The WROCC Guide to Networking

Part 1

£3.00

Contents

An Introduction to Network Addresses

October 2009

by Steve Fryatt – stevef@wrocc.org.uk

September's meeting saw us take a whistle-stop tour through the subject of networking and afterwards, we promised – perhaps foolishly – to look at it in more depth through the pages of the newsletter over the coming months. I'll start this time with the basics of network structure and addresses; later, we plan to move on to the other things, such as file transfer, that can be done with a network.

Separating the parts

At its simplest, a network is a means of connecting a collection or computers – or devices which contain computers – together so that they can share data and resources. Along with basic desktop PCs, all kinds of other devices can be connected: printers, file-storage units, ADSL and cable modems for internet access, MP3 players. The list is almost endless.

Networking hardware comes in two main flavours: wired and wireless. Most of us are probably familiar – at least by sight – with wired networks: Cat 5 cables ending in RJ45 connectors. This is what you will get if you order a modern network interface card for a RISC OS machine; the older co-axial leads with the round BNC sockets are obsolete these days. The other option is wireless: laptops, routers, mobile phones and PDAs often have this included.

However, once it's set up and working, the complexity of how all the bits of hardware talk to each other becomes invisible to the computers using the network. It doesn't matter whether your network uses Cat 5 cable, wireless links, mains-borne HomePlug connections or even any combination of the three – once the connections have been made, the devices on the network can all be treated and configured in the same way.

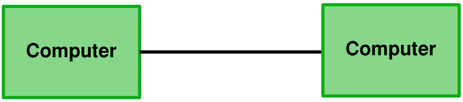


Figure 1: A very simple network

In some ways, RISC OS users have it easy. Our machines tend to come with wired network ports and no wireless option, so connecting things together is simple as long as enough Cat 5 cable is available. As a result, I won't be mentioning the kind of connections much in this article – we may come back to look at things like wireless links in a later newsletter.

A simple network

The simplest network consists of two machines, connected by a single cable (as shown in figure 1 above). Modern network cards may be able to do this with a standard network cable; older ones will probably require something known as a 'crossover cable'. The two machines can talk to each other, but that's it.

To add anything else, we need an additional piece of hardware known as a 'switch' or a 'hub' – switches are better than hubs (for

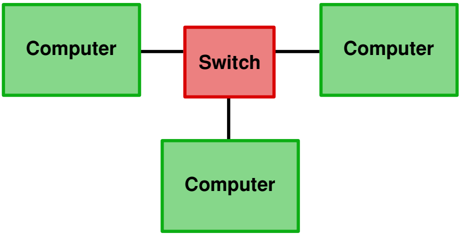


Figure 2: A network with some more devices

largely technical reasons that won't worry most home users), and these days you will have to search very hard to find anyone still selling the latter. Adding a third computer is simply a case of connecting each one to a central switch, using standard Cat 5 network cables: this is shown in figure 2 opposite.

As long as you don't need to connect to the internet, or anyone else's networks, then this setup is all that is required for a simple home or office use. All of the computers can see all of the others, and as long as suitable software is installed, they can share files, printers and more besides. However, we need to be able to identify which computer is which, so the next step is to give each of them a unique 'tag' – an address.

Addressing the issue

On all but the simplest of networks, all the devices will have a couple of different addresses: an IP address and a MAC address. MAC addresses are unique to a device, and belong to that device from the time of its manufacture; IP addresses, on the other hand, will still be unique for every device connected to the network, but the address a device has may change many times in its life, depending on where and when it is connected.

To use an analogy, a MAC address is a bit like a person's name: it uniquely identifies the person in question, and they will often keep their name from birth. An IP address, meanwhile, is like their home address: it uniquely identifies where they can be found at any given point in time, but is subject to change if they move house.

MAC addresses are 48 bits long, and chunks of addresses are allocated by the IEEE to manufacturers of network hardware. For human convenience, they are usually written down as six pairs of hexadecimal digits, and often appear as 12-34-56-78-90-ab or 12:34:56:78:90:ab. On a RISC OS machine, the MAC address of the network interface can be found using the `*ifconfig -a` command.

Due to their 'permanent' nature, MAC addresses are often used by network hardware to identify actual devices on the network. An example of this would be a wireless access point only allowing access to devices whose MAC addresses have been added to a list of 'allowed' machines. If you configure your wireless router to recognize only your laptop's MAC address, for example, then your neighbour can't connect using their computer.

The right protocol

When data is being routed around a network, however, it's IP addresses – not MAC addresses – that are used to arrange delivery. Rather like postcodes in conventional mail there's a logic to the numbers, and hardware in the network knows how to use this to direct the information along the correct wires.

Rather like MAC addresses, IP addresses are just a binary number – only they are 32 bits long instead of 48 bits (although this isn't always true – see the box below). For human consumption, they are usually split into blocks

Not all IPs are equal

The internet is currently in a state of upheaval, although you would be forgiven for not having noticed. Traditional IP addresses (those of the form 192.168.0.1, and known as 'IPv4') are a scarce resource, and – depending upon who you believe – will soon run out. The solution is to switch to a new system known as 'IPv6': an address system using 128 bits instead of 32.

IPv6 addresses are written as eight groups of four hexadecimal numbers, such as 2001:db8:85a3:0:0:8a2e:370:7334, and are much more of a mouthful. RISC OS does not support the new system, of course, but at present global uptake is very low anyway.

At least for the time being, RISC OS users can safely assume that all the IP addresses they encounter will be IPv4.

of 8 bits and written as four numbers: for example 192.168.0.7 or 10.1.0.203.

To ensure that data gets delivered to the correct computer, every machine on a network must have a unique IP address allocated to it. This only applies to things that can have data delivered to them – things like switches and hubs, which merely pass packets through on their way somewhere else, don't need one.

In order to do anything useful with your network, you will need to allocate each machine on it an IP address. Just like MAC addresses, these are given out by a central body – in this case the IANA; fortunately, there are several blocks of addresses allocated for use on 'private networks'. The most common of these are in the range 192.168.*x*.*x* and 10.*x*.*x*.*x*.

There are a number of different ways that addresses can be given to devices. Until recently, RISC OS systems had to have the addresses set manually by the user; now, depending on what else is on the network, the job can be done automatically. We'll come back to this next month.

Sub-networks

In order to route data around the network, and on to other networks, IP addresses are split into two parts: one identifies the network, while the other specifies individual machines on that network. The number of bits either side of the divide varies depending on the network concerned, and its location is given by something known as the 'subnet mask'.

Subnet (or just 'net') masks are usually written in the same style as IP addresses, and the most common one on home networks is '255.255.255.0'. Viewed in binary, the bits of the mask that are set indicate bits in the IP address which refer to the network; those which are unset show the bits that identify the individual machines. So with this net mask, the address 192.168.0.7 points to machine 7 on network 192.168.0. Figure 3 below shows this more graphically.

As home networks normally use the netmask 255.255.255.0, all the devices connected to them must use addresses with the same first three sets of numbers. These need to fall into those ranges allocated by IANA to private networks, but beyond that it's up to you (using the same network that your broadband router's default is in, if you have one, is a good idea).

