SUSTAINABLE Softworks

*products*


# Getting Started with IPNetRouter



## Use our IPNetRouter Configuration Application


(strongly recommended for first time IPNetRouter users)

## Go to General Setup Examples

## Download these instructions as PDF

These pages provides simple directions to start using IPNetRouter. The directions on these pages assume you are running the latest version of IPNetRouter.

Before getting into specific examples, I'd like to provide a quick overview and some general instructions. First, IPNetRouter offers three main features:

1. IP Multihoming - the ability to use more than one IP interface like Ethernet and PPP or two Ethernet cards at the same time.
2. IP Forwarding - the ability to route IP traffic from one network interface to another.
3. IP Masquerading - the ability to hide multiple hosts behind a single public IP address.

You can use any or all of these features as needed. If you just want to access two IP networks without routing traffic between them, do not enable IP Forwarding. If you do not need to hide multiple hosts behind a single public IP address, do not enable IP Masquerading (which acts like a firewall by using Network Address Translation).

Taken together, these three features allow you to share a single connection to the Internet with your entire LAN. One of your Macintosh computers will run IPNetRouter and become a "gateway" that provides Internet access to the rest of your LAN. The other machines on your LAN may be called "slaves" or "clients" and do not need to run IPNetRouter, Open Transport, or even be

Macintoshes. Any computer with a configurable TCP/IP stack may be used as a client machine, including other Macs, Windows machines, and Linux/UNIX boxes.

To configure your gateway, you will use the TCP/IP Control Panel to setup one IP interface, and then use IPNetRouter to configure and bring additional IP interfaces as needed. Which interface should be configured from the TCP/IP Control Panel depends on your specific situation (see the examples below).

Since IPNetRouter depends on Open Transport and TCP/IP being present, there are two very important setup considerations for the machine running IPNetRouter:

1. **You MUST UNCHECK "Load Only When Needed"** in the TCP/IP Control Panel so that TCP/IP will remain loaded at all times (**Edit > User Mode** set **Advanced** mode, select **Options...**). If you are connecting via PPP, you must uncheck "Load Only When Needed" for both your Ethernet and PPP configurations.

2. **Your IPNetRouter machine MUST NEVER go to sleep**.

---

# EXAMPLES

The following examples cover the most common network configurations.

### Basic Networking

- Ethernet Basics

### General IPNetRouter Setup

- Connect to the Internet using OT/PPP and Ethernet
- Connect to the Internet using a cable modem (or ADSL/DSL modem) and **single** Ethernet
- Connect to the Internet using a cable modem (or ADSL/DSL modem) and **dual** Ethernet <-- **HIGHLY RECOMMENDED!** (here's why....)
- Connect to the Internet using PPPOE (e.g. Sympatico PPPOE, EnterNet or MacPoET drivers)
- Connect to the Internet using a one way cable modem

- Using LocalTalk (MacIP over AppleTalk)
- Using DHCP

**Advanced Techniques**

- Using an Unnumbered Interface
- Inbound Port Mapping
- Building Your Own Low Cost Firewall
- DHCP and Mac OS
- IPNetRouter Guide to AirPort

**Help** (what to do if it doesn't work right away)

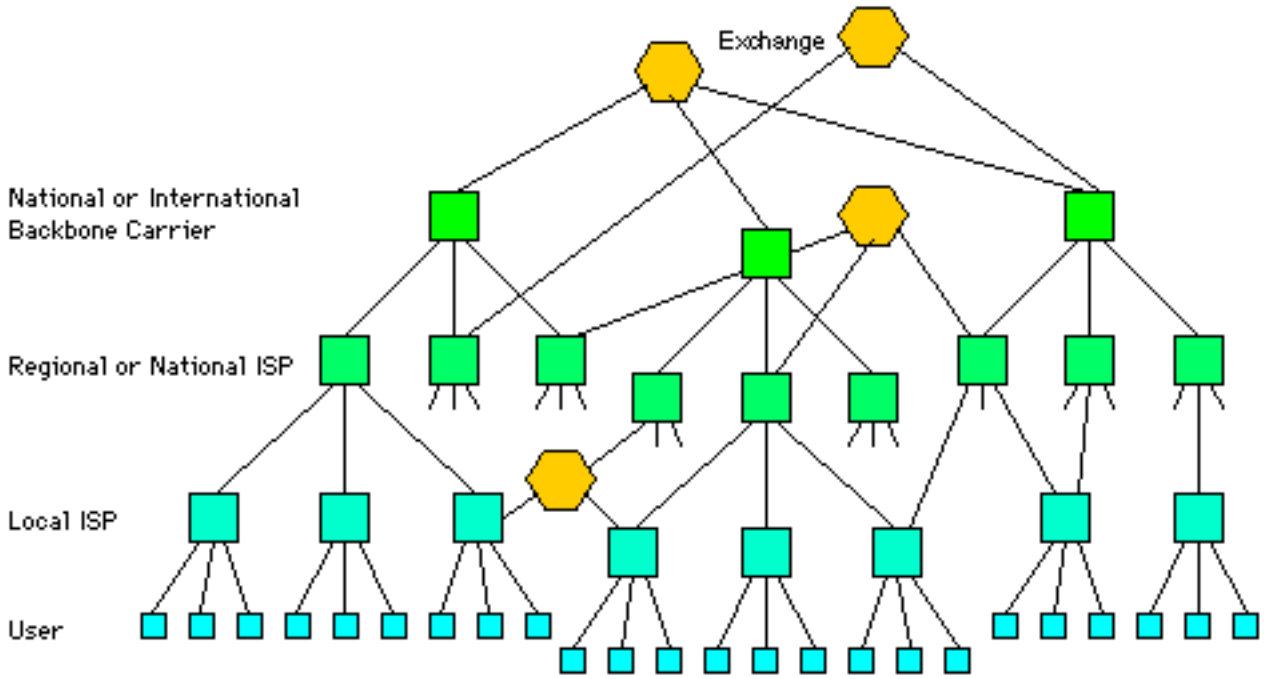
- Troubleshooting Tips

These directions may seem foreign if you are not familiar with IP networking. Here is some further background information to help you get started.

## Internetworking 101

The Internet is actually a network of thousands of privately run networks using different equipment with only minimal coordination needed between them. This minimal coordination gives the Internet the ability to expand and evolve rapidly since almost anyone can add their network to the Internet. It can also lead to problems when some piece of equipment you know nothing about breaks down or coordination fails.

To manage this vast enterprise, the Internet is organized into hierarchical sections or domains.




Message processing computers called "Routers" or "Gateways" are used to connect individual networks together, with specialized Gateways used at exchange points where different carriers or service providers can exchange traffic for their respective networks.

In order to communicate with another computer on the Internet, your computer will normally go through four steps:

1. Lookup the address of the host you wish to communicate with. Since people aren't very good at remembering lots of numbers, host computers on the Internet are usually identified by a name. The name usually includes both the name of the individual host, and a hierarchy or list of names that describe the domain or part of the network where it resides. This is like calling directory service to find someone's telephone number. On the Internet, this type of directory service is called Domain Name Service (DNS), and computers that handle these requests are called Name Servers.

2. Determine if the address is local to this network, or if the message needs to be forwarded to another network. This is kind of like determining if you need to dial 1 plus the Area Code before dialing the rest of a phone number.

3. If the address is local to this network, the message is delivered directly.

4. If the address is not local to this network, the message is sent to a router or gateway that can forward the message on to its destination. This process can be repeated through a dozen or more networks before the message is actually delivered.

Understanding these steps is helpful because it tells you what information every computer must have to access the Internet.

1. A unique IP address that identifies the IP interface.
2. A network mask used to divide the IP address into a **network number** and **host number**.
3. The address of a Name Server used to translate domain names to their corresponding IP address.
4. The address of a router or "Default-Gateway" to forward any packets that cannot be delivered directly.

The secret to efficient routing on the Internet is that IP addresses are carefully assigned based on where a computer is attached to the network. Just as the area code and exchange part of a phone number are used to identify where a call needs to be routed, the first part of an IP address serves the same purpose. When you move to a new calling area, you are assigned a phone number for that area. Similarly, when you choose an ISP (Internet Service Provider), you are assigned an IP address for that part of the network.

To make configuring your computer easier and reduce the total number of IP addresses required, many ISPs will assign you a temporary IP address every time you dial-in to one of their access servers. You simply set your computer to get its IP address from the server. On some systems, the server can also provide your network mask, router, and name server address. Configuring an IP interface essentially means identifying what network connection the interface should use, and then filling in the required information described above.

IPNetRouter now includes a DHCP (Dynamic Host Configuration Protocol) server so you can optionally enable this server and then tell any clients on your LAN (Local Area Network) to configure via DHCP. Using DHCP avoids the need to manually configure each computer on your LAN before it can be used, and is especially useful if you have many computers on your LAN or frequently move computers from one network to another. To get started using DHCP, press the

help button in the lower left corner of the IPNetRouter DHCP window.

## All I want to do is share my Internet connection between two machines, why do I have to learn all this stuff?

IPNetRouter is the first low cost program that allows you to do this on a Macintosh using Open Transport. The program to make it simple hasn't been written yet.

For more information on TCP/IP networking, may I suggest:

- "Internetworking With TCP/IP, Volume 1", by Douglas E. Comer
- "TCP/IP Illustrated, Volume 1" by W. Richard Stevens

For more tutorial information on Macintosh networking:

- Three Macs & a Printer

---

## When sharing my cable/DSL/ADSL modem why should I consider using the dual ethernet setup over the single ethernet setup?

