you are lucky, you may manage to find a file that either can't be sent across your network or that takes longer to transfer than other similar size files. Then test using -p option of Ping examine for file repeated patterns.

Example

```
*Ping -v tp1 1024 20
```

Related Commands

None

*Route

Manually manipulate the routing tables

Syntax

```
*Route [-e] [-n] [-v] [-q] add | delete | change | get |
monitor | flush [-net | -host] destination gateway flags
*Route [-e] [-n] [-v] [-q] flush | monitor
```

Parameters

destination	an Internet hot or network specified either by name (which must be present in the respective host or network name data base <InetDBase$Path>...) or by address (using the standard Internet dot notation)
gateway	the next hop gateway to which packets should be addressed
flags	various commands used to manipulate the routing

Use

You can use the Route program manually to manipulate the Internet module's network routing tables.

Routes to a particular host are distinguished from those to a network by interpreting the Internet address associated with the destination. The optional keywords net and host force the destination to be interpreted as a network or a host, respectively. Otherwise, if the *destination* has a local address part of 0 or if there is a symbolic name of a network, the route's presumed to be a host. All symbolic names specified from a destination or a gateway are looked up first as a host name; if this fails, the name is then looked up as a network name.

The -e option makes Route place any fatal error report string in the system variable Inet$Error instead of writing it to the standard output.

The -n option bypasses attempts to print host and network names symbolically when reporting actions. However, translating between symbolic names and numerical equivalents can be time consuming and may require correct operation of the network.

The -v option (verbose) allows the printing of additional details.

The -q option allows the suppression of all output.

There are many flags which can be attached to the routing command line and they are as follows:

-cloning	generates a new route on use
-iface	destination is directly reachable
-static	is a manually added route
-nostatic	is a pretended route added by kernel or daemon
-reject	to emit an ICMP unreachable when matched
-blackhole	silently discards packets during updates
-rtt n	shows initial round-trip time
-rttvar n	shows initial RTT variance
-mtu n	shows initial MTU
-expire n	shown expiry time

Examples

```
*Route -n monitor
```

Related Commands

None

*Status Ether3
EtherB

Displays the configured state of Acorn Ethernet 3 cards

Syntax

```
*Status Ether3
```

Use

*Status Ether3 displays the configured state of all fitted Acorn Ethernet 3 cards.

Example

```
*Status Ether3
ether3 enabled 0
```

Related commands

*Configure Ether3 (page 33)

*Status EtherH

Displays the configured state of Acorn Ethernet H cards

Syntax

```
*Status EtherH
```

Use

*Status EtherH displays the configured state of all fitted Acorn Ethernet H cards.

Example

```
*Status EtherH
etherH virtual on
```

Related commands

*Configure Ether (page 33)

6 NFS file mapping

There are a number of differences between the UNIX and RISC OS models of a filing system, the more important being:

- length of filenames
- use of special characters in filenames
- numbers of attribute bits stored with files
- meaning of attribute bits
- use of file types
- soft links.

Because of these clashes changes have to be made when mapping RISC OS file names and attributes to UNIX ones, and vice versa. Generally the changes made when mapping one way are reversed when mapping the other way, so the system is as transparent as possible if only viewed from RISC OS. If you view the files using the remote filestore though, you'll notice some differences.

This chapter outlines how the mapping of file names takes place, and what differences you'll notice between the RISC OS view of a file and the UNIX view.

NFS – its file mapping from RISC OS to UNIX

This section describes how RISC OS NFS maps files from RISC OS to UNIX.

Filenames

Character translation

The first change RISC OS NFS makes to a filename is to translate the character '/' (the UNIX directory separator) to '.', for example:

RISC OS name	UNIX name
fred/c	fred.c
/profile	.profile

File type extensions

RISC OS NFS then adds a filename extension to store the RISC OS file type.

You can set the extension used for any given file type to one of your choice. To do so you must edit the `extensions` file, held within the Internet application. See section *Editing the extensions file* on page 56.

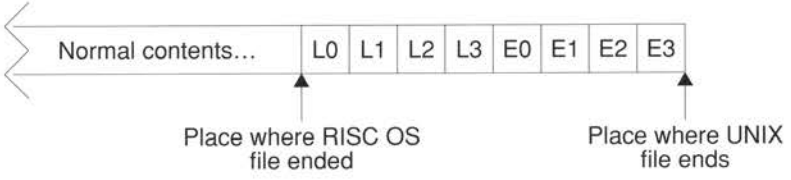
If you haven't set up a filename extension for a given file type, then a default extension gets used instead. The default mapping of a RISC OS file called `Fred` is as follows:

RISC OS type	UNIX name	Notes
Text	Fred	
UNIX Ex	Fred	
Draw (&AFF)	Fred,aff	
Obey (&FEB)	Fred,feb	and other file types similarly...
dead †	Fred,xxx	
untyped	Fred,lxa	
directory	Fred	

† A dead file is one that has been created but the contents of which are being updated. For example when NetFS copies a file to a file server it reserves space by creating a dead file before writing to it.