Since IP addresses and netmasks often need to be given together, there are a number of different forms available to write them down. The long-winded approach is to write each out in full: '192.168.0.7/255.255.255.0'. This is clearly a mouthful, so it's common to see the shorter version: '192.168.0.7/24' – the '/24' is a way of showing that the first 24 bits of the netmask are set.

Finally, there are two special machine numbers to be aware of: that with all its bits unset, and that with all its bits unset. These should not be used for devices on the network, as they have special meanings; in the network above, these would be 192.168.0.0 and 192.168.0.255. This gives a useful range of machine numbers from 1 to 254 on the average home network.

Adress:	192	168	0	7
	11000000	10101000	00000000	00001011
Netmask:	11111111	11111111	11111111	00000000
	255	255	255	0
	Network			**Machine**

Figure 3: The make-up of an IP address and netmask

Escaping the net

On a simple, self-contained network consisting of a number of machines connected by one or more hubs or switches, all of the IP addresses must be on the same network number, and all of the netmasks must be the same. If they are not, then things will very quickly go wrong.

In order for two machines on two networks using different netmasks or network numbers in their IP addresses to communicate, a piece of hardware called a ‘router’ is required to sit between them. This has two network connections, one on each network, and it passes packets of data back and forth. A router could easily be a computer with two network cards and the right software installed – each card having a different address.

If you have ADSL or cable internet and it connects to your computer via the network port, then your broadband modem will also be a router: it will be transferring packets between your home network and that which your ISP is operating. However, because home networks will be using one of the private address ranges, it will also be performing a function called Network Address Translation, or NAT.

Without NAT, if machine 192.168.0.2 in figure 4 contacted 87.204.78.92, then each would know the address of the other. Unfortunately, since addresses from the public ranges will be in use all around the world, there could be many millions of 192.168.0.2s connected to the internet at any one time with no way to tell them apart.

So that packets return to the right place, the broadband router uses NAT to remove all references to the private address (that is 192.168.0.2) before sending the data on to 87.204.78.92, making it look as if the information really came from the router itself – the router’s address of 87.204.78.219 isn’t from a private range, and so *will* be unique on the internet. If a reply comes back from 87.204.78.92, the router remembers which machine on its private network sent the original request, and forwards the information on.

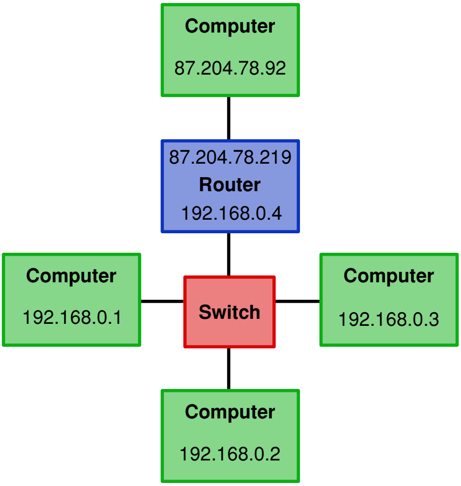


Figure 4: Linking two networks with a router

How to Select and Set IP Addresses

November 2009

by Steve Fryatt – stevef@wrocc.org.uk

Last month, we started this series on networking by looking at the basic structure of a network and at the addresses which are used to identify the different machines connected to it. It's all very well knowing what the addresses mean, however, but when setting up a network for real it's also necessary to know what addresses to use and how to let the hardware know what they are.

Private networks

All the network devices that you buy will already have MAC addresses set inside them: as we saw last month, these are allocated by the IEEE via the manufacturers, and uniquely identify each and every piece of networkable hardware in existence. Therefore, when putting your network together, all that's left is to decide what IP addresses to use.

Like MAC addresses, IP addresses are allocated by a central body: in this case the IANA. Fortunately, there are three blocks of addresses allocated for 'private networks' which are free to be used without registration, so long as they do not appear on the internet at large. For anyone setting up a stand-alone network (with no internet connection at all) or an internet connection using a router which performs Network Address Translation (which is basically any broadband internet connection), these are the ones to use.

There are two blocks of addresses in common usage: those starting with 10 (10.*x*.*x*.*x*) and those starting with 192.168 (192.168.*x*.*x*). There is also a third block, in the range 172.16.*x*.*x* through to 172.31.*x*.*x* – these seem to be a lot less common, and you may have to search hard to find anyone using them. Home networks will generally be using a subnet mask of 255.255.255.0, meaning that the first three numbers of the address will give the network number and therefore be the same on all machines – so how do we go about choosing a network number?

Look at your hardware

If you have a broadband router, the best strategy for deciding on a network number is to find the address that it defaults to on the local network, and take the same one. So if your router defaults to 192.168.0.1 (as many Netgear devices do, for example), then use 192.168.0.*x* as your network; if it defaults to 10.1.0.100, then use 10.1.0.*x* as your network. This means that should your router require resetting for whatever reason, it will still be accessible to any computers on your network which have a fixed IP address set (we'll come back to this later).

If you share an internet connection from a Windows machine using Windows' Internet Connection Sharing, then this always uses an address of 192.168.0.1 for its connection on to the local network. Since this can not be changed, you *must* use a network of 192.168.0.*x* when accessing the internet via Windows ICS.

If you don't have a broadband router, then you're free to choose whatever network number you wish. However, it's probably wise to stick with one of the common ones: 192.168.0.*x* or 10.1.0.*x*. If you ever expect to use a Windows machine on the network, then 192.168.0.*x* is probably a wise choice to give compatibility with ICS in the future.

As for the subnet mask, stick with 255.255.255.0 and limit yourself to 254 devices on the network. Enough software defaults to assuming this setting that trying to change it is more hassle than it's worth.

Address allocation

Having decided on a network number, the next step is to decide what addresses to give each device and then to configure each one. There are two ways to do this: manually, or automatically. Manual configuration is tedious, but it does give you overall control and some devices require it. Automatic configuration is much less effort, but it can have limitations and not all versions of RISC OS support it.

To configure a network manually, it's simply necessary to decide how to allocate machine numbers to the different devices, and then to configure each machine in turn – remembering not to use either of the special machine numbers mentioned last month. If you're methodical and keep a written record of what you have done, the process isn't too hard.

Alternatively, the most common method of automatic address configuration is a system called the Dynamic Host Configuration Protocol, or DHCP for short. A single device on your network acts as the 'DHCP server', and when they connect, other devices contact it and ask for an IP address to use. Most broadband routers offer this facility, as does Windows ICS.

Using DHCP has many advantages. For portable computers such as laptops it means that the computer can pick up an address for any network it is connected to without manually changing its configuration: it's safe to assume that any 'public' network, such as WiFi hotspots, will have DHCP available. For other devices, such as networked printers, which can only be configured over the network, it gets around the chicken and egg problem of how to set the IP address if you can't get to the configuration tools before the IP address is set.