Wherever possible, we strongly recommend the dual ethernet setup for several reasons:

- It is much more secure. Your private LAN is physically isolated from the cable/DSL/ADSL modem network with the dual ethernet setup. With the single ethernet configuration, all of your client machines are physically connected to the public network, they just have different IP addresses (hopefully).
- There is potential for an IP address conflict with the single ethernet setup. Someone else on the cable/DSL/ADSL modem network might be running a NAT router (Mac, Windows, LINUX, etc.) in the single ethernet configuration and choose the same IP addresses for their client machines. One of you will get bumped (your ethernet driver will shut down).
- The dual ethernet configuration offers much better performance. Packets are routed over physically separate interfaces...which drastically eliminates packet collision problems. If you are running a web or other server on a client machine, the dual ethernet setup is almost a necessity in order to eliminate this packet collision problem.

- You can safely run the IPNetRouter DHCP server with the dual ethernet setup (you SHOULD NOT run the DHCP server with most single ethernet setups since you may become a DHCP server to the entire cable/DSL/ADSL modem network). This may or may not be important to you (depends on how you wish to setup your client machines...manually or using a DHCP server).

Enjoy,

Peter Sichel
February 6, 1998

---

# Building Your Own Low Cost Ethernet

Like many people, I have more than one generation of Macintosh computers at home that are networked together to share files and a LaserWriter. When I decided to upgrade my Local Area Network from LocalTalk to Ethernet, I had a lot of practical questions that didn't seem to be answered in any one place.

I created this page to describe the practical details of building your own low cost Ethernet.

**Contents:**



## Is It Really Worth Upgrading?

I think so. Ethernet offers about 5-10 times the performance of LocalTalk. Ethernet interface hardware is also better designed to off-load your CPU. It is much easier to continue working while sharing files or printing via Ethernet.

## Why 10Base-T

Ethernet can use several different types of physical wiring or "media" to satisfy different design requirements. By today however, 10Base-T Unshielded Twisted-Pair (UTP) Wiring is by far the best choice for most installations.

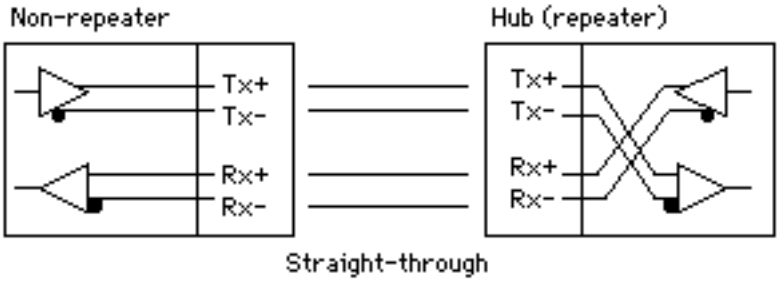
Some key advantages of 10Base-T Wiring:

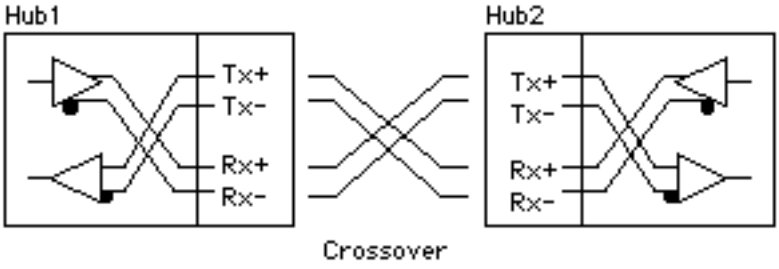
- 10Base-T interfaces are already included on many desktop Macs
- Can add or remove devices without disrupting the network
- Easiest to test and track down any cabling problems
- Can be upgraded to full-duplex and 100 Megabits per second

With 10Base-T, each segment of network cable connects exactly two devices. To connect more than two devices, you must use a "Hub" or multi-port repeater.

# Can I Get By Without A Hub If I Only Want To Connect Two Devices?

Perhaps. While this can be made to work in many cases, it is not officially supported or recommended. Normally each 10Base-T cable segment runs between an end node such as a host or printer and a hub. Hub ports are designed to be connected to end nodes using a straight-through cable. To connect two end nodes directly, or two hub ports, it is necessary to use a crossover cable so that each transmitter is connected to the corresponding receiver at the other end of the cable as shown below.




I normally carry a 10Base-T crossover cable with my Powerbook in case I want to setup an instant two node network with another host. Before you assume this is all you need however, it is important to realize that many 10Base-T end nodes assume they are connected to a hub.

Upon Startup, Macs with built-in Ethernet normally check to see if they have a valid Ethernet connection before starting AppleTalk on the Ethernet port. If your Mac is connected to a hub this works as expected. If your Mac is connected through a crossover cable to another Mac that is not powered on, your system may complain there is a problem with your Ethernet connection and automatically switch to using LocalTalk. Even

if the remote machine is turned on, you may still encounter problems getting your Mac to recognize it has a valid Ethernet connection. [I've been told Macs with built-in Ethernet look for a valid carrier on the receive line. Some Network Interface Cards provide this, others don't.]

A small 10Base-T hub is no longer expensive and gives your network room to grow.

**Notes**:

1. Some hubs provide an optional "uplink" port that is pre-wired or can be switched to connect directly to another hub, called an MDI-X (Media Direct Interface-Crossover) port.
2. If you do use a crossover cable, be sure to label it carefully so you don't confuse it with other 10Base-T cables.

# How Do I Choose A Good Low Cost Hub?

I like the TRENDnet TE-900 series because:

- An 8-port hub costs about $70.
- The hub provides two diagnostic LEDs for each port to show the Link state (up or down), Rx data (blinking), and Partition (when a port has been isolated due to excessive collisions or other interface problem).
- The designers didn't confuse the front panel with the rear panel. The front panel is clearly labeled and contains all the diagnostic LEDs. The rear panel contains all the network and power connections.
- The hub is small, lightweight, and can be wall mounted or stacked with other equipment (and the color even matches).

10Base-T hubs are generally very reliable and guaranteed for years. Choose a hub that complies with IEEE 802.3 specifications for use on 10Base-T UTP cabling based on price and any features you find attractive. Some hubs include a ThinWire 10Base-2 uplink port that can be useful if you encounter older 10Base-2 equipment, or as a "backbone" port for connecting more than four hubs.

You may see so called "managed hubs" advertised for considerably more money. Managed hubs allow a network administrator to view the hub's operating status from a remote management console. For small networks, it's easier to "manage" a hub by simply looking at the status LEDs on the front panel. Managed hubs are intended for larger networks with lots of equipment to keep track of.

# What Do I Need to Know to Install My Own Wiring?

If all the stations you want to network are in a single room, you can simply run 10Base-T

patch cables between each station and your network hub. If your network spans several rooms or floors, you will probably want to install wall jacks (similar to but not the same as common telephone wiring). Each cable segment between an end node and hub must not exceed 100 meters. You can daisychain or "cascade" up to four hubs using 10Base-T.

10Base-T uses 8-pin RJ-45 modular style connectors available from many electronic supply stores and catalogs.

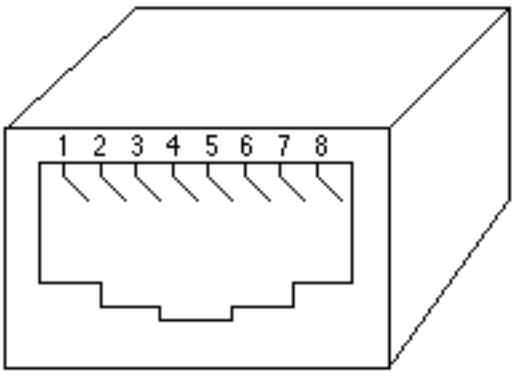
Eight-conductor data cable contains 4 pairs of wires. Each pair consists of a solid (or predominantly) colored wire and a white wire with a stripe of the same color. The pairs are twisted together. To maintain reliability, you must connect each pair to the appropriate pins and not untwist them any more than necessary.

Data cables normally use AWG #22-26 wire with #24 the most common. Solid conductor is preferred for longer runs and in-the-wall wiring, while more flexible stranded conductor cables are generally used for patch cords. If you decide to buy a crimping tool (about $40 for a decent one) and make your own cables, notice there are different RJ-45 plugs for solid versus stranded conductor cable. The more common modular plugs are designed for stranded patch cable. Look for specially labelled solid conductor plugs if that is what you need.

PVC insulation is fine for patch cords, but for in-the-wall wiring, you should use cable with a "Plenum" or Teflon jacket to meet fire safety codes.

Two grades of data cables and jacks are commonly available referred to as Catagory 3 (CAT-3) and Catagory 5 (CAT-5). The difference is that CAT-5 requires more twists per inch and can support 100 Megabit per second Ethernet. CAT-3 is usually a little less expensive and fine for 10 Mbps Ethernet. I always install CAT-5 for in-the-wall wiring since it is difficult to change later, but don't worry too much about patch cords.