File contents

The contents of files are unchanged when transferring to UNIX, save for untyped files. These have their load and execute addresses appended to the file, making it 8 bytes longer:

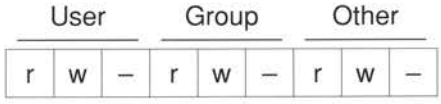


L0 is the least significant byte of the load address, L3 the most significant. Bytes E0 to E3 are the execute address.

Access attributes

When creating a new file or directory

You can use the system variable `NFS$CreateAccess` to define the default read/write access attributes for user, group and other that RISC OS NFS sets when creating a file or directory on UNIX. This variable uses six of its lowest nine bits:



You can set it in octal by using a leading '0' (you'll find this familiar if you've ever used the UNIX `chmod` command with numbers), or in hexadecimal by using a leading '0x', or in decimal by just using a number. So the following would all set the variable to specify user read/write access, group read only access, and no access to others:

```
*Set NFS$CreateAccess 0640          (using octal)
*Set NFS$CreateAccess 0x1A0         (using hexadecimal)
*Set NFS$CreateAccess 416           (using decimal)
```

You can override the value of the `NFS$CreateAccess` variable for a specific mount by setting a system variable `NFS$CreateAccess_`*`mountname`*.

You should set these access variables in a boot file; see your RISC OS 3 *User Guide* if you need help on this.

If a relevant access variable exists then files and directories are created with the read/write access it specifies. Files of type UNIX Ex also have their execute attributes set to be the same as the corresponding read bits in the variable.

If no relevant access variable exists then files are created with user read/write access, and with user execute permission if the files' type is UNIX Ex. Directories are created with user read, write and execute permission.

When mapped from RISC OS

When RISC OS NFS sets the access to a UNIX file using RISC OS attributes they are mapped as follows:

RISC OS bit	UNIX bit
owner read	user read
	user execute is also set if owner read is set and the file's type is UNIX Ex
owner write	user write
public read	group read and other read
	group execute and other execute are also both set if public read is set and the file's type is UNIX Ex.
public write	group write and other write
locked	(discarded)

Similarly, when RISC OS NFS sets the access to a UNIX directory using RISC OS attributes they are mapped as follows:

RISC OS bit	UNIX bit
owner read	ignored – i.e. user read is left unchanged
owner write	ignored – i.e. user write is left unchanged
	user execute is always set
public read	group read and other read
public write	group write and other read
locked	NOT group execute and NOT other execute

Dates

UNIX date stamps any files just as usual if you use RISC OS NFS to create or amend them.

Finding an object

When RISC OS NFS is finding an object it searches in this order, using the first match it makes:

1. It searches for an exact name match.
2. It searches for an exact name match after any RISC OS specific extension has first been removed.
3. It searches for a name match ignoring case, after any RISC OS specific extension has first been removed.

NFS – its file mapping from UNIX to RISC OS

This section describes how RISC OS NFS maps files from UNIX to RISC OS.

Filenames

File type extensions

The first change RISC OS NFS makes is to remove any filename extension used to store the RISC OS file type.

It starts by looking through the `extensions` file to see if the filename has an extension that matches one you specified; if so, the extension gets removed. You can in fact set up a different mapping for each direction of file transfer, so you can map many UNIX file extensions to single RISC OS file types. See section *Editing the extensions file* on page 56.

If RISC OS NFS can't find a matching filename extension in the `extensions` file it then tries to remove any of its own default extensions; so the following all appear as `Fred` under RISC OS:

UNIX name	Notes
Fred	
Fred,*hhh*	*hhh* is 3 lower-case hex digits
Fred,xxx	
Fred,lxa	

Truncation

The next thing RISC OS NFS does to a filename is to truncate it to the length set by the system variable `NFS$TruncateLength`. By default this is set to the value 10 – the same length as the maximum that the desktop Filers can handle. It only gets read once, when the NFS module is loaded.

If you want a different truncate length use the `*Set` command, say in a boot file:

```
*Set NFS$TruncateLength 12
```

If you're using NFS from the command line you may want to override filename truncation. To do so set the variable to a large number, e.g. 1000000.

Character translation

The final change RISC OS NFS makes to a filename is to translate the character '.' (the Acorn directory separator) to '/', for example:

UNIX name	RISC OS name
fred.c	fred/c
.profile	/profile

File contents

RISC OS NFS makes the last 8 bytes of any file with a '`,lxa`' extension invisible; this is to hide the load and execute addresses it presumes itself to have appended.

If you generate a file in UNIX with a '`,lxa`' extension which is less than 8 bytes long, you will get unpredictable behaviour if you try to manipulate it from RISC OS.

Access attributes

When the access attributes of a UNIX file or directory get translated by RISC OS NFS they are mapped as follows:

UNIX bit	RISC OS bit
user read	owner read
user write	owner write
user execute	(discarded)
group read	(discarded)
group write	(discarded)
group execute	(discarded)
other read	public read
other write	public write
other execute	(discarded for files) NOT locked for directories

Dates

RISC OS NFS always uses the UNIX last modified date stamp to map to a RISC OS date stamp. It assumes the UNIX date stamp to be in GMT, and uses the value set by `*TimeOffset` to convert this to local time. For details of `*TimeOffset` see page 73.