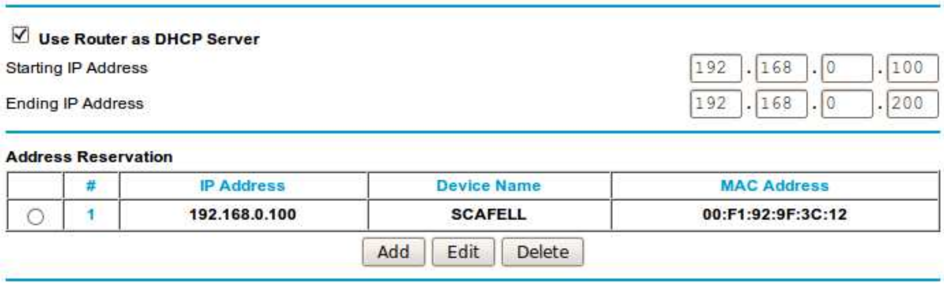
Pick and mix

There are two perceived problems with DHCP. The first is that older versions of RISC OS don't support it. It's in RISC OS 5 and 6, and it was added to RISC OS 4 in the second Select release, but users of versions prior to this can only set their IP addresses manually (or via an AUN system which isn't much use when accessing the internet).

The second problem is that for devices which need to be accessed across the network (such as printers, network-attached storage, or even computers acting as different types of server), it is desirable for them to keep the same IP address every time they are switched on. Often the addresses of devices such as printers get stored in numerous places (such as the Printers configuration on RISC OS), so updating these should the DHCP give them a different address could quickly become a nuisance.

Fortunately, in most cases, it's possible to have both DHCP and manually configured addresses on the same network. Routers can be told to allocate ('lease' in the DHCP jargon) addresses from a limited range, so if this is configured as (say) 192.168.0.100 to 192.168.0.254 then it's also possible to give other devices fixed addresses so long as they fall in the range 192.168.0.1 to 192.168.0.99.

The other trick, which some routers support, is to tell them to always give the same address to a particular device – as explained last month, this is done by MAC address. If your laptop is



☑ Use Router as DHCP Server

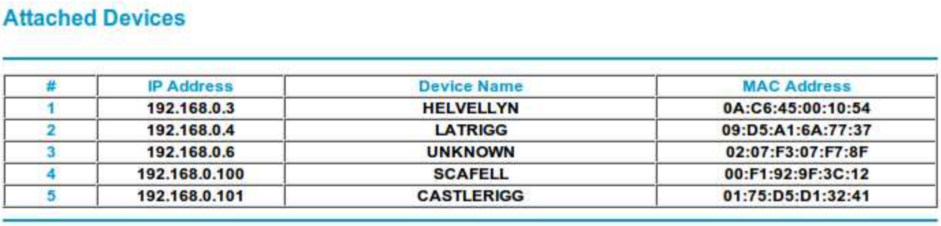
Starting IP Address 192 . 168 . 0 . 100

Ending IP Address 192 . 168 . 0 . 200

Address Reservation

	#	IP Address	Device Name	MAC Address
○	1	192.168.0.100	SCAFELL	00:F1:92:9F:3C:12

The DHCP configuration from a Netgear broadband ADSL router, showing that it is able to lease addresses from 192.168.0.100 to 198.162.0.200. In addition, one IP address in that range (192.168.0.100) has been reserved for a device with the MAC address 00:f1:92:9f:3c:12.



Attached Devices

#	IP Address	Device Name	MAC Address
1	192.168.0.3	HELVELLYN	0A:C6:45:00:10:54
2	192.168.0.4	LATRIGG	09:D5:A1:6A:77:37
3	192.168.0.6	UNKNOWN	02:07:F3:07:F7:8F
4	192.168.0.100	SCAFELL	00:F1:92:9F:3C:12
5	192.168.0.101	CASTLERIGG	01:75:D5:D1:32:41

The Attached Devices list from the same Netgear ADSL router, showing how to find the IP addresses which have been allocated by the DHCP. Note that 'Scafell' turns up with the correct address of 192.168.0.100 – as reserved in the screenshot on the previous page. The 192.168.0.101 address has also been allocated by DHCP, while the others have been statically set on each bit of hardware.

configured to use DHCP so that it can be used on different networks, you can still tell your router to reserve one particular address from it's DHCP range for it so that it always appears in the same place. A similar approach can be used to ensure that devices such as printers always appear at the same address when allocated their address via DHCP.

What's that in the ointment?

If your router doesn't have the ability to reserve IP addresses in this way, all is not lost. DHCP can still be a useful way to override the default addresses set in things like printers and network -attached storage units so that they can be seen on the local network. Once this has happened, they can be accessed using the temporary address long enough to be re-configured to use a fixed IP address.

Those using Windows ICS have another problem to deal with. While it offers DHCP, ICS does not appear to have a means of limiting the range of addresses that it offers to

Configuring RISC OS

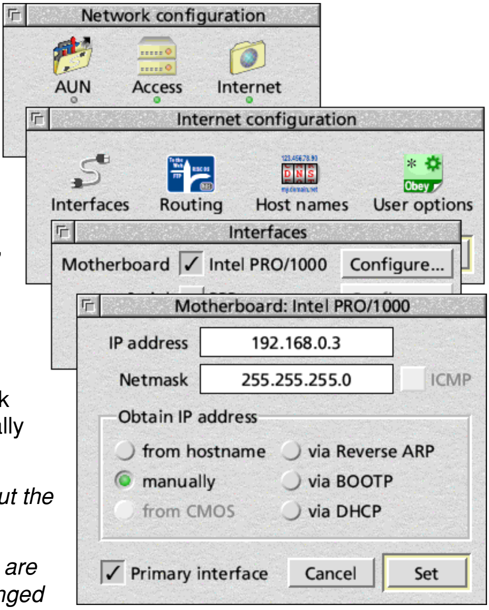
On native RISC OS systems, the IP address is set from within the Network configuration section of Configure (double-click on ***!Boot***, or select *Configure...* from the Task Manager's iconbar menu). Since each network interface can have its own address, the options are set from the Interfaces section.

Inside Network Configuration, select *Interfaces*, then click on *Configure...* beside the entry for the network interface which is to have its address set.

To configure the address manually, select *manually* and enter the IP address and netmask into the fields. To set the addresses automatically using DHCP, select *via DHCP*.

Note that this screenshot is from RISC OS 5, but the options are very similar in RISC OS 4 and 6.

On Virtual Acorn systems, the network settings are picked up from the host and should not be changed in RISC OS – see Configuring Windows opposite.



Motherboard ☑ Intel PRO/1000 Configure...