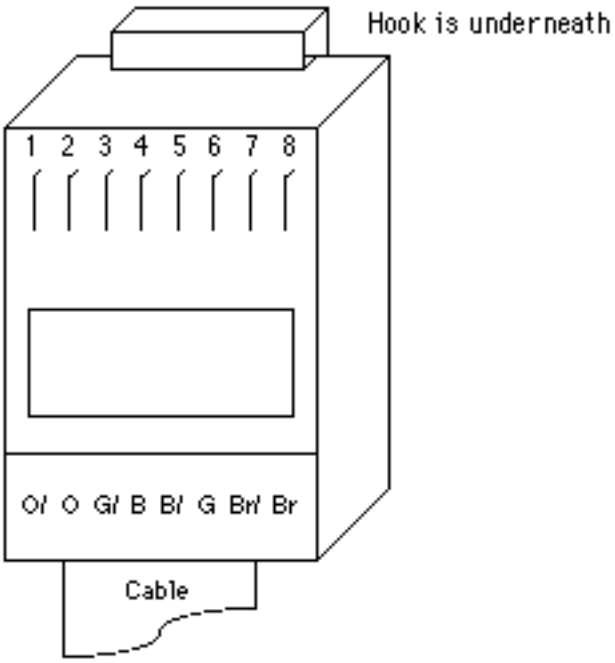
The illustrations below give the correct wiring.




**RJ-45 Connector**

| Pin | Media Direct Interface Signal | |
|---|---|---|
| 1 | Tx+ | (transmission) |
| 2 | Tx- | (transmission) |
| 3 | Rx+ | (reception) |
| 4 | | Not used |
| 5 | | Not used |
| 6 | Rx- | (reception) |
| 7 | | Not used |
| 8 | | Not used |

The pairs designated for 10Base-T Ethernet are Orange and Green. The other two pairs, Brown and Blue, can be used for a second Ethernet line or for phone connections. Note that the Blue pair is on the center pins and conveniently corresponds to the Red and Green pair in a normal phone line.




**Color Code for RJ-45 Plug**

| Pin | Color | |
|---|---|---|
| 1 | O/ | Orange stripe |
| 2 | O | Orange |
| 3 | G/ | Green stripe |
| 4 | B | Blue |
| 5 | B/ | Blue stripe |
| 6 | G | Green |
| 7 | Br/ | Brown stripe |
| 8 | Br | Brown |

There are actually several different Color Code conventions for wiring RJ-45 plugs and jacks, so don't assume the wiring just by looking at the colors. As long as you follow a single convention that assigns wire pairs to the corresponding signals, you shouldn't have any trouble.

# Do I Need A Transceiver?

Macs with built-in Ethernet may have one of two different interface connectors:

1. An RJ-45 10Base-T port
2. An AAUI (Apple Attachment Unit Interface) port

Some Macs include both. If your Mac already has a 10Base-T port (most PCI PowerMacs with built-in Ethernet), you don't need a separate transceiver. If your Mac only provides an AAUI port (NewBus PowerMacs and all 68K Macs with built-in Ethernet), you will need an AAUI to 10Base-T transceiver ($30-$40). The purpose of the AAUI port is that it allows you to select a transceiver for the type of Ethernet media you want to use (10Base-T, ThinWire, or ThickWire).

If your Mac doesn't include built-in Ethernet, you will need to get a Network Interface Card with the correct interface (10Base-T) for the media you want to use.

# Now That I Have My Macs on Ethernet, How Do I Connect My LaserWriter?

You can choose one of your Macs to run Apple's "LaserWriter Bridge" software (free) to make any LaserWriters attached to that machine via LocalTalk appear on your AppleTalk Ethernet. The machine acting as the LaserWriter Bridge must be turned on for your other computers to print.

If you don't have an Apple LaserWriter or compatible printer, you can buy a software LocalTalk Bridge (around $50), or buy a dedicated hardware LocalTalk to Ethernet Bridge (around $250).

# It's A Wrap

I hope you found this useful and welcome any comments.

- Peter Sichel
<mailto:psichel@sustworks.com>

## Related Links

Three Macs & a Printer

Crossover Cables

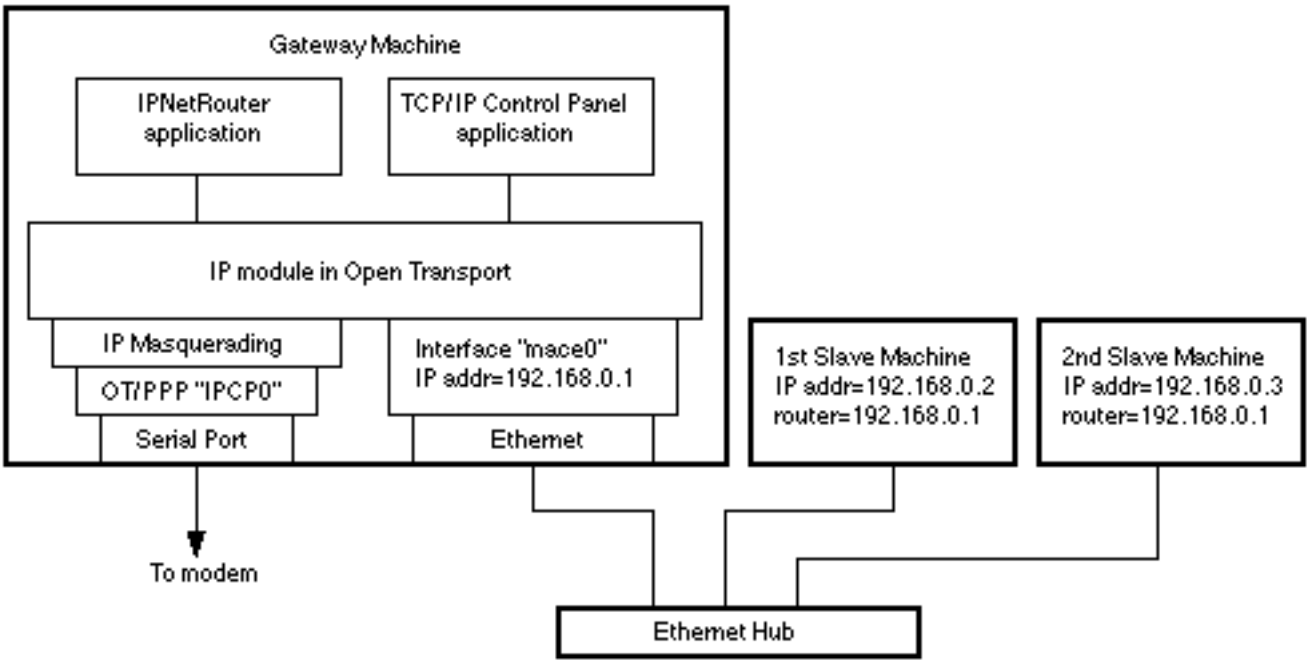
products

# IPNetRouter

IPNetRouter.PPC

Return to Getting Started

## Connecting to the Internet using OT/PPP and Ethernet



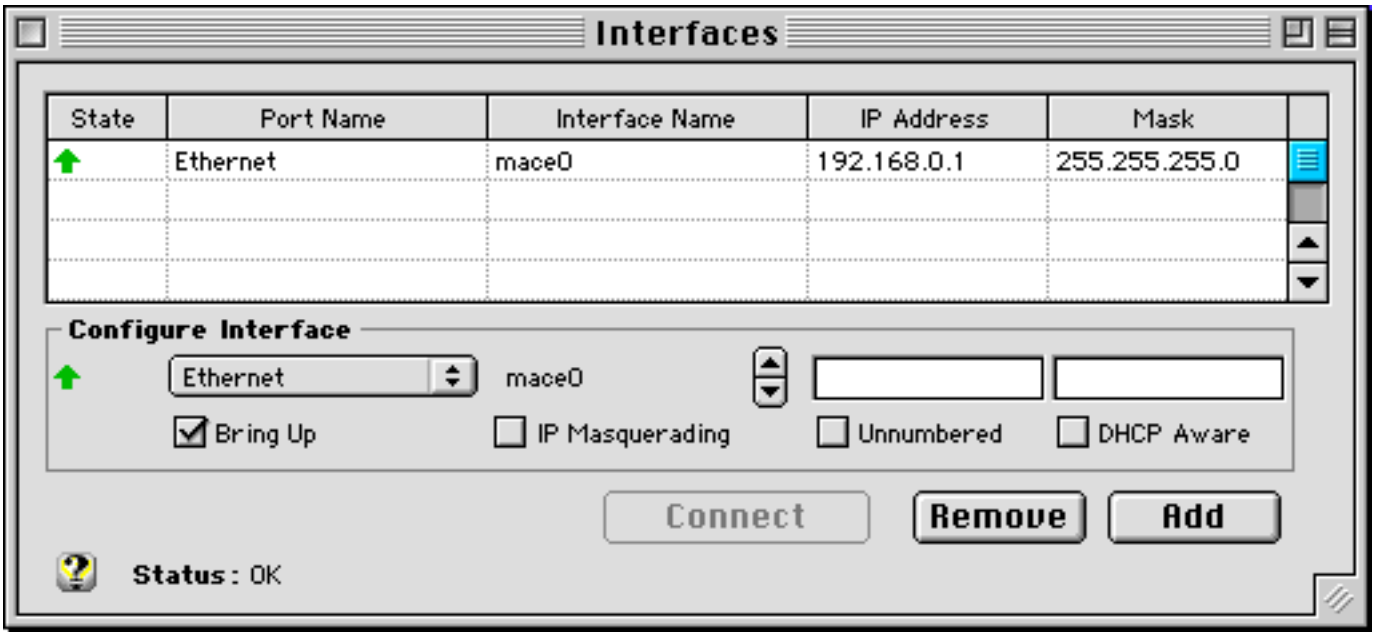

These directions assume that you normally connect to an Internet Service Provider using OT/PPP and that your TCP/IP Control Panel is initially configured so that PPP will connect automatically when you launch a TCP/IP application. Notice in Mac OS 8.5 the PPP control panel has been renamed "Remote Access."