File types

RISC OS NFS resolves file types by looking for any filename extension used to store the RISC OS file type. It does so at the same time as it resolves filenames – see also the earlier section *Filenames* section on page 53.

It starts by looking through the `extensions` file to see if the filename has an extension that matches one you specified; if so, it sets the file to the corresponding file type. See section *Editing the extensions file* on page 56.

If RISC OS NFS can't find a matching filename extension in the `extensions` file it then sets the file type using its default file extensions. So, again taking the example of a file that will be displayed as `Fred`:

UNIX name	Notes	RISC OS type
Fred	UNIX directory	Directory
Fred	no execute bit is set	Text
Fred	any execute bit is set	UNIX Ex
Fred,*hhh*	*hhh* is 3 lower case hex digits	&*hhh*
Fred,xxx		dead
Fred,lxa		none, undated

See also *section Soft links* on page 56.

Load and execute addresses

If a UNIX file has the extension '`,lxa`' then RISC OS NFS assumes it to be a RISC OS untyped file that it created on UNIX. It uses the last 8 bytes of the file to give the load and execute addresses. So if they were:

the load address would be &01234567, and the execute address would be &89ABCDEF.

Soft links

RISC OS NFS resolves soft links up to eight times – that is, whilst following a soft link, it only allows eight soft links to be traversed. If this traversal reaches an existing object other than a soft link:

- the object's UNIX attributes and contents get used
- the soft link's UNIX name gets used to determine the RISC OS file type.

In other words soft links behave transparently except that, where there is more than one soft link to a file, its type may differ depending on which soft link you use to view it.

RISC OS NFS can't traverse a soft link that leaves a mount. If a UNIX link name starts with the character '/' then RISC OS NFS treats it as the root of its mount. Consequently absolute soft links will only work if you've mounted the UNIX root directory '/' and if the soft link does not leave the root filing system. For example, if you had mounted `/usr` then this UNIX soft link in the `/usr` directory would be traversed:

```
lrwxrwxrwx  1 root     wheel        11 Feb 23 17:19 man -> ./share/man
```

whereas this one wouldn't be:

```
lrwxrwxrwx  1 root     wheel        11 Feb 23 17:19 man -> /usr/share/man
```

We advise that when you make soft links on UNIX you always make relative links (i.e. start them with '`.`' or '`..`') rather than absolute ones.

If a soft link does not resolve to an existing non-soft-link object within eight expansions it's displayed as a file with type 'SoftLink' (&FDC). You can't do anything from RISC OS with one of these dead soft links.

Other object types

Block and character special files and named sockets are displayed as UNIX Ex files. Fiddle with these from RISC OS at your peril!

Editing the extensions file

The `extensions` file is held in the `files` subdirectory of the Internet application, and configures the mapping of RISC OS file types to UNIX filename extensions. To add your own filename extensions for specific RISC OS file types you need to edit this file:

1 Load Edit onto the icon bar – if it's not already loaded.

2 Open the directory display containing the `!Internet` application – if it's not already open.

3 Open the `!Internet` application directory by holding down the Shift key while you double-click on the `!Internet` icon.

4 Open the `files` directory.

5 Load it into Edit by dragging its icon to the Edit icon on the icon bar.

6 Add to the file your own mappings of file type to UNIX file extension.

- There are two sets of mappings: one for files coming from RISC OS (starting immediately beneath the 'From extensions:' line), another for files returning to RISC OS (starting immediately beneath the 'To extensions:' line).
- The general syntax is:

  `RISC_OS_file_type new_extension [anything]`

  The RISC OS file type can be the name of a file type, or its file type number in hexadecimal. So to give Data files (type &FFD) the extension '`.dat`' you could use either of these lines:

  ```
  Data .dat
  ffd .dat
  ```

- If you add a third field ('*`anything`*' above) then the extension becomes 'sticky'.

  When moving to UNIX the extension is only added if it's not already present. So if the line were to read:

  ```
  ffd .dat sticky
  ```

  the Data file `output` would be renamed `output.dat`, whereas the Data file `output.dat` would not be renamed.

  When returning from UNIX the extension doesn't get removed; otherwise it's handled the same as ever, so the file type gets set using this extension.

We expect you'll want the 'To extensions:' part to duplicate the entries in the 'From extensions' part, so any extension that gets added when a file is transferred to UNIX gets removed again if the file returns to RISC OS. However, there may be a lot of UNIX extensions that you wish to convert to a single RISC OS file type. For example, you may have several UNIX applications each of which generates text files with different extensions – say '`.txt`', '`.doc`' and '`-asc`'. To do so, just add extra entries to the 'To extensions', thus:

```
Text .txt
Text .doc
Text -asc
```

7 When you've added all the extensions you want to, save the edited `extensions` file, overwriting the old one.

Appendices

Appendix A: Installing Internet

In the earlier chapter *Introduction to* TCP/IP on page 23 we outlined different ways in which you could set up OmniClient. You'll have to edit some configuration files to do so. This chapter tells you how to make those changes and install the software. It assumes you are using the desktop, and are familiar with simple use of it. If you have any problems refer to the RISC OS *User Guide* supplied with your RISC OS computer.