Motherboard: Intel PRO/1000

IP address 192.168.0.3

Netmask 255.255.255.0 □ ICMP

Obtain IP address

○ from hostname ○ via Reverse ARP

● manually ○ via BOOTP

○ from CMOS ○ via DHCP

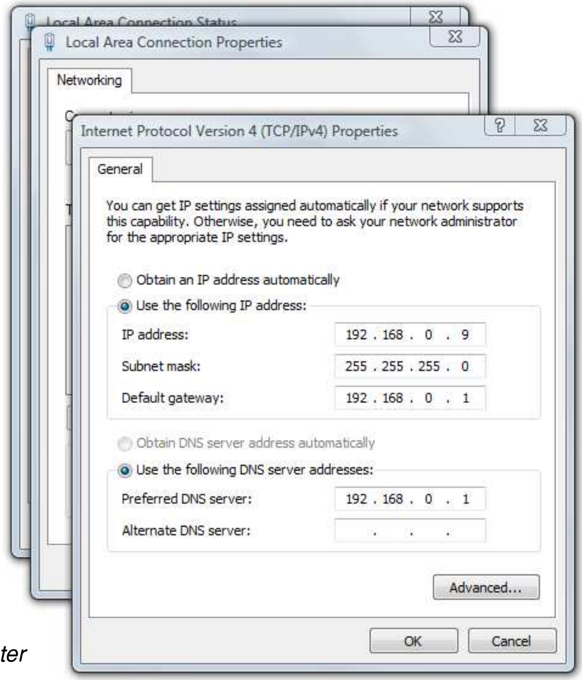
☑ Primary interface

lease. As such, it starts with 192.168.0.2 (remember that ICS always gives itself 192.168.0.1), and will happily work through to 192.168.0.254 if you connect enough things to the network.

If two devices on the network get the same address, one of them will disconnect until the clash is resolved. As such, users of ICS (in particular – clashing IP addresses is a problem that can affect any network if mistakes are made) need to watch out of this. In practice, given that most home networks will be used with few devices, it's possible to work around the limitation by using high numbers for fixed IP addresses and keeping an eye out for problems.

Configuring Windows

As with RISC OS, Windows configures IP addresses individually for each network interface. However, it can be quite difficult to get to the configuration options.



Internet Protocol Version 4 (TCP/IPv4) Properties

You can get IP settings assigned automatically if your network supports this capability. Otherwise, you need to ask your network administrator for the appropriate IP settings.

○ Obtain an IP address automatically

● Use the following IP address:

IP address: 192 . 168 . 0 . 9

Subnet mask: 255 . 255 . 255 . 0

Default gateway: 192 . 168 . 0 . 1

○ Obtain DNS server address automatically

● Use the following DNS server addresses:

Preferred DNS server: 192 . 168 . 0 . 1

Alternate DNS server: . . .

Local Area Connection Status
Local Area Connection Properties
Networking
General
Advanced...
OK
Cancel

Go to the Control Panel (from the Start Menu). On Vista, select *Network and Internet*, then *Network and Sharing Center* and finally pick *Manage Network Connections* under Tasks in the left margin. On XP, go to *Network and Internet Connections* in Control Panel, then select *Network Connections* from the left margin. If you use 'Classic View' for the Control Panel on either system, then simply select *Network Connections* to get to the same place more directly. On Windows 7, select *View network Status and tasks*, then *Change adapter settings* from the margin.

The Network Connections window contains an entry for each network interface on the computer: double-click on the one corresponding to your network card. This opens a Connection Status dialogue, at the bottom of which is a *Properties* button – click on this, and the Connection Properties dialogue will open on top.

In the centre of this new dialogue will be a list of items 'used' by the connection. Select the entry for 'Internet Protocol Version 4' (or just 'Internet Protocol' on Windows XP), then click on the *Properties* button beneath the list to open the Internet Protocol Properties dialogue (pictured).

To use DHCP on the connection, simply select *Obtain an IP address automatically*. Alternatively, selecting *Use the following IP address* will allow a static address and subnet mask to be entered in the appropriate fields.

Note that this screenshot was taken on Windows Vista – similar dialogues exist on Windows XP and Windows 7, although there is no reference to IPv4 or IPv6 in XP.

Looking Up Domain Names

December 2009

by Steve Fryatt – stevef@wrocc.org.uk

Over the past couple of months, this short series has looked at the way that network addresses can be set up on a home network which includes RISC OS systems. We have seen that each machine needs to be given an IP address, and that these addresses can then be used to transfer data between any two points on the network.

Easy to remember

In the first article I mentioned that similar addresses are used across the internet: all the computers attached to the network are identified by an IP address, and – with the exception of networks protected by routers performing Network Address Translation (NAT) – each one has its own address which is completely unique.

When we visit websites, we usually type in addresses like `www.bbc.co.uk` or `www.wrocc.org.uk` – not least because they are easy to remember. Before your browser can visit the site and fetch the pages it contains, however, it needs to turn those textual names into numerical IP addresses. You could just as easily type `http://212.58.253.67/` into NetSurf, and you would still end up on the BBC website (or at least on the server that powers it).

The process of converting textual addresses (or 'domains') into IP addresses is known as 'name resolution', and on RISC OS it is handled by the Resolver (although some applications use their own implementation, supplied via libraries such as UnixLib).

Addresses on the internet

As already noted, when NetSurf is faced by a domain such as `www.wrocc.org.uk`, or POPstar has to access `smtp.orpheusnet.co.uk`, they will make use of a resolver. If the application chooses to use the central RISC OS Resolver module it can access it via a set of SWI calls; otherwise it will be using one built into itself; similar options exist on other systems.

Either way, the first thing that the resolver will do is to check whether the address it has been given is one that it already knows about: that is, one that it has been asked to resolve 'recently'. If POPstar fetches mail every ten minutes from `smtp.orpheusnet.co.uk`, for example, then the first time in a session it will need to look it up but on subsequent visits it *may* be able to remember the IP address from the last time.

The reason for that 'may' is that while a domain will always map to the same IP address for simple servers, this isn't always the case for larger sites. Places like Google, for example, will have many servers located around the world, and the address that `www.google.com` resolves to may change on a regular basis as demand varies or sites go offline for maintenance. To allow for this, when the resolver gets details of the address belonging to a domain it will also be told how long it can remember it for.

If the resolver doesn't know the details of the address itself, the next step is to check local lists of addresses in something called the 'hosts file'. If a match for the domain is found here, then the associated IP address will be used. This can be a useful tool, and we'll return to it later in the article.

A giant internet directory

If the resolver can't find details of the address in its memory or the hosts file, the final step is to look up the domain in the Domain Name System (or DNS). This is a directory which contains details of every possible domain (the bit of an internet address following the initial 'http://' and up to the first '/') – as you might expect, it contains a lot of information.

To make the system manageable and robust, it is divided up into a strict hierarchy of servers, each of which look after parts of the domain name. At the top of the tree are a set of servers known as the 'root', which deal with spitting the system up into groups based on the Top

Level Domain (or TLD). Domains ending .com will be handled separately from those ending .org; those ending .uk will be handled separately again, and so on. The system is looked after by the IANA along with ICANN.

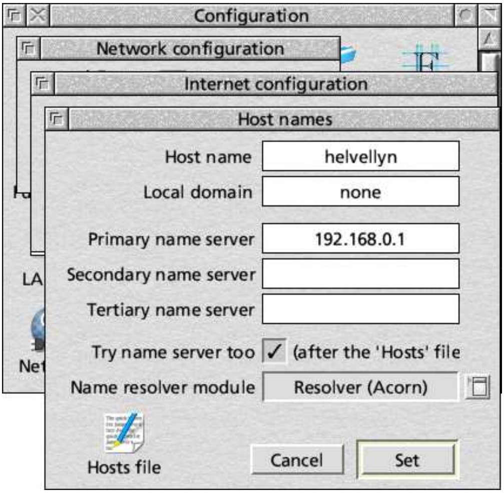
At the next level down the tree are servers that deal with the addresses belonging to each TLD. In the UK, all addresses ending .uk will be handed by a collection of DNS servers managed by Nominet – the body who also handle the registration of most .uk domain names. For small TLDs this may be all that is required, but generally there will be further levels below this: in the UK, for example, Nominet break the domains down into groups such as .co.uk, .org.uk, .me.uk and so on; they handle most of them, but some have been delegated to third parties (such as .ac.uk and .gov.uk, which are administered by JANET).