Open the TCP/IP Control Panel and press Cmd-K to bring up the "Configurations" dialog. Select the PPP configuration you normally use to connect with your ISP and rename it to "IPNetRouter". Then press Duplicate to make a copy. Give this copy a different name like "Ethernet LAN", press "Make Active", and then using the TCP/IP control panel change it

to "Connect via: Ethernet", enter the IP address of this machine on your Ethernet LAN (example: 192.168.0.1), enter the subnet mask (example 255.255.255.0), leave the router address empty, leave the name servers as currently configured to work with your ISP, close the control panel and save this configuration.

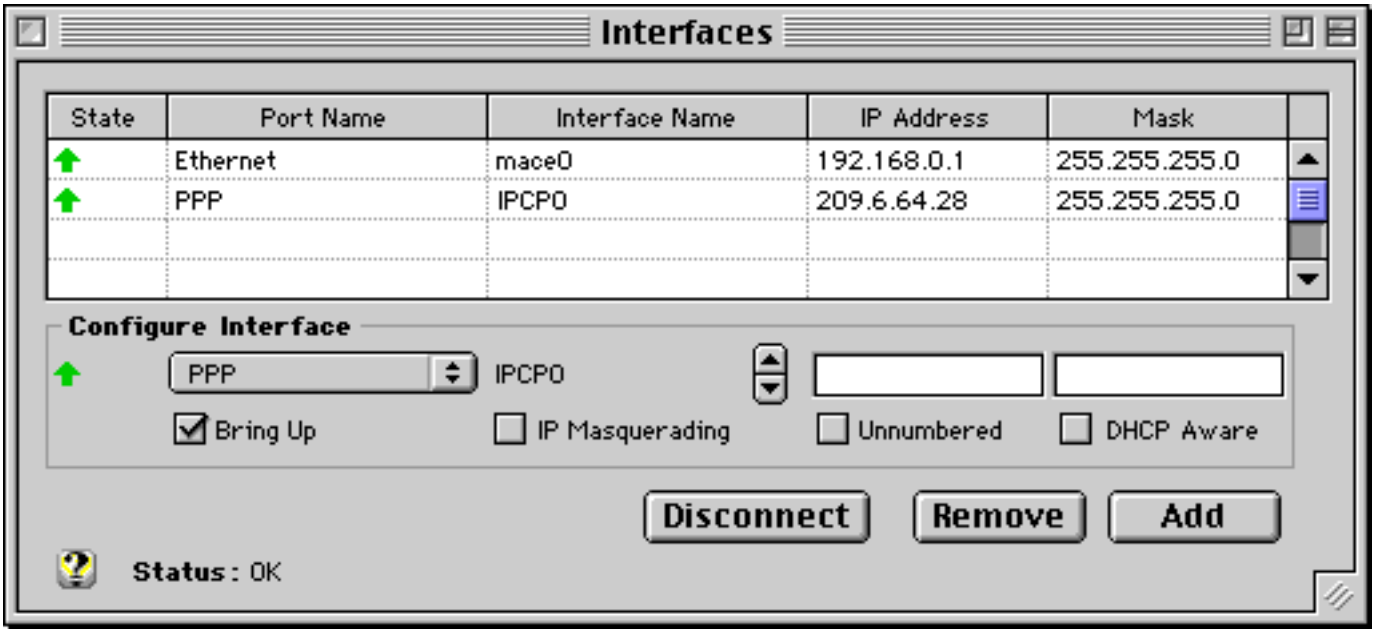
> Notice it is important to make this new "Connect via: Ethernet" configuration "Active" or the "primary interface" currently selected with the TCP/IP Control Panel. If you leave OT/PPP as the primary interface, Open Transport will make TCP/IP services unavailable and unload the TCP/IP stack when your PPP connection closes causing your system to crash.
>
> When IPNetRouter tries to open a PPP connection, PPP needs to be able to find its configuration settings in the TCP/IP Preferences File. It does this by looking for the currently selected configuration, but we just set the TCP/IP Control Panel to a "Connect via: Ethernet" configuration. To prevent PPP from complaining that it is not currently selected in the TCP/IP Control Panel, IPNetRouter temporarily modifies the TCP/IP Preferences File to make PPP appear as if it is currently selected. Since it is possible to have more than one PPP configuration saved in the TCP/IP Preferences File. IPNetRouter uses the configuration named "IPNetRouter", or if no such configuration exists, the first PPP configuration it finds. For this reason you should name the PPP configuration you want IPNetRouter to bring up "IPNetRouter".

Launch IPNetRouter. An "Interfaces" window like the one below should appear.




Port Name "Ethernet" is the ethernet interface you configured using the TCP/IP Control Panel.

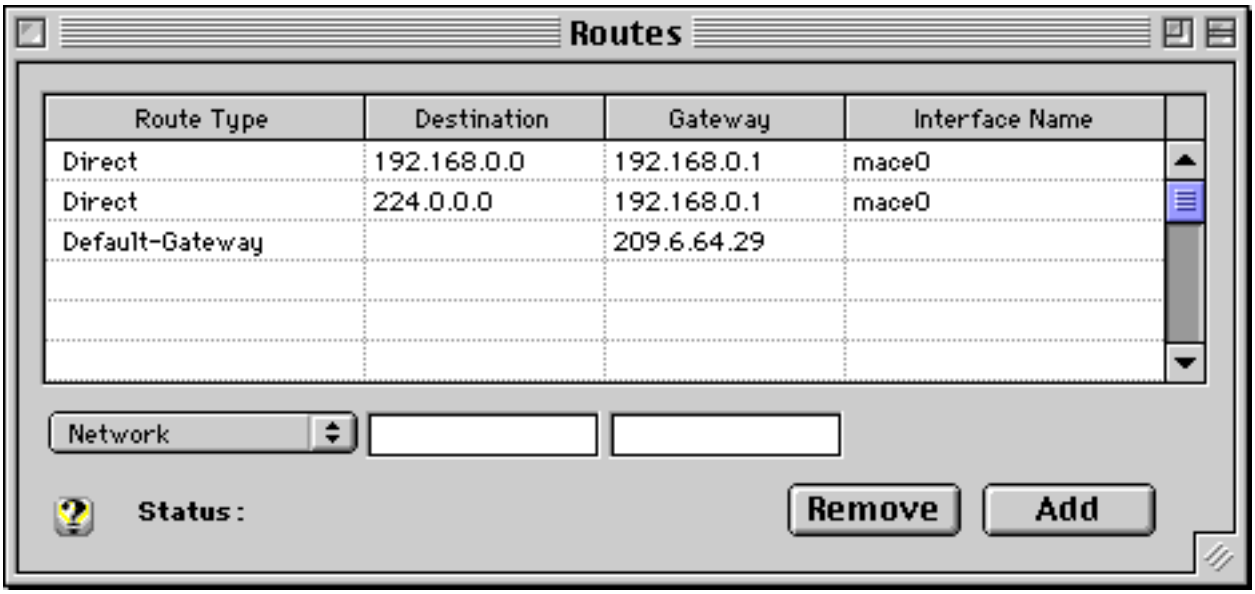
Select your PPP device using the Port Name popup menu and press Add (Apple's OT/PPP also called Remte Access is recommended).




This should cause PPP to connect automatically and add an IP interface to your ISP using the PPP device you selected.

> If IPNetRouter reports no such device or address, it means Open Transport was unable to configure OT/PPP. Make sure the "Connect via: PPP" configuration you want IPNetRouter to use is named "IPNetRouter" in the "Configurations" dialog of the TCP/IP control panel that appears when you press Cmd-K (so IPNetRouter can find it). Also make sure OT/PPP is configured to connect automatcially when starting TCP/IP applications.

The "Routes" window should now appear as follows.