Configuration files

The configuration files that you need to edit are held within the Internet Application. Rather than refer them all the time by their lengthy full pathnames, we'll just use the leafname. The files are supplied on the Network distribution disc as:

Filename	Leafname
`$.!Internet.files.hosts`	`hosts`
`$.!Internet.files.extensions`	`extensions`
`$.!Internet.files.mimemap`	`mimemap`
`$.!Internet.files.networks`	`networks`
`$.!Internet.files.protocols`	`protocols`
`$.!Internet.files.services`	`services`

The hosts, networks, protocols and services are collectively known as the *databases*.

What the files do

The hosts file gives the host names and the Internet addresses of all the computers you wish to refer to by their hostname.

The extensions file allows 'from' extensions to be added when converting from RISC OS to Host, and 'to' extensions to be removed when converting to RISC OS from Host.

The mimemap file maps the each content type to (a) particular file type(s).

The networks, protocols and services files contain databases of network, protocol and service names. These files are unused by the TCP/IP Protocol Suite (Release 2), and are provided to support any extra software that uses TCP/IP protocols.

The files specify the default values normally used on all computers that support the TCP/IP protocols; consequently, you shouldn't ever need to edit them. If you do, you should see respectively the UNIX networks, protocols, and services manual pages.

There are also more configuration files which can be modified and they are as follows:

Filename	Leafname
$.!Boot.Choices.Internet.routes	routes
$.!Boot.Choices.Internet.startup	startup
$.!Boot.Choices.Internet.user	user

What the files do

Any extra *Route commands, apart from the default route that can be set using !InetConf can be put into this file.

This file is generated by InetSetup which includes the configurations set by the user. It shouldn't be edited by hand unless you know what you're doing.

Any special configuration options that cannot be set using !InetConf can be put into this file. It is executed immediately after the automatically generated Startup file.

InetSetup Configuration

InetSetup does most of the configuration of the software. It sets the principal host name of a computer. It configures each interface setting their Internet addresses, their netmasks, the driver modules to be used and establishes the routing information. It also defines where to find the configuration files for the *databases*. Finally, it sets whether or not the station will forward packets between multiple interfaces, and hence whether it acts as a TCP/IP gateway.

Different ways to configure the software

There are some decisions you have to make on how to configure the software. This section outlines what those decisions are. For details of how to configure the choices you make, see *Installing the* TCP/IP *Protocol within* OmniClient on page 64, and the instructions and examples in the files you'll need to edit.

Ways to set the Internet address of each interface

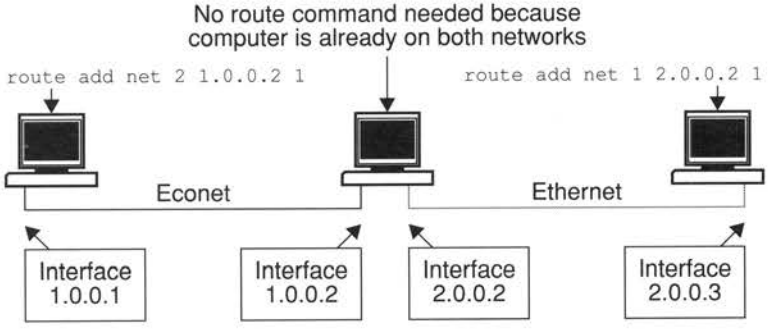
You can set the Internet address of each interface in three ways:

- you can set it explicitly on the computer to which it's fitted
- you can use the interface name to look up the Internet address in the `hosts` database
- you can use the physical address to look up the Internet address using *Reverse* ARP exchanges with an ARP *server*.

Ways to establish routing information

If you are using gateways, you need to ensure that each RISC OS computer knows their location, and the route to subnets other than the one to which it is connected. There are three ways you can do this:

- Use the `*Route` command to explicitly define the routing of each gateway. For example:



Installing the TCP/IP Protocol within OmniClient

Backing up the distribution discs

In setting up stations you'll need to alter the software. **You must not alter the distribution discs themselves**; always work on copies of them, made either on media that you can access from other computers (i.e. an Acorn file server) or that you can take to other computers (i.e. a floppy disc):

- If your RISC OS computers can access an Acorn file server, make a directory on the file server called (say) `TCP_IP`, and open its directory display.

  Put the *Network* distribution disc in the floppy drive, open its directory display, select all the files and drag them to the `TCP_IP` directory display. Dismount the floppy disc. Repeat this process for the *Applications* distribution disc.
- Alternatively, use the **Backup** option from the floppy disc drive's icon bar menu to back up the two distribution discs to another pair of floppy discs.

Where you can install the software

When you actually install the software for a particular station, you can do so on any media you like, such as a hard disc, floppy discs or a remote file server. All that is important is that you preserve the directory structure that is on the distribution discs – things that are in the same directory must stay together.

- If you're installing the software on a hard disc or an Acorn file server, we suggest you make a directory called (say) `TCP_IP` to hold the software. This directory can be anywhere you like – it needn't be in the root directory.
- You don't have to do this, and may prefer to put the software in a directory that already contains other applications.

Configuring the software

Run !InetSetup and the required configuration changes can simply be made by following the necessary menus and entering the details as required:

- AUN
- Access
- Internet

Edit the Hosts file

- If you need to edit the hosts file this can be done through the Internet-Host names option and then double-clicking on the Hosts file.