At the bottom of the tree, each domain has a DNS server that knows its details. For the club site at `www.wrocc.org.uk`, for example, this is handled by the DNS servers at our hosts, Purley Hosting. Every request to look up the IP address of our site will end up here, although just like the resolver, other servers in between will remember the details after the first request for as long as they are allowed to do so to ease the load on Purley Hosting's server.

Using the DNS

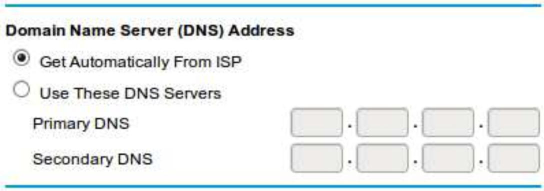
To use the DNS on our computers, we need to point the resolver to one or more DNS servers. On RISC OS, details are set up in either the *Resolver* section of *Network* or the *Host names* section of *Network – Internet Configuration* in Configure, depending on what version of the OS you have installed. Three fields here allow primary, secondary and tertiary servers to be set up: the idea is that if the resolver can't get an answer from the first, it will try the second and then as a last resort the third.

Most ISPs will supply two DNS servers which act as a gateway between their subscribers' machines and the rest of the DNS system: the IP addresses will usually be given in their sign-up information (this is one place where using



Host name: helvellyn
Local domain: none
Primary name server: 192.168.0.1
Secondary name server:
Tertiary name server:
Try name server too ✓ (after the 'Hosts' file
Name resolver module: Resolver (Acorn)

The RISC OS 5 host names configuration. The name of the machine is 'helvellyn', at 'none', and all DNS requests go to the router at 192.168.0.1.



Domain Name Server (DNS) Address

● Get Automatically From ISP

○ Use These DNS Servers

Primary DNS . . .

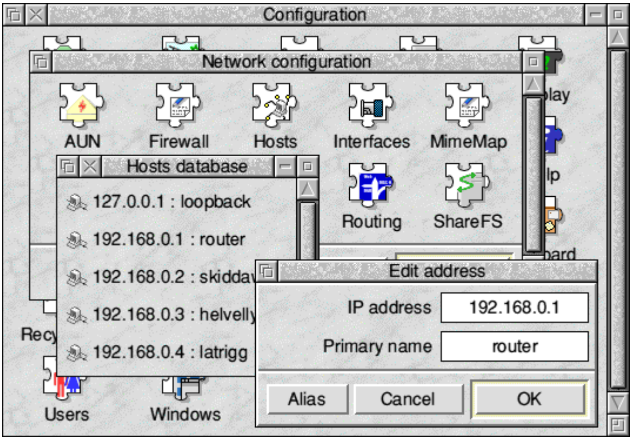
Secondary DNS . . .

The DNS configuration from a Netgear router, set to pick up details of the ISP's DNS servers automatically

textual domain names isn't possible). Generally it isn't necessary to have a third, tertiary server specified – if you wish to do so, then there are some free services available. Take care here – a malicious DNS server can direct you to whatever internet addresses it chooses, so make sure the ones you choose are legitimate.

If you access the internet via a broadband router, then it will usually receive details of your ISP's DNS servers automatically as part of the connection process. In this case, it is often best to set the primary server to be the local address of your router and leave the other two fields blank: this way the router handles all DNS requests and takes care of any problems should your ISP change the addresses of its servers. Since the mainstream operating systems can also collect this information automatically, many ISPs don't bother to notify their subscribers of such changes – leaving us RISC OS users scratching our heads when the resolver stops working.

The host names dialogue also contains fields to set the machine's own name and local domain. If you set up names for the machines on your network as described below, then the host name should be the same as the one given here. The local domain doesn't really matter: if you don't know better, using 'home', 'invalid' or something similar will be OK.



127.0.0.1 : loopback

192.168.0.1 : router

192.168.0.2 : skidda

192.168.0.3 : helvell

192.168.0.4 : latrigg

IP address: 192.168.0.1

Primary name: router

Configuring the Hosts file on RISC OS 6 – users of RISC OS 4.02 or 5 can access the file via the Host names dialogue seen opposite

Local settings

Earlier I mentioned that the resolver will look for a local hosts file before going to the DNS servers that have been configured. This is a simple text file stored on the machine, and contains a list of domains and the IP addresses that they relate to. It can be used to give names to addresses on the local network, as well as to override entries in the public DNS.

On RISC OS, the hosts file lives at ***InetDBase:Hosts***, which is usually located inside the !Internet resource within !Boot. Fortunately its exact location doesn't usually matter, as it can be accessed from Configure.

If you have RISC OS 6 or a recent version of RISC OS 4, then opening Configure and going to *Network – Hosts* will access the hosts database. This will normally contain a single entry for loopback, which should be left alone; new entries can be added by selecting *New host...* from the menu and entering an IP address and name. More than one name can be given to the same machine using *Alias*: the entry for 127.0.0.1 has two additional names by default.

On other versions of RISC OS, go to the *Network* section of Configure, then to *Internet – Host names*. On RISC OS 5 there is a text file icon at the bottom with the name *Hosts file*, which can be double-clicked to open Hosts into a text editor; the same dialogue on RISC OS 4.02 has a *Hosts file...* button which does the same thing.

If you have to edit the file manually, the format is fairly simple. It consists of a series of entries, one per line; lines starting with '#' are comments and get ignored. Each entry consists of an IP address followed by some spaces or tabs, and a domain name. For example, if the following lines were in a hosts file:

```
192.168.0.1  router
192.168.0.2  skiddaw
192.168.0.3  helvellyn
192.168.0.4  latrigg
```

then typing `http://latrigg` would have the same effect as `http://192.168.0.4` to a browser.

Changing the internet

Entries in the hosts file don't have to be on the local network, however. Assuming that the details above about the BBC's website always hold true, we could add an entry reading

```
212.58.253.67  www.bbc.co.uk
```

and then all attempts to look up `www.bbc.co.uk` would be handed without having to ask the DNS for help. In most cases this would not be a good idea, as there is no guarantee that the BBC will never change the IP address of their website. However, if you wanted to block all access to the BBC's site from the computer in question, an entry of

```
127.0.0.1  www.bbc.co.uk
```

could be added to Hosts and then all attempts to access the site would simply try and connect to a local webserver (which would fail unless something like WebJames was installed).

Any questions?

This brings us to the end of our look at network addresses: armed with the information in these three articles, it should be possible to get a simple home network up and running, and ensure that the machines on it can talk to each other and to the outside world. With these basics in place we can do more interesting and useful things with the network: fetch emails, visit websites, share files between different computers, and much else besides.