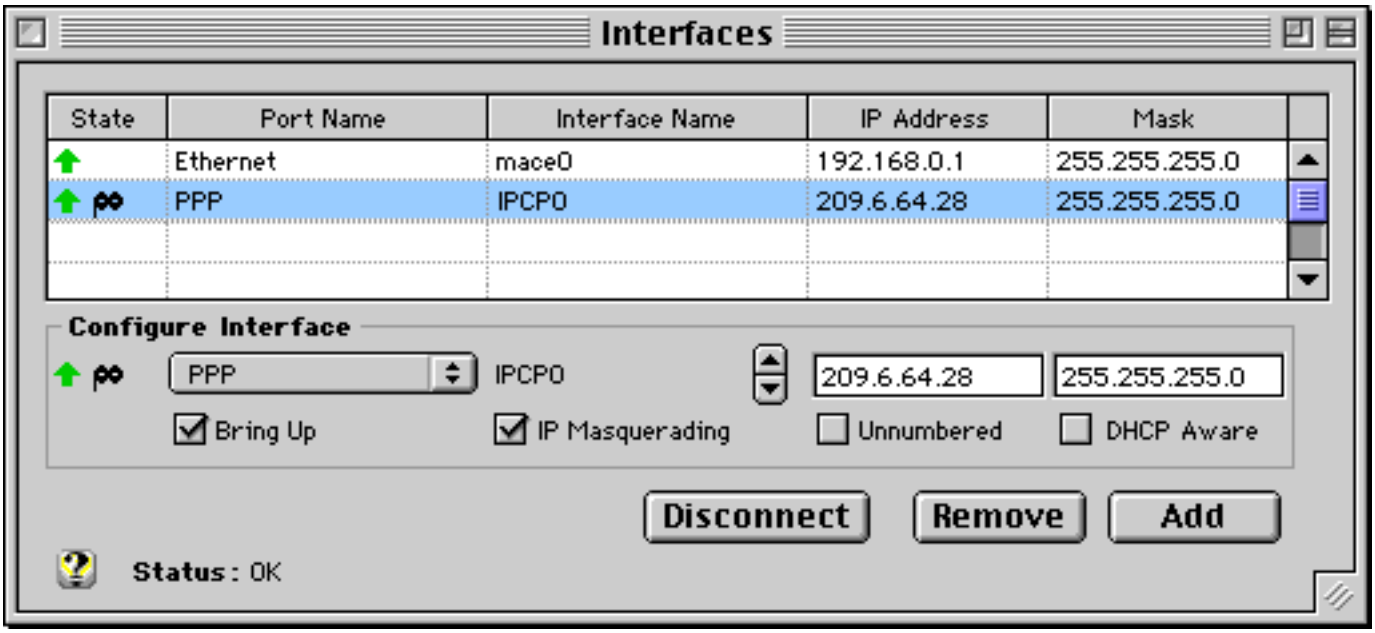



Notice the address of the Default-Gateway is very close to the address of your PPP interface. This default gateway is a "pseudo-gateway" at the far end of your PPP connection. If there is more than one Default-Gateway, a router address was probably specified in the TCP/IP Control Panel. You must remove this extra Default-Gateway if there is no such actual gateway. Select the corresponding row and press Remove. In

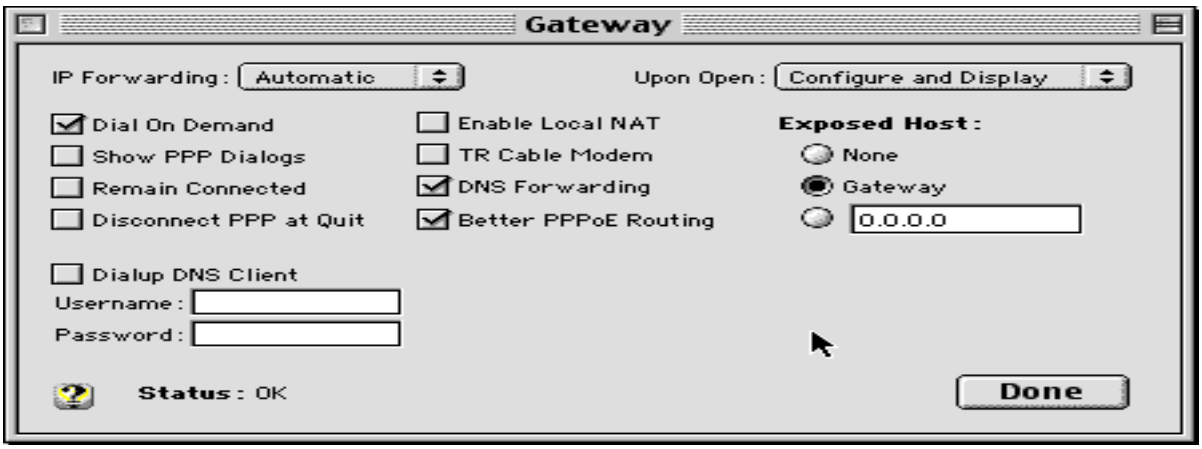
general, you should only have one Default-Gateway.

If you have a Ping tool like IPNetMonitor, you can now ping hosts on the Internet as well as hosts on your Ethernet LAN. Congratulations, your Macintosh is now multihoming!

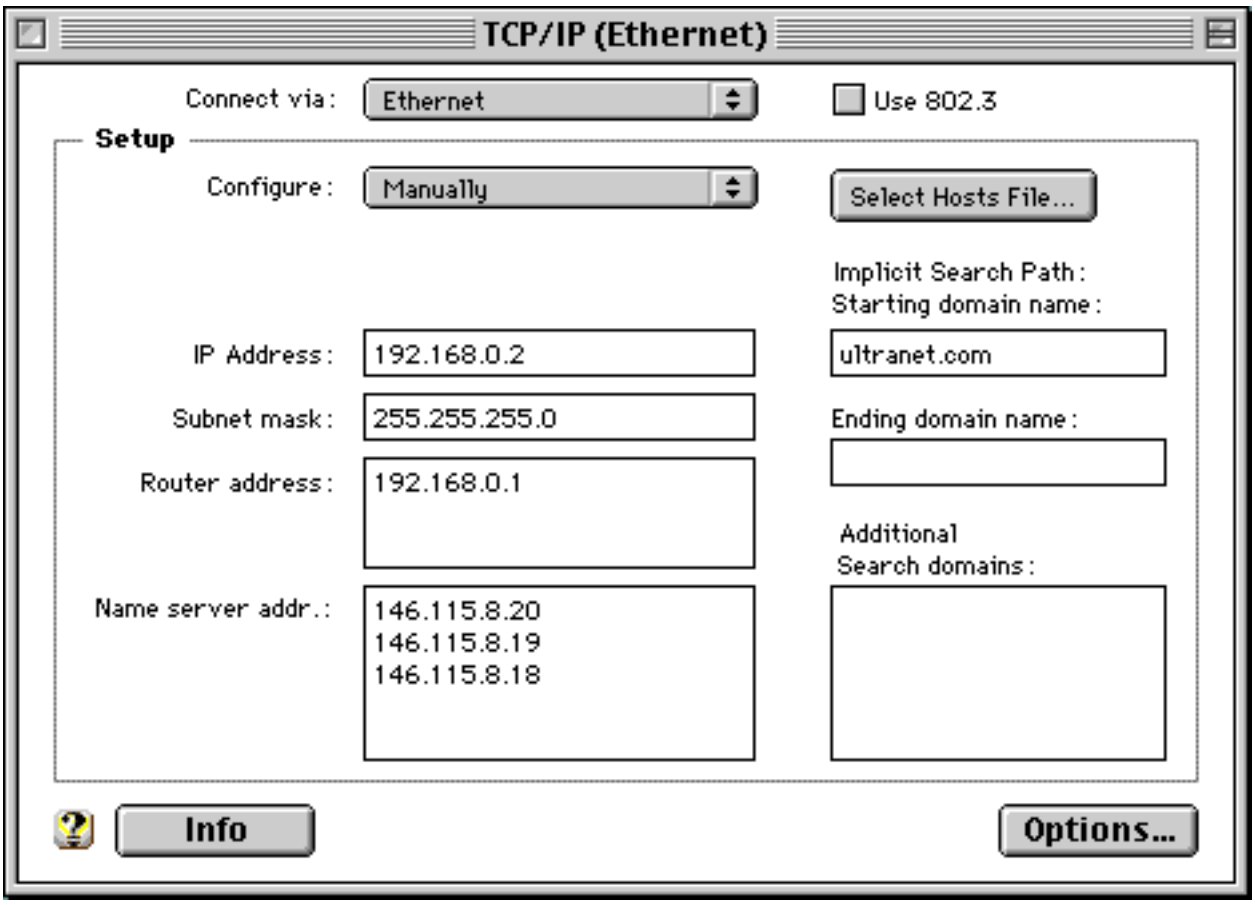
To allow other machines on your Ethernet LAN to use your PPP connection to the Internet, select the PPP interface in the Interfaces window. Notice the values from this row are copied to the Configure Interface box below. In the Configure Interface box, check "IP Masquerading" and then press Add (you must press Add to actually update the record in the Interfaces table). A small mask should appear by this row in the table.




When you enable IP Masquerading, IPNetRouter automatically sets "IP Forwarding: Automatic" in the Gateway Window.




For other machines on your Ethernet LAN to connect to the Internet through the computer running IPNetRouter (your IP Gateway), you must configure these other machines to use your IP Gateway as their router address (example 192.168.0.1) in the corresponding TCP/IP Control Panels. The TCP/IP Control Panel for your First Slave Machine might appear as follows.




To save your IPNetRouter configuration so you can easily restore it later, select Save under the File menu. To restore this configuration, double-click on the resulting configuration document, or include it in your Startup Items folder.

You can now close the Interfaces window, but you must leave the IPNetRouter application running to continue multihoming. If your PPP connection closes, it should reconnect automatically when there is IP traffic for that interface. You can tell it to manually reconnect by clicking on the "Connect" button in the Interfaces window.

Notice the IP address of this interface and corresponding pseudo-gateway will change if your ISP assigns you a new IP address.

## Additional Notes:

Some internet software (like your ftp client) needs to send the IP address of your Macintosh to the server. Since your Macintosh now has more than one IP address, the software may not realize it needs to check which address to use and try to send the wrong one. If this happens, you will get an error from the server. This is not a problem with multihoming, but simply a matter of updating these applications to check which IP address to use. IP Masquerading can provide a work around for this problem since the incorrect address in the ftp PORT command will be translated to appear as the correct address when IP Masquerading is enabled.

**IP Addressing**: I chose the example IP addresses from network 192.168.x.x because this network range is reserved for private Internets (see RFC 1918). In order to route IP

datagrams between two networks, each network must normally have its own network number (the most significant part of the IP address logically ANDed with the network mask). If you wish to follow the suggested example, each machine on your ethernet LAN must have a unique address from network 192.168.0.x (192.168.0.1, 192.168.0.2, 192.168.0.3, etc.). The Macintosh running IPNetRouter (192.168.0.1 in this example) becomes the default router or gateway for the other machines on your ethernet LAN.

Enjoy!