- Edit the file so that it contains the Internet addresses and host names of all the RISC OS computers you wish to refer to by host name. (If a computer's host name isn't in here you'll instead have to use its Internet address whenever you want to communicate).
- If you want to add any UNIX hosts to the file, their addresses and names must match those in your UNIX /etc/hosts files. For a small number of hosts you'll probably find it best just to print out and copy a UNIX hosts file. For a larger number of *hosts*, you might find it better to copy across your UNIX hosts file later.

If you need help, see the comments in the supplied hosts file, and the UNIX hosts manual page.

After you have finished editing the hosts file and the required updates have been made you simply save the file so the changes can take effect.

Further Reading

The following variables are used to configure the software but are for further reading purposes only:

Variable	Notes
`Inet$EaCount`	This sets the number of Ethernet cards present
`Inet$Email`	This sets the email address associated with the machine. Inet$EtherDevice This sets the driver module to be used to interface the TCP/IP software with your Ethernet interface.
`Inet$EtherIPAddr`	This sets whether your station has an Ethernet interface that you wish to configure for use with TCP/IP, and (if so) sets the Internet address, or specifies that it be looked up using Reverse ARP.
`Inet$EtherIPMas`	This sets the netmask for an Ethernet interface; the default is correct for most cases.
`Inet$EtherType`	This sets the type of Ethernet device used.
`Inet$HostName`	This sets a station's principal hot name
`Inet$IsGateway`	This sets whether the machine is to forward IP packets, and hence whether it will act as a gateway.

`Inet$LocalAddr`	This sets the local address of the machine.
`Inet$LocalDomain`	This sets the local domain of the machine.
`Inet$MimeMappings`	This sets the mapping of content types to file types.
`Inet$Path`	This sets the pathname of the directory containing the Internet application.
`Inet$Resolvers`	This sets the internet resolver module IP addresses.
`Inet$RouteDOptions`	This sets whether RouteD – the Internet routing module – is run and (if so) with what options.
`Inet$Started`	This sets whether the Internet application has started or not.
`Inet$Startup`	This sets the pathname for the startup file.
`InetDBase$Path`	This sets the pathname of the directory containing the database files.
`InetLogs$Path`	This sets the pathname of a directory containing the logs file.
`InetSetup$Dir`	This setss the directory of the inetsetup file.
`InetSpool$Path`	This sets tha pathname of the directory containing the spool file.

Appendix B: NFS star commands

*Free

Displays free space on a mount

Syntax

```
*Free [mountname]
```

Parameters

mountname	the name of an NFS mount point

Use

*Free displays your available free space on a mount, the total free space on that mount, and the size of the mounted filesystem. If the mounted filesystem is using quotas, *Free shows your free and used space on it, the quota beneath which you are expected to remain, and the absolute limit of usage that you cannot exceed.

If no mountname is given, the current one is used.

Example

```
*Free tp1usr
No quota on tp1usr
Space on filing system:
Free 69 889K
Available 33 168K
Size 367 218K

*Free home
Bytes free     9 626K
Bytes used       374K
Bytes quota   10 000K
Bytes limit   12 000K
```

Related commands

None

*Logon

Sets the name server to use and/or authenticates a user/password pair

Syntax

```
*Logon [-Host hostname] [username[[CR]password]]
```

Parameters

hostname	the host to be used as the name server for this and subsequent name requests
username	the user whose details are to be authenticated on the name server
password	the user's password, which – if omitted – you will be prompted for

Use

*Logon sets the name server to use and/or authenticates a user/password pair, depending on the parameters passed:

- if a *hostname* is given, then it sets the name server to use for this and subsequent authentication using *Logon
- if a *username* and *password* are given, they are passed to the current name server for authentication; they are also used to access any mount points you create in the future
- if no parameters are given, then the current user and name server are displayed.

Examples

`*Logon -Host tp1`	*Sets tp1 as the current name server.*
`*Logon mhardy`	*Prompts for the password, which is reflected as dashes; authenticates the pair using the current name server (ie tp1); if the password is valid, sets the current user to mhardy.*
`*Logon`	*Displays the current user (mhardy) and the current name server (tp1).*

Related commands

*Mount (page 69)

*Mount

Lists or mounts NFS mount points

Syntax

```
*Mount [-Host hostname] [mountname [mountpath]]
```

Parameters

hostname	the host to be used as the NFS server for this and subsequent mount requests
mountname	the name by which RISC OS will refer to this mount point
mountpath	the directory on the NFS server which shall act as $ on NFS::*mountname*

Use

*Mount lists or mounts NFS mount points, depending on the parameters passed:

- if a *hostname* is given, then it sets the host as the NFS server to be used for this and any subsequent mount requests; it also lists all mount points on the host, giving the mount name, mount path and user name for each mount point
- if a *mountname* and *mountpath* are given, then the *mountpath* is mounted, using the last user name that was successfully authenticated with *Logon, and the URD, CSD, PSD and Library are all set to $ (i.e. the *mountpath*)
- if no *mountpath* is given, but a *mountname* is given for which a mount already exists, then the URD, CSD, PSD and Library are all reset to $ for that mount
- if no parameters are given, then all current mount points are listed, showing the mount name, mount path and user name for each mount point.