Accessing the Internet

January 2010

by Steve Fryatt – stevef@wrocc.org.uk

So far in this series, we have seen how to set up a simple home network and get the machines talking to each other. The articles have covered giving the machines addresses and configuring them so that they can look up the names of servers and websites – these articles will still be available from the website for another month or so if you missed any of them.

With these essential basics out of the way, we can move on to look at actually doing something useful with the network. For anyone with a broadband connection, one central requirement is likely to be the ability to access the internet connection from any machine – fortunately this is also one of the easiest to achieve.

Internet connections

Back in the first article, in the October issue, we saw how an ADSL router acts as a gateway between the machines on the local network and the internet at large. If you use such a router, then accessing the internet from local machines is simply a case of telling each machine where on the network the router (or gateway) can be found.

If you use DHCP, then this information is provided automatically as part of the details the router sends back to each machine when it requests an IP address. Otherwise, on RISC OS you need to go into Configure, then access *Network – Internet – Routing* and enter the router's local IP address into the Gateway field. The other options in the dialogue box should be left unticked (as shown on the right).

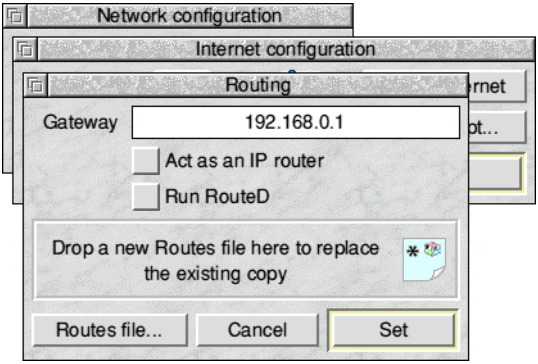
On Windows systems, the Default gateway field can be found in the same dialogue as that which sets the machine's IP address – see the November issue for details.

Using a router as a gateway on a local network where the IP addresses are taken from one of the 'private' ranges requires that it can support Network Address Translation (or NAT) – we looked at this in the October issue. Fortunately, it's extremely unlikely that any broadband router will not do this these days: you may need to watch out for some early devices that only came with a single network port, though.

Sharing a modem

If you have a simple modem – ADSL or dial-up – which is connected directly to a computer, then sharing the connection is more difficult. RISC OS cannot do this at all (as it does not support NAT), but if you have the modem attached to a Windows box then Internet Connection Sharing (ICS) can be used.

To configure ICS, follow the instructions in the November issue to get to the Network Connection Properties dialogue for the connection relating to the modem (not the local network). On the *Advanced* tab, tick the option to *Allow other network users to connect through this computer's internet connection* and the connection will be available on the local network. No further action is necessary, although November's article covered a number of pitfalls to be aware of when using ICS on a home network.



Gateway: 192.168.0.1
□ Act as an IP router
□ Run RouteD

Drop a new Routes file here to replace the existing copy

Configuring the gateway under RISC OS 4.02. The options are very similar in later versions of the OS, including RISC OS 5, Select and RISC OS 6.

Setting up ShareFS

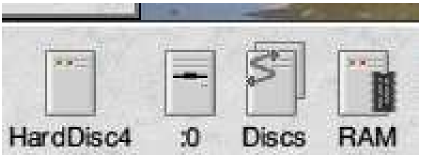
March 2010

by Chris Hughes – chris@wrocc.org.uk

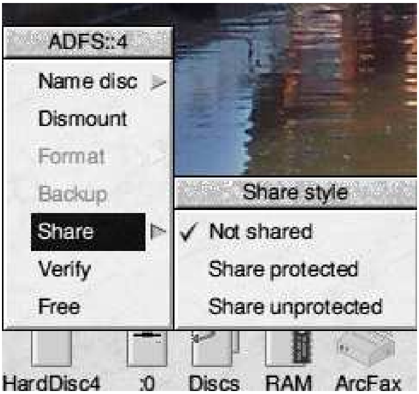
ShareFS (Share Filing System) is a network filing system for sharing files between RISC OS computers, regardless of whether they are on native hardware or in virtual form on a PC or Mac.

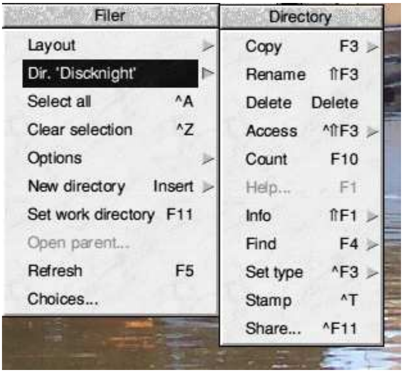
How do we use it?

In its simplest form, then depending on which version of the OS you have, you either have to load the ShareFS module or at least turn it on (to make it active) in Configure. Double-click on !Boot to open Configure, select either *Network – Access* or *Network – ShareFS* (depending on your version of RISC OS) and ensure that *Enable Access* or *Enable ShareFS* is ticked. Once turned on, a new Discs icon appears on your iconbar.



Once enabled, then for simple sharing you can go to your hard disc icon on the iconbar, click Menu and look at *Share*, and you then will see three options. This is a way to share an entire hard drive, CD, RAM disc or floppy drive, but it will not work on Virtual RiscPC: sharing is done differently on VRPC, and I will explain the alternative method in a minute.

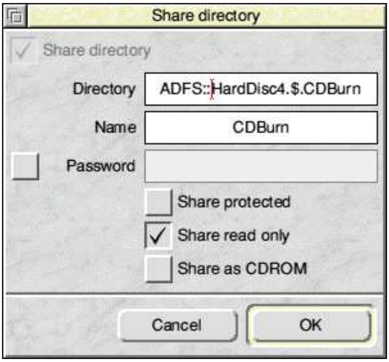




Another way to share things is just to share a specific folder. This is done by going to the folder you wish to share, clicking Menu over its icon and selecting *Dir. 'foldername' – Share...* as in the example shown here. Unlike the method for whole discs, this *does* work in VRPC.

Once you have clicked on the *Share...* option a new window opens (shown at the top of the opposite page), and you can decide to share the folder as 'read only' or 'protected'. Read only does what you would expect, while protected implements the public read and write permissions for the files in the share (see *Selection – Access* – in the filer's menu), with a couple of catches: it seems that some versions of ShareFS treat *either* read *or* write being set as allowing read *and* write access. In addition, the folder being shared as protected must also have public access itself, otherwise the share will seem to work but you might get the cryptic 'file not found' error when trying to access it from another machine.

You will also notice you have the option to change the name of the share: in this example from CDBurn to something else. This can be useful to help more easily identify what the particular folder is when looking at a list of shares on another machine (note it does not change the normal folder name on your hard drive: it's an alias name).