- Peter Sichel

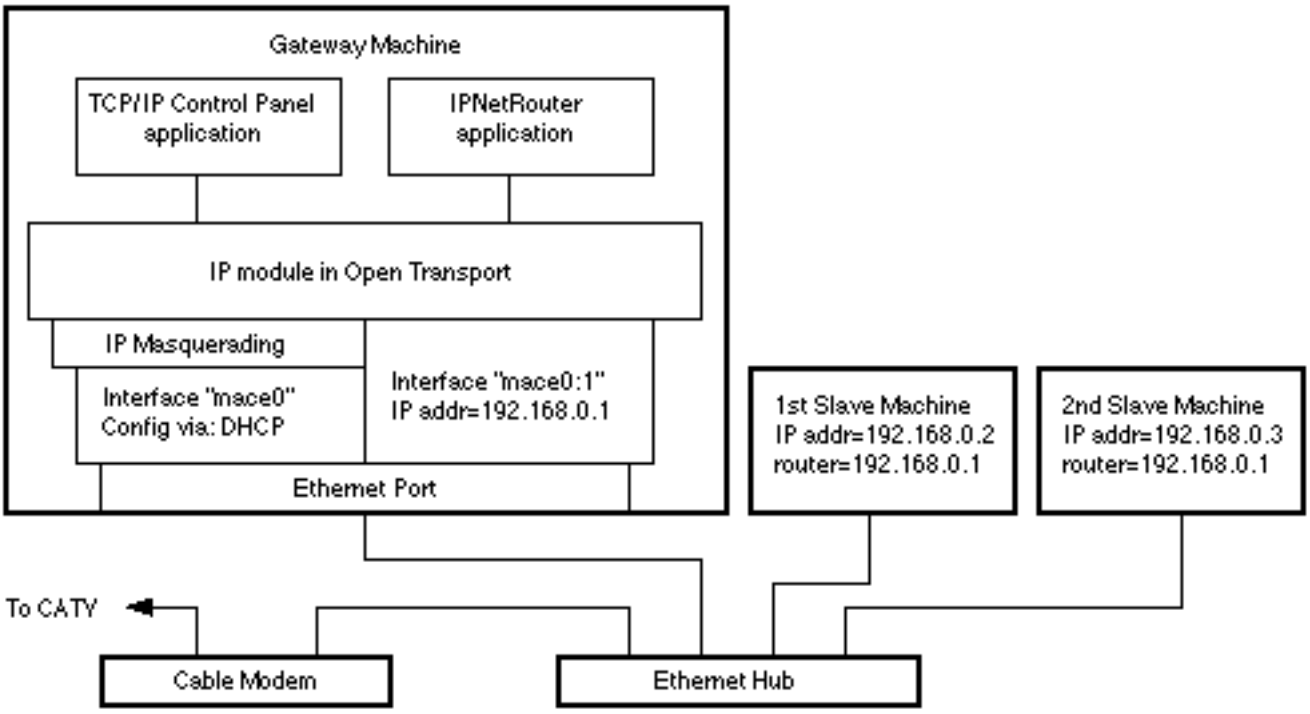
products

# IPNetRouter

IPNetRouter.PPC

Return to Getting Started

# Connecting to the Internet using a cable/DSL/ADSL modem and single ethernet




These directions assume that you normally connect to the Internet using a cable/DSL/ADSL modem. To share your cable/DSL/ADSL modem with other machines on your ethernet LAN, your cable/DSL/ADSL modem and the machines on your LAN should be connected to a 10Base-T hub. In the single ethernet setup, the cable/DSL/ADSL modem is attached to the "UPLINK" port of the HUB and the gateway and client machines are connected to regular ports of the HUB.

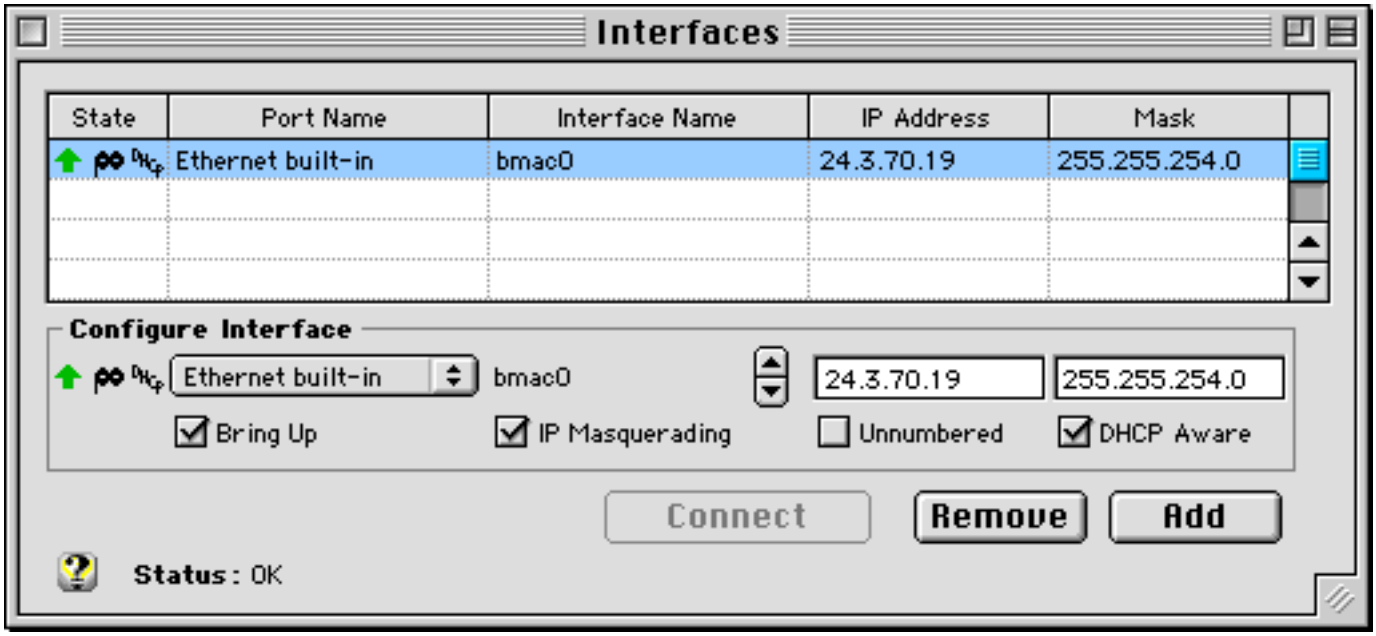
Alternatively, if you have two ethernet ports in your gateway machine, you can connect the cable/DSL/ADSL modem to one ethernet, and your LAN to the other (in this case, your 2nd IP interface will be on your 2nd ethernet). Notice the two Ethernet configuration has important security advantages if you use personal file sharing since it isolates your

private LAN from the cable/DSL/ADSL modem network.

The basic steps to configure IPNetRouter to work with a cable/DSL/ADSL modem in the single ethernet port setup are as follows:

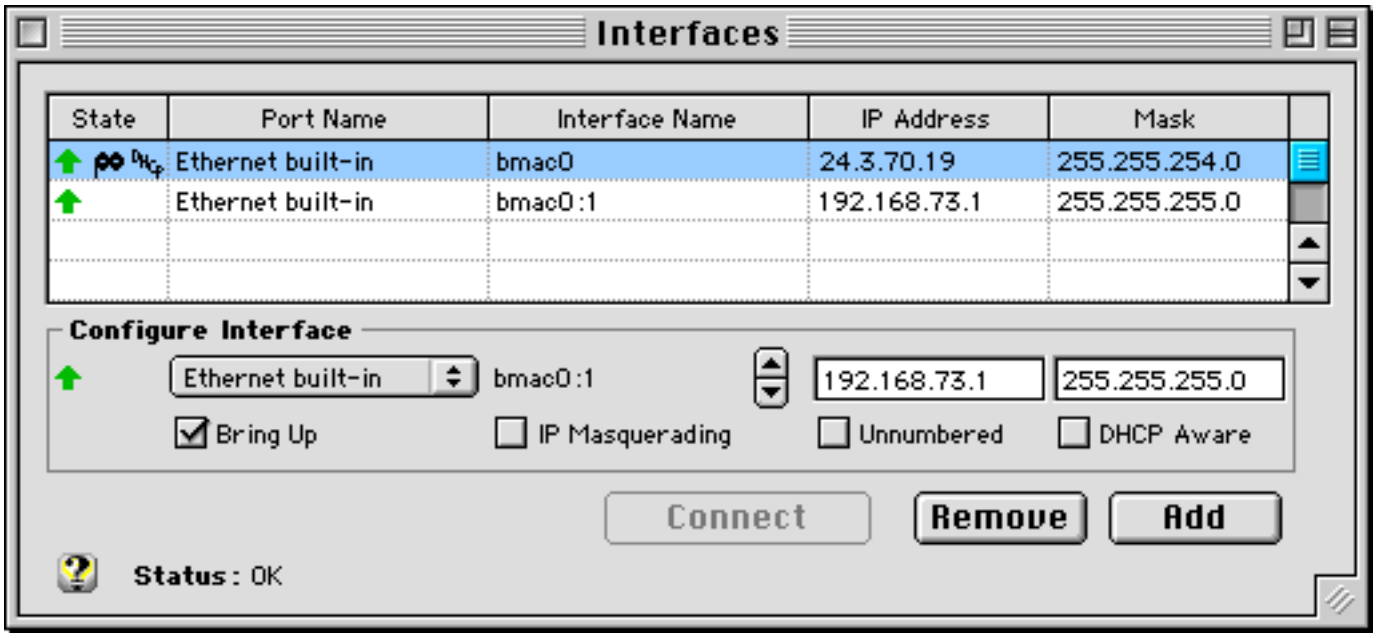
1. On the gateway machine, use the TCP/IP control panel to configure the Ethernet interface to use your cable/DSL/ADSL modem as instructed by your service provider (example: @Home). **Make SURE that the "Load Only When Needed" option is UNCHECKED** (you must be in Advanced User mode to get to the TCP/IP options. Choose User Mode under the Edit menu and select Advanced). If you can already access the Internet from this machine via your cable/DSL/ADSL modem and you are SURE that the "Load Only When Needed" option is UNCHECKED in the TCP/IP control panel, you're ready to proceed to the next step.