The files accessed via a mount point are always accessed using the details of the user which was current at the time the *Mount command was executed.

Examples

`*Mount -Host tp1 tp1usr /usr`	*Mounts the directory /usr on the host tp1, giving it the mount name tp1usr.*
`*Mount tp1usr`	*Resets the* URD, CSD, PSD *and Library for the mount named tp1usr to* $ *on that mount*
`*Mount -Host tp1`	*Lists all mount points on tp1.*
`*Mount`	*Lists all mount points.*

Related commands

*Logon (page 68)

*NFS

Selects NFS as the current filing system

Syntax

```
*NFS
```

Use

*NFS selects NFS as the filing system for subsequent operations. Remember that it is not necessary to switch filing systems if you use the full pathnames of objects. For example, you can refer to ADFS objects when NFS is the current filing system.

Example

```
*NFS
```

Related commands

*ADFS, *Net, *RAM, *ResourceFS, *SCSI

*NFSInfo

Displays NFS module internal statistics

Syntax

```
*NFSInfo
```

Use

*NFSInfo displays detailed information about NFS module activity, including known hosts, users, URDs, CSDs, PSDs and Libraries, and details of its cache performance.

Most of the information displayed is runic in nature. It is presented mainly as an aid to trouble-shooting, should you require it.

Example

```
*NFSInfo
Host list:
name=<tp1>, usage = 1
name=<tp2>, usage = 1
name=<tp3>, usage = 2

Cache hits 90051; Cache misses 36385

User list:
name=<mhardy>, uid=<1234>, gid=<27>, usage = 1
name=<nobody>, uid=<32767>, gid=<9999>, usage = 1
CSD: 0x1844144, <>
PSD: 0x1844144, <>
URD: 0x1844144, <>
Library: 0x1844144, <>
```

Related commands

None

*TimeOffset

Sets the time offset in minutes from GMT to local time

Syntax

```
*TimeOffset minutes
```

Parameters

minutes	minutes offset from GMT to local time

Use

*TimeOffset sets the time offset in minutes from GMT to local time. RISC OS NFS adds this offset when it interprets the server's date stamps, which it assumes to be in GMT. The offset can be negative or positive.

Example

```
*TimeOffset 60
```

Related commands

None

Appendix C: LanMan star commands

*Connect

Sets up a connection to a shared resource

Syntax

```
*Connect name server dir-name [username password]
```

Parameters

name	mount name
server	file server name
dir-name	directory path
username	the user whose details are to be authenticated on the server
password	the user's password

Use

*Connect sets up a connection to a shared resource.

Examples

```
*Connect Home OmniClient1 fbloggs Fredbloggs Atlas
```

Related commands

*Disconnect (page 78)

*Configure LanManTransport

Selects transport protocol to be used by Lan Manager

Syntax

```
*Configure LanManTransport IP|NETBEUI
```

Parameters

`IP`	Internet Protocol
`NETBEUI`	Basic Lan Manager protocol

*Configure FS

Configures the security controller for a set of resources

Syntax

```
*Configure FS server_name|domain_name
```

Parameters

server_name	the name of the server controlling security on the network
domain_name	the name of the domain where security is controlled on the network

*Disconnect

Disconnects from a shared resource

Syntax

```
*Disconnect name
```

Parameters

name	mount name

Use

*Disconnect disconnects from a shared resource.

Examples

```
*Disconnect Home
```

Related commands

*Connect (page 75)

*LanMan

Selects Lan Manager as the current filing system

Syntax

```
*LanMan
```

Use

*LanMan selects Lan Manager as the current filing system.

Example

```
*LanMan
```

Related commands

None

*LMinfo

Displays debugging information

Syntax

```
*LMinfo
```

Use

*LMinfo displays information on current mounts, and debugging information including details of previous errors.

Examples

```
*LMinfo
```

Related commands

None

*LMLogoff

Logs a user off a workgroup or domain

Syntax

```
*LMLogoff
```

Use

*LMLogoff logs a user off a workgroup or domain.

Examples

```
*LMLogoff
```

Related commands

*LMLogon (page 82)

*LMLogon

Allows a user to log on as part of a workgroup or domain. It attempts to authenticate the user with the domain controller, and connects to the user's home directory (if defined).

Syntax

```
*LMLogon workgroup username password
```

Parameters

workgroup	name of workgroup
username	the user whose details are to be authenticated on the server
password	the user's password

Use

*LMLogon allows a user to log on as part of a workgroup or domain.

Examples

```
*LMLogon History fbloggs squiffy
```

Related commands

*LMLogoff (page 81)

*LMNameMode

Sets the way LanManFS capitalises file and directory names

Syntax

```
*LMNameMode 0|1|2
```

Use

*LMNameMode sets the way LanManFS capitalises file and directory names:

0	sets all capitals
1	sets lower case
2	intended to preserve case (including a mixture of cases). Currently capitalises initial letter.

Examples

```
*LMNameMode 0
```

Related commands

None

*LMprinters

Adds a known server and list of printers

Syntax

```
*LMprinters server printer_name(s)
```

Parameters

server	file server name
printer_name	printer name

Use

*LMprinters adds a known server and list of printers.