Share directory

☑ Share directory

Directory: ADFS::HardDisc4.$.CDBurn

Name: CDBurn

□ Password:

□ Share protected

☑ Share read only

□ Share as CDROM

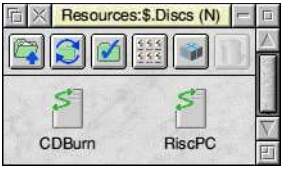
Cancel | OK

Sharing a directory from the desktop

Finding the shares

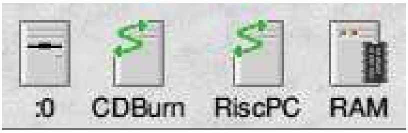
Once you have designated a folder or hard disc as shared, you should be able to see the shared items on the other computer or in VRPC (as an aside, you can also see them on the originating machine as well). If you click Select on the Discs icon on your iconbar, a new window opens offering you the shares that have been made available as shown below.

As you can see it has the CDBurn folder showing and another one called RiscPC (this is my alias for HardDisc4, as explained later). As some of you will remember from our networking talk, setting aliases for discs can be useful to prevent confusion, since each RISC OS machine has a HardDisc4 or similar by default and if both machines have shared HardDisc4 it can get confusing as to which one you are looking at.



In order to actually see the contents of a particular share or shares, you need double click on them from the window: when you do this, the grey Discs icon changes to a set of icons with the names of the shares underneath. If you now click on these icons they open a ShareFS filer window in the normal way, but you are now sharing the data between your computers.

Selecting shares from the Discs icon is fine for one-off sessions, but if you will regularly need



Sharing at the Command Line

The dialogue for sharing folders was added by RISCOS Ltd as part of the Select scheme, and so features in Adjust and RISC OS 6. Users of other versions of the OS need not despair, however, even if they don't have a copy of AccessSet – the `*Share` command is always present, and offers all of the options provided by ShareFS.

Along with using `*Share` in an Obey file to give whole discs an alias name (as described on the next page), the command can also be used to share particular folders when the share directory dialogue isn't available. It can be used in a similar way to when sharing discs: simply give the full pathname of the directory to be shared:

```
*Share ADFS::HardDisc4.$.CDBurn CDBurn
```

This example would share the CDBurn folder, with the same effect as the method shown earlier. The same options can also be specified: the 'read only' option can be added like this:

```
*Share ADFS::HardDisc4.$.CDBurn CDBurn -readonly
```

In a similar way, `-protected` and `-cdrom` emulate the other tick boxes.

The `*Share` command offers another option, which isn't in the dialogue: `-noicon` stops the share showing up in Discs or on the iconbar of other machines, although it can still be accessed by its pathname:

```
*Filer_OpenDir Share::CDBurn.$
```

to access the same folders or drives then you can save your 'mounts' (as they are called) so that they always appear in the iconbar. Click Menu over any of the ShareFS iconbar icons and select *Save Choices* or *Save Mounts*.

Virtual systems and ShareFS

Because Virtual RPC uses HostFS rather then ADFS for its standard filing system, you are really using the underlying Windows or Mac OS filing system. As a result, you can't share whole drives as ShareFS mounts from the iconbar in the same way, but you can share individual folders as already described.

Virtual Acorn have supplied a program with each copy of VRPC called AccessSet. It is normally located in the default Networking folder on the HostFS drive with a new installation. To run it, just double click on it as normal: this then opens a window like the one shown in the example below.

To use the program, you can simply drag and drop a folder on to the window: the path will change to that folder and you can then select your choice of options. If you wish to share the whole drive, just edit the path to something like HostFS::HardDisc4.$ – this would share all of your Hard drive in VRPC with another RISC OS machine or another copy of VRPC.

Folders can be selected for editing from all those set up with AccessSet from the pop-up menu to the right of the folder path.

On Virtual Acorn systems, AccessSet is used to configure access shares to work with other RISC OS machines.

Preventing duplicate names

As you may by now have realised, you could share two or more hard drives called HardDisc4 from different computers – this will cause confusion when selecting mounts in the Discs window. There is a way to prevent this by means of 'alias' names for the shared drives, which can be set in a similar way to the aliases for shared directories. In one example earlier, I had given my RiscPC's HardDisc4 drive an alias of RiscPC so I knew I was looking at my RiscPC's hard drive.

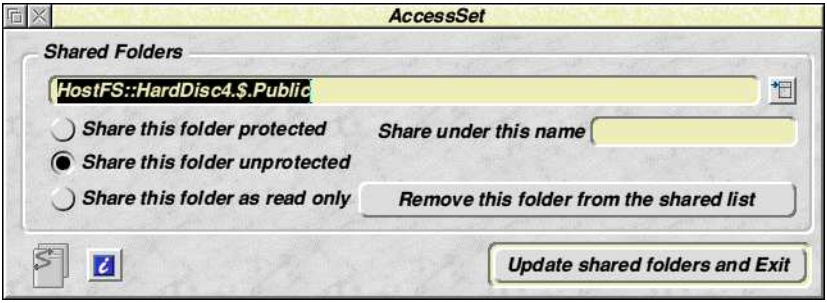
Aliases can't be set up for whole discs from their iconbar menus, but can be done from the command line. I did this by creating a simple Obey file using Edit (you could use any text editor, like Zap or StrongED if you prefer) with the following command:

```
Share ADFS::HardDisc4.$ RiscPC
```

Broken down this means Share the disc at "path to hard drive use" with an alias of "RiscPC" for this share.

You can have multiple entries in the Obey file: just put more Share commands on subsequent lines. The file should be saved in the Boot Tasks folder (I use a file name of ***AccessSet***): on RISC OS 6 this is at ***!Boot.Choices.Users.Single.Boot.Tasks***. On older versions of the OS, this might be at ***!Boot.Choices.Boot.Tasks*** (and if you use different users on your machine, 'Single' might be their username).

On Virtual RPC you can use the AccessSet program instead, and just fill in the *Share under this name* field to give your share an alias.



AccessSet

Shared Folders

HostFS::HardDisc4.$.Public

○ Share this folder protected

● Share this folder unprotected

○ Share this folder as read only

Share under this name

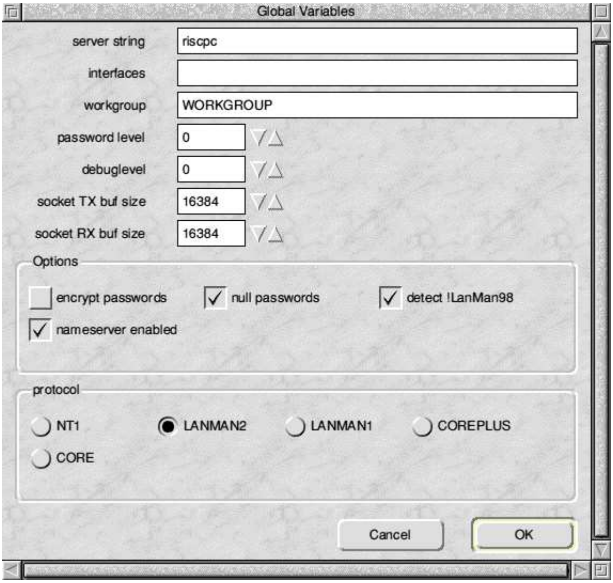
Remove this folder from the shared list
Update shared folders and Exit

Sharing with Samba

April 2010

by Chris Hughes – chris@wrocc.org.uk

What is Samba? Server Message Block (or SMB, or Samba) is a protocol for allowing computers to share data between them. RISC OS has some Samba Server software, thus allowing other computers like Windows-based ones to access folders, files and printers on RISC OS computers.