2. Startup IPNetRouter on the gateway machine. In the Interfaces window, click on the top row (with your public IP address on the ethernet port). With this interface selected, check the "IP masquerading" box and then press Add (if you receive your IP address via DHCP from your cable/DSL/ADSL modem or DSL modem provider, then also check the the "DHCP Aware" box). You should see a little mask icon appear in the state column for the Ethernet interface (and a DHCP icon if you checked the DHCP checkbox and added this feature). Notice you need to enable IP Masquerading on the interface that communicates directly with your cable/DSL/ADSL modem. Your Interfaces window should now appear similar to the following (your IP address and mask values will vary depending upon those values assigned by your ISP):




3. Next you need to create a 2nd IP interface on your Ethernet port. Continue to work in the Interfaces window using the "Configure Interface" input area just below the main display grid. Select Ethernet from the popup menu under the Port Name column. Click the up arrow to the right of the Interface Name field once. You should see ':1' added to the interface name. Enter a private IP Address and Mask. We

suggest an IP address like 192.168.X.1 (where X is a number between between 1 and 254...in this example we choose 73, but you should choose your own subnet number.) and a mask of 255.255.255.0. Only the "Bring Up" checkbox should be checked. box. Press Add. You should see all the values you just entered appear in a new row in the main Interfaces table. Your Interfaces window should now appear similar to the following:




You might ask why we choose the 192.168.73.X subnet for this installation. The reason is that you do NOT want to have conflicting private IP addresses with other potential NAT router users out on the cable/DSL/ADSL/DSL modem network (and with the single ethernet configuration you are exposing your private interfaces to the public network...which can result in possible IP address conflicts). The most often used private IP subnet is 192.168.0.X, so you probably want to avoid using this subnet for your private network. This is why we went with the 192.168.73.1 IP address for the IPNetRouter machine. You can use any different subnet, such as 192.168.149.X, etc.. Just make sure to address your client machines accordingly.

4. Select SaveAs from the File menu to name and save your configuration. Then just double click on this config file anytime you want to startup IPNetRouter with this configuration. You can put an alias to this config file in your Startup Items folder so that it runs automatically at each startup. Your gateway should now be working!!! [If it doesn't work right away, try restarting once to give Open Transport a chance to rebuild your TCP/IP stack with the supplied Proxy module inserted.]

Now each slave machine on your LAN needs to be configured to take advantage of this new gateway. The general idea is as follows:

1. Each slave machine needs its own unique IP address from the same private sequence that the gateway is using. In the above example the gateway is at

192.168.73.1, so your slaves should each use a unique addresses like 192.168.73.2, or 192.168.73.X where X is anything from 2 up to 255.

2. All the slave machines need to be told to use the gateway machine (192.168.73.1) to access the Internet. So set the Default Gateway (router) parameter of all slaves to 192.168.73.1

3. You can use the IPNetRouter IP address as the Name Server address in each of the clients (e.g. 192.168.73.1). Just make sure that the DNS forwarding option is checked in the Gateway window of IPNetRouter. (you MUST be running IPNetRouter v1.4.8 or later for DNS forwarding). Otherwise you need to enter the true Name Server for your ISP in each of the client machines.

4. **Important:**You SHOULD NOT use the DHCP server within IPNetRouter to configure your client machines when using this single ethernet setup. If you do so, your IPNetRouter gateway becomes a DHCP server for the entire cable/DSL/ADSL/DSL modem network and your Internet connection may be terminated by your service provider.

Other servers such as mail and news servers should all be set to the standard values specified by your provider (@Home). You may, however, have to use FULLY SPECIFIED DOMAIN NAMES for Mail servers when accessed from your client machines (e.g. instead of just using "mail" for the SMTP mail server, you might have to fully specify the name like "mail.srst1.fl.home.com"). Also make sure the slaves each have a unique IP address and will use the gateway (192.168.73.1) as their default gateway. On a Mac use the TCP/IP or MacTCP Control Panel, on Win95 use the Network Control Panel.

## Additional Notes:

You may need to power cycle your cable/DSL/ADSL modem if it was being used by something other than the gateway Mac previously. The safest thing to do is to turn on the cable/DSL/ADSL modem first, then the gateway Mac. Access the Internet from the gateway Mac to make sure the cable/DSL/ADSL modem learns the gateway's address before any other. After the cable/DSL/ADSL modem has seen the gateway Mac at least once, you can power up any of the devices in any order until you power cycle the cable/DSL/ADSL modem. Then you'll need to make sure it sees the gateway Mac first.

The 10Base-T Ethernet interface on cable/DSL/ADSL modems is usually designed to act like a hub port. This should attach to the uplink port on your 10Base-T hub, or you will need to use a "crossover" cable/DSL/ADSL. See Building Your Own Low Cost Ethernet for details.

If you need to determine the IP address of the default Name Server (DNS) which your

ISP has you currently using, you can use the tools in IPNetMonitor to determine this IP address. From the gateway machine, launch IPNetMonitor and open the NSLookup Window (Name Server Lookup). From the NSLookup window, open the Ping window (Cmd-T), this will invoke a ping test to your default Name Server used by NSLookup. Your default Name Server address will be visible in the Ping window. You could then use this IP address in each of your client machines... but just using the IPNetRouter address with DNS masquerading is normally much easier.

If your gateway is configured using DHCP in the TCP/IP control panel, you should check the "DHCP aware" checkbox for the corresponding IP Interface in the IPNetRouter Interfaces window. This tells IPNetRouter to use your dynamically assigned DHCP address instead of the address saved with your configuration file when IPNetRouter restores your saved settings.

**IP Addressing**: I chose the example IP addresses from network 192.168.x.x because this network range is reserved for private Internets (see RFC 1918). In order to route IP datagrams between two networks, each network must normally have its own network number (the most significant part of the IP address logically ANDed with the network mask). If you wish to follow the suggested example, each machine on your ethernet LAN must have a unique address from network 192.168.73.x (192.168.73.1, 192.168.73.2, 192.168.73.3, etc.). The Macintosh running IPNetRouter (192.168.73.1 in this example) becomes the default router or gateway for the other machines on your ethernet LAN.

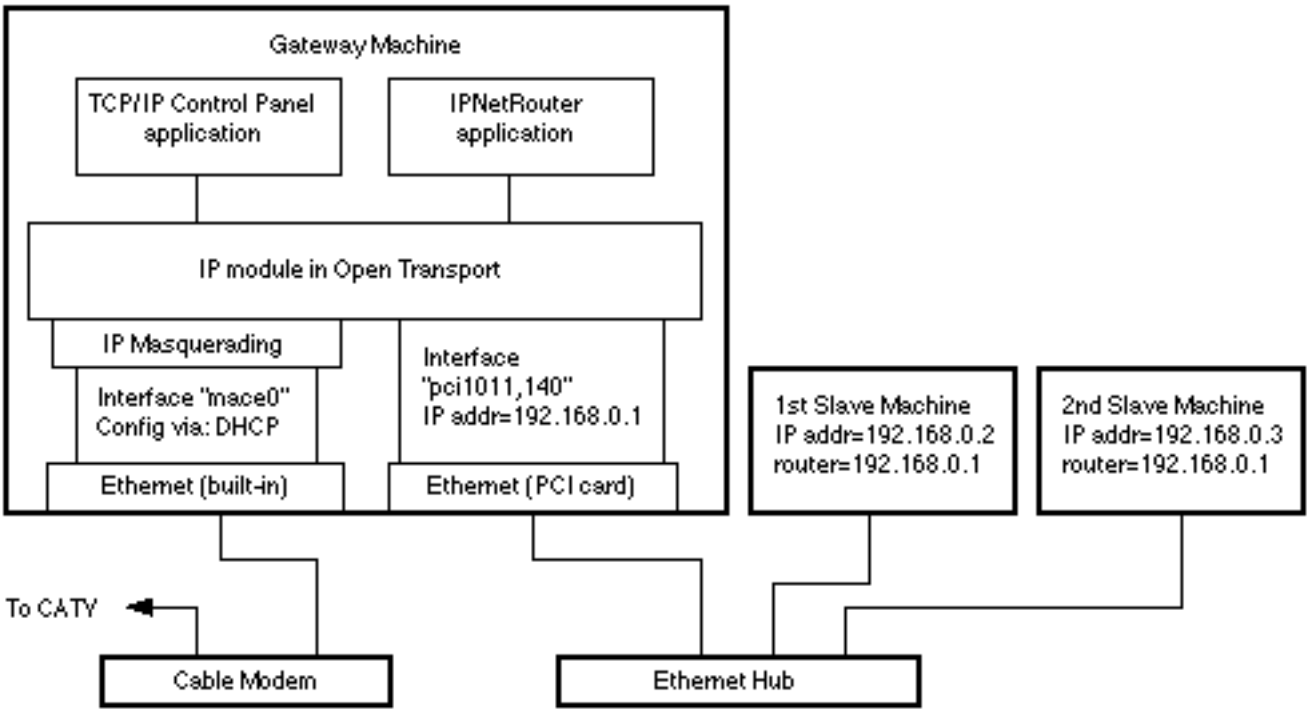
products

# IPNetRouter

IPNetRouter.PPC

Return to Getting Started

# Connecting to the Internet using a cable/DSL/ADSL modem and dual ethernet



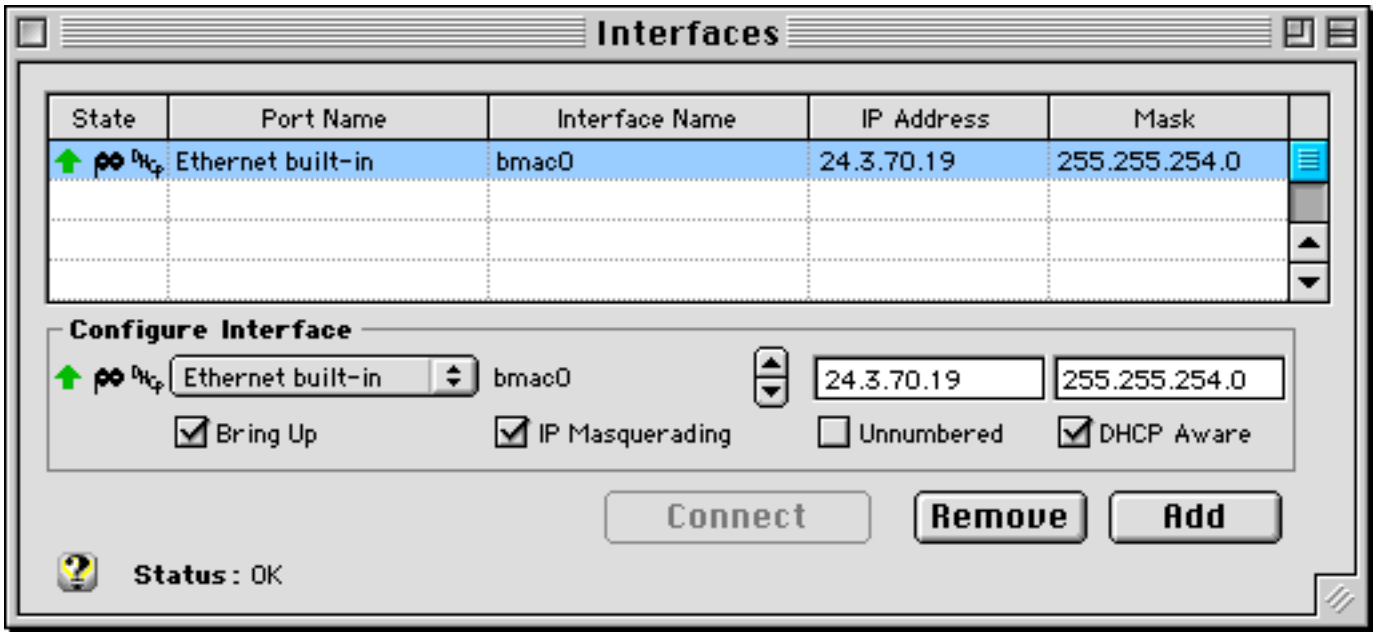

These directions assume that you normally connect to the Internet using a cable/DSL/ADSL modem, and that you have a 2nd ethernet port (card) used to connect to your LAN.

The basic steps to configure IPNetRouter to work with a cable/DSL/ADSL modem in this dual ethernet port setup are as follows:

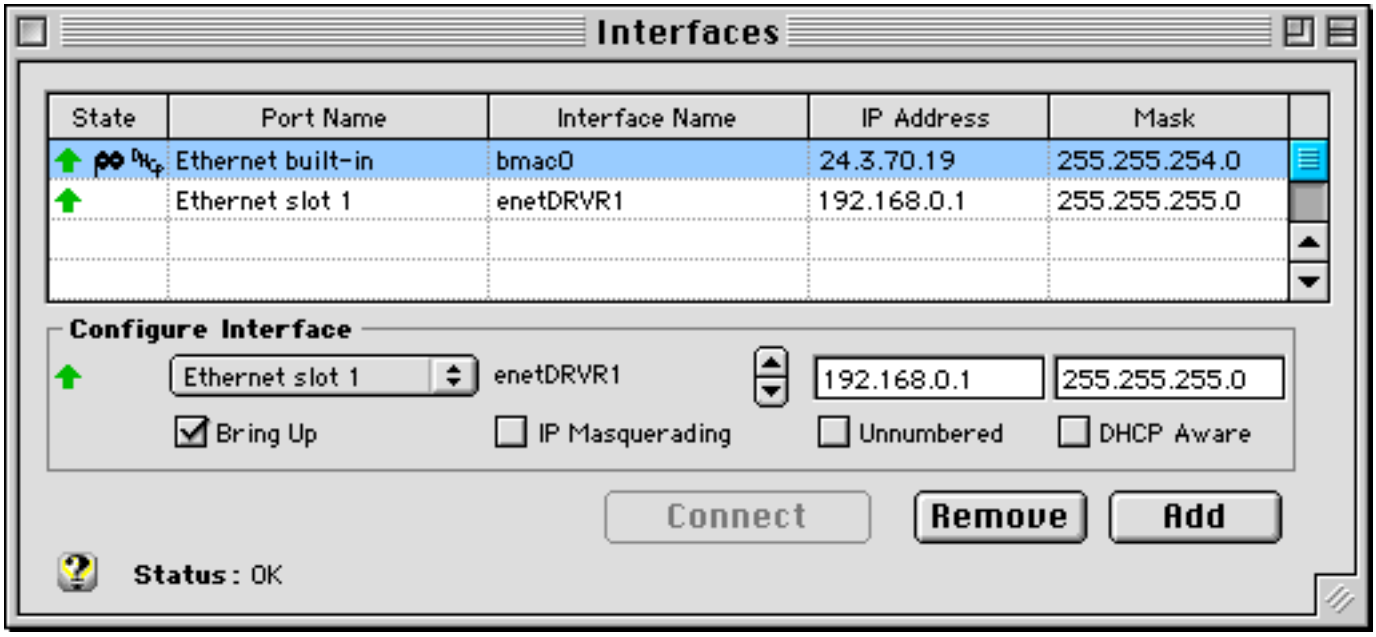
1. On the gateway machine, use the TCP/IP control panel to configure the Ethernet interface to use your cable/DSL/ADSL modem as instructed by your service provider (@Home). **Make SURE that the "Load Only When Needed" option is UNCHECKED** (you must be in Advanced User mode to get to the TCP/IP options.

Choose User Mode under the Edit menu and select Advanced). If you can already access the Internet from this machine via your cable/DSL/ADSL modem and you are SURE that the "Load Only When Needed" option is UNCHECKED in the TCP/IP control panel, you're ready to proceed to the next step.
IF you are using an older NuBus machine on which to run IPNetRouter, this cable/DSL/ADSL modem connection MUST be made into the built-in ethernet port. With PCI bus machines, the cable/DSL/ADSL modem can be plugged into either ethernet port.

2. Startup IPNetRouter on the gateway machine. In the Interfaces window, click on the top row (with your public IP address on the ethernet port). With this interface selected, check the "IP masquerading" box and then press Add (if you receive your IP address via DHCP from your cable/DSL/ADSL modem provider, then also check the the "DHCP Aware" box). You should see a little mask icon appear in the state column for the Ethernet interface (and a DHCP icon if you checked the DHCP checkbox and added this feature). Notice you need to enable IP Masquerading on the interface that communicates directly with your cable/DSL/ADSL modem. Your Interfaces window should now appear similar to the following (your IP address and mask values will vary depending upon those values assigned by your ISP):




3. Next you need to create a 2nd IP interface on your other Ethernet port. Continue to work in the Interfaces window using the "Configure Interface" input area just below the main display grid. Select the other ethernet port from the popup menu under the Port Name column. The corresponding interface name is entered for you automatically (you should not need to change this). Enter a local IP Address and Mask. We suggest 192.168.0.1 and 255.255.255.0 respectively. Only the "Bring Up" checkbox should be checked. Press Add. You should see all the values you just entered appear in a new row in the main Interfaces table.




4. Select SaveAs from the File menu to name and save your configuration. Then just double click on this config file anytime you want to startup IPNetRouter with this configuration. You can put an alias to this config file in your Startup Items folder so that it runs automatically at each startup. Your gateway should now be working!!! [If it doesn't work right away, try restarting once to give Open Transport a chance to rebuild your TCP/IP stack with the supplied Proxy module inserted.]

Now each slave machine on your LAN needs to be configured to take advantage of this new gateway. The general idea is as follows:

1. Each slave machine needs its own unique IP address from the same private sequence that the gateway is using. In the above example the gateway is at 192.168.0.1, so your slaves should each use a unique addresses like 192.168.0.2, or 192.168.0.X where X is anything from 2 up to 255.

2. All the slave machines need to be told to use the gateway machine (192.168.0.1) to access the Internet. So set the Default Gateway (router) parameter of all slaves to 192.168.0.1

3. You can use the IPNetRouter IP address as the Name Server address in each of the clients (e.g. 192.168.0.1). Just make sure that the DNS forwarding option is checked in the Gateway window of IPNetRouter. (you MUST be running IPNetRouter