Examples

```
*LMprinters omniclient printer_green
```

Related commands

None

*LMserver

Adds a known server and list of mount paths

Syntax

```
*LMserver server mount_path(s)
```

Parameters

server	file server name
mount_path	mount path name

Use

*LMserver adds a known server and list of mount paths to the list of network servers.

Examples

```
*LMserver omni2 public
```

Related commands

None

Appendix D: OmniClient and discless workstations

Introduction

Currently, OmniClient must be loaded from some form of secondary storage medium, such as a disc: this poses the question of how to load it into network stations that don't have a local disc.

This is not difficult to do, but does require care. There are two principle cases to consider, depending upon whether the target machine has a DCI 2 or a DCI 4 network protocol stack available from its onboard ROM.

All of the following instructions assume that you have correctly upgraded your network's shared copy of !System, using !SysMerge as supplied with this release.

As mentioned elsewhere, it's a simple matter to determine whether your machines have DCI 2 or DCI 4 software: at the command line, type the command

```
help intern.
```

(Note the full stop.)

- If the machines responds with a line including a reference to a module `InternetA`, or to an `Internet` module with a version of `2.xx` (where `xx` is anything), then you have a DCI 2 protocol stack loaded.
- If the response refers to `Internet`, with a version of 4.03 or higher, then you have a DCI 4 stack loaded.

DCI 4 stack

If your machine has a DCI 4 stack the procedure is comparatively trivial, since there is no requirement to replace the running network stack.

There are a few simple methods which can be used to load OmniClient depending on your system configuration. You can either use Acorn Access+, or using the Applications Accelerator supplied with Acorn Level 4 Release 3, or if you have Acorn Access cards with LanManager protocols installed, you may load OmniClient from an NT server (but check with your supplier if in any doubt as to which protocols are available on your Ethernet card).

It is possible to use other methods, such as loading directly from a Level 4 fileserver, although this is not recommended for performance reasons. Another way is to use one of the available third-party application loading mechanisms. In

such cases contact the software publisher if it is not obvious how to proceed (you may well find that the following instructions provide sufficient information to enable you to set this up, however).

If you don't have Acorn Access+

- load the Applications Accelerator from your fileserver (usually, you would do this anyway during the client stations' bootstrapping procedure: if you don't already have a suitable procedure set up, then either contact your local network support agent or Customer Services for advice on how to do this).
- Run !OmniClient from a read-only mount, either shared from your Access+ 'server' station, or from your Applications Accelerator server.

If you do have Access+

If you do have Access+, you will find that the function of the Access 'discs' icon is taken over by OmniClient (assuming that you've selected 'Access' as one of the OmniClient protocols in OmniSetup): if you are using the Applications Accelerator, you will find that the 'discs' icon is removed from your icon bar, but that you do not have available the Acorn Access functions within OmniClient. This is because OmniClient is designed to operate with full versions of Access only, not with the read-only version that forms the basis of the Applications Accelerator. This does not pose any problems, since the actual ShareFS filing system is still available, and any command line access to it will still work (such as takes place when OmniClient loads its component files). Indeed this can be viewed as a positive advantage, in that it further simplifies a user's interaction with the network – it's one less fileserver type to worry about.

If you have LanManager

If you are using LanManager and have Access+ with OmniClient you will need to ensure that your machine is configured correctly in order to boot OmniClient directly from an NT server, or equivalent. In order to do this, you will need to make the following configuration changes to your machine from the command line:

Star command	Action
`*Configure Boot`	This command sets the configured boot action so that a poweron, reset or Ctrl Break runs a boot file. Depending on the operating system version being used, this option may or may not need to be changed.
`*Configure FileSystem LanMan`	This sets LanMan as the configured filesystem.

*Configure LMTransport IP | NetBEUI	Configures the filesystem to the IP or NetBEUI protocol, depending on your network configuration.
*Configure FS *name*	This sets the domain name (or fileserver) from which LanManager will attempt to boot.

Each of these changes will take effect on the next power-on or hard reset.

DCI 2 stack

If your machine has a DCI 2 stack you should ideally consider upgrading the software on your network cards, as otherwise new network software will have to be loaded via the network. Potentially the connection will be broken in the middle of this process, while the DCI 4 protocol stack is loaded.

There is, however, a simple way to address this, and that is to use the Applications Accelerator supplied with Acorn Level 4 Release 3. Note that it is important to use this version of the Applications Accelerator, as it will operate correctly with the existing DCI 2 stack, rather than the more up-to-date DCI 4 version (which as its name suggests, requires DCI 4, which we don't yet have!).

We also assume here that you have the correct DCI 4 driver module for the cards that you are using: if you don't have Acorn-supplied cards, you will need to contact the card manufacturer to obtain a suitable driver (some third-party drivers are supplied in our standard distribution, but not all).

There are several possibilities here, covering differing combinations of network software already in use in the Acorn environment: we will consider those relating to Acorn-supplied software in common use. The example files listed below are supplied on the support disc.

AUN/Level 4 Server alone

This assumes a network using a Level 4 fileserver, but not using the Applications Accelerator for resource and application delivery.

In this case, we advise that you use the Applications Accelerator: if you have more than a couple of machines, then it's quicker to have them load the Applications Accelerator client software from the Level 4 server, and then use this to load the rest of the components required for OmniClient, than it is to try loading these

items directly from the Level 4 server. The sequence is also much simpler, due to the 'connectionless' nature of the Access protocols used by the Applications Accelerator.

AUN/Level 4 and Acorn Access / Applications Accelerator

This assumes a network based around a Level 4 fileserver, used in conjunction with either Acorn Access or the Applications Accelerator for applications and resource delivery.

We assume that all your network cards are of the same type – that is, they require the same driver: if this is not the case, you either need an appropriately-configured copy of !Bootnet for each type, or you need to take equivalent action.

We will also assume that your client machine boot sequence already loads the Applications Accelerator client software – if it doesn't, then as suggested in the DCI 4 section above, contact your local network support agent, or Customer Services, for advice on how to improve your boot sequence to make use of this mechanism. In fact, it's quite simple to run this software, and then issue a Filer_Run command to transfer control of the bootstrap sequence to a !ShareBoot application, located on a shared resources disc.

The following sequence of commands should be used as a basis for your !ShareBoot application, to replace the running DCI 2 protocol stack with a new DCI 4 stack, and then to install OmniClient:

!ShareBoot.!Run

```
Set ShareBoot$Dir <Obey$Dir>
Desktop Obey -c <ShareBoot$Dir>.!Deskstart
```

!ShareBoot.!DeskStart

```
ChangeDynamicArea -RamFsSize 700K
Copy <ShareBoot$Dir>.!Bootnet RAM:!Bootnet ~V ~C R
CDir RAM:Modules
CDir RAM:Modules.Network
Copy System:Modules.Network RAM:$.Modules.Network ~V ~C R
Set Sys$Temp <System$Path>
Set System$Path RAM:$. Run RAM:!Bootnet
Set System$Path <Sys$Temp>
UnSet Sys$Temp
Wipe RAM::RamDisc0.$.* ~c~vfr
RMKill RamFS
```

```
ChangeDynamicArea -RamFsSize 0
Filer_Run <ShareBoot$Dir>.!Omni
... remainder of boot file
```

This assumes that a suitably-configured !Bootnet and !Omni are both stored within your !ShareBoot application: you may choose to hold them elsewhere; if you do so, then simply modify the paths given as appropriate to reflect your chosen location for these items.

Acorn Access only

This assumes a network using Acorn Access in conjunction with a fileserver which requires the use of OmniClient; for example an Xemplar SchoolServer.

As with any Access boot sequence, you need to set up a 'shared disc' to act as your boot server. Configure your clients to Boot, from the filesystem ShareFS, and save this disc as the first (leftmost) mount from the icon bar menu.

This is essentially the same sequence as above, with the important exception that !Bootnet does not get run – it is only necessary to load and initialise the required modules, which can be done as follows, assuming the use of an AEH70 network card:

!ShareBoot.!Run

```
Set ShareBoot$Dir <Obey$Dir>
desktop obey -c <ShareBoot$Dir>.!Deskstart
```

!ShareBoot.!DeskStart

```
ChangeDynamicArea -RamFsSize 700K
CDir RAM:Modules
CDir RAM:Modules.Network
Copy System:Modules.Network.Internet RAM:$ ~V ~C
Copy System:Modules.Network.AUNmsgs RAM:$ ~V ~C
Copy System:Modules.Network.Ether3-16 RAM:$ ~V ~C
Copy System:Modules.Network.MManager RAM:$ ~V ~C
RMKill ShareFS
RMKill Freeway
RMKill InternetA
RMKill AccMsgs
RMKill Ether3 (or whatever is appropriate for your cards)
RMLoad RAM:AUNMsgs
RMLoad RAM:MManager
RMLoad RAM:Ether3-16 (or whatever is appropriate for your cards)
RMLoad RAM:Internet
RMReInit Freeway
```

```
RMReInit ShareFS
Wipe RAM::RamDisc0.$.* ~c~vfr
RMKill RamFS
ChangeDynamicArea -RamFsSize 0
Filer_Run <ShareBoot$Dir>.!Omni
... remainder of boot file
```

This mechanism is somewhat more efficient than the simpler copy operation used in the AUN case, though it gives the appearance of being more complex.

Index

Symbols

A

C

D

E

F

G

H

I

L

M

N

O

P

Q

R

S

T

U

Reader's Comment Form

Acorn OmniClient
Part number 0484,952

We would greatly appreciate your comments about this Manual, which will be taken into account for the next issue:

Did you find the information you wanted?

Do you like the way the information is presented?

General comments:

If there is not enough room for your comments, please continue overleaf

How would you classify your experience with computers?

☐ **First-time user** ☐ **Used computers before** ☐ **Experienced User** ☐ **Programmer**

Cut out (or photocopy) and post to:

Dept RC, Technical Publications
Acorn Computers Limited
Acorn House, Vision Park
Histon, Cambridge CB4 4AE
England

Your name and address:

This information will only be used to get in touch with you in case we wish to explore your comments further

Notes

Notes

Notes


Acorn