Global Variables

server string: riscpc
interfaces:
workgroup: WORKGROUP
password level: 0
debuglevel: 0
socket TX buf size: 16384
socket RX buf size: 16384

Options
☐ encrypt passwords ☑ null passwords ☑ detect !LanMan98
☑ nameserver enabled

protocol
○ NT1 ● LANMAN2 ○ LANMAN1 ○ COREPLUS
○ CORE

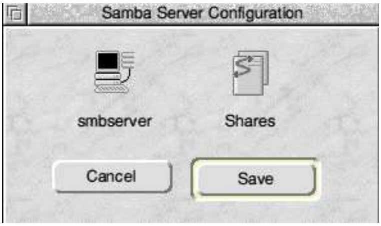
Setting the main server options in SMBServer

SMBServer, as it is called, is available from `riscossmbserver.sourceforge.net/downloads.html` You will see two versions there; the examples below are based on the 0.07a one.

Installation and use

So download a copy and install it. When you double-click on the application it puts an icon called ‘Samba’ on the left hand side of the iconbar. If you then click Menu over the icon you get a menu with a *Configure...* option: choosing this brings up the window shown here.

The configuration contains two choices. *SMBServer* is where you set up the actual server, while *Shares* is similar to ShareFS in that it defines what folders on your RISC OS machine can be accessed from other computers.



Set up the server

Looking at *SMBServer* first, these settings control the server – such as how other machines find it and connect to it. The *Server string* is the name that you give to your server: it is used to connect from other machines. *Interfaces* should be left blank, whilst *Workgroup* should be the same as your PC’s workgroup (either ‘MSHOME’ or ‘WORKGROUP’ being the most common default, depending on the version of Windows OS you have).

Leave the *Password level*, *Debug level*, *Socket TX buf size* and *Socket RX buf size* fields as they are.

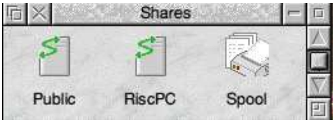
In the *Options* section I suggest using the same options as shown in the screen shot above: *Encrypt passwords* off, and the other three options on. In the *Protocol* section, try *LANMAN2* or if that fails, *LANMAN1*.

With the options set, click on *OK*: that sets the basic Samba server up. Ensure that you remember to click on *Save* in the Samba Server Configuration window before moving on.

Share your folders

Now that the server is set up, you have to decide what folders and printers you are going to let other computers have access to on your RISC OS computer. To see the default shares, either click Select over the Samba icon on the iconbar or open the Samba Server Configuration dialogue again and select *Shares*.

Two default shares have been set up: one for a Print Spool, to allow other machines to use a printer attached to the RISC OS box, and the other is to the Public folder on your computer's hard drive. The second has a share name of 'RiscPC' by default; you can change this by creating a replacement.

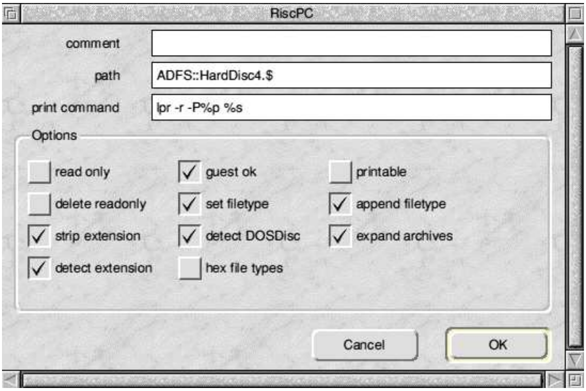


To create a new share click Menu over the Shares window you have just opened, go to *New share* and follow the arrow to the right where a box will ask you for the share name. When you click *OK*, you see something like the example here. If you double-click on one of the existing Shares shown, you will get a similar window where you can configure that particular share.

The *Path* should be the path to the folder that you want sharing with the other platforms: in this example, I have shared the entire hard disc and have made it read and write enabled by not ticking the *Read only* box .

Windows files on RISC OS

While SMBServer allows files on a RISC OS machine to be seen on the Windows side, it does not make the reverse possible. To see files on the Windows machine's disc in RISC OS, a copy of LanManFS or LanMan98 will be required – we will look at this in another issue.



comment

path ADFS::HardDisc4.$

print command lpr -r -P%p %s

Options

□ read only ☑ guest ok □ printable

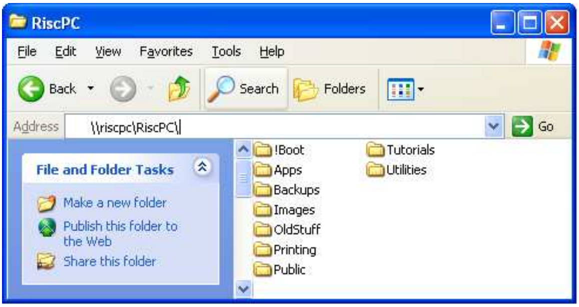
□ delete readonly ☑ set filetype ☑ append filetype

☑ strip extension ☑ detect DOSDisc ☑ expand archives

☑ detect extension □ hex file types

Configuring the details of an individual share

The only difference between the normal folder access ('Public' and 'RiscPC' in the example here) and the 'Spool' one is that *Printable* is ticked and *Detect extension* is unticked. This allows it to share a printer, as set by the *Print command*.



Address \\riscpc\RiscPC\

Putting it to use

Basically that is it: you can now access RISC OS hard drives, folders and files from Windows PCs and even from Linux (although in this case, using Moonfish is often easier, as we will see in a later issue).

To access the files in Windows, go to Windows Explorer (the 'filer') and type the server and share name. Using this server here, which I have given the name 'riscpc', and the share named 'RiscPC', you would type '\\riscpc\RiscPC' into the Explorer address bar and given the settings above you should see the contents of the RiscPC's HardDisc4 as shown.

The WROCC Guide to Networking

Following a successful evening 'demystifying networking' in September 2009, some of the WROCC members who ran the meeting started a series of articles in the Club newsletter – *The WROCC* – investigating the subject in more depth.

Between October 2009 and April 2010 we looked at how to connect computers together and make them talk, how to share an internet connection, and how to transfer files via Acorn's ShareFS and SMBServer.

This booklet collects together the articles that have been written so far into a single handy reference. It isn't the end of the story, however – over the coming months we intend to investigate the use of LanManFS and LanMan98 for accessing files stored on a Windows machine, and the use of Sunfish and Moonfish for doing the same with Linux and MacOS X.

If you would like to receive *The WROCC*, you can join us today for the introductory price of £7 for the first year. This isn't only for those who live within travelling distance of Yorkshire – our many 'postal members' receive a PDF copy electronically or in printed form via the Royal Mail for a small extra fee to cover printing and postal costs.

If you would like to know more about joining us, just ask at the Club stand for more information or get in touch after the show.

Steve Fryatt – Editor, *The WROCC*
editor@wrocc.org.uk

Wakefield RISC OS Computer Club
www.wrocc.org.uk
3 Riverdale Avenue, Stanley, Wakefield, WF3 4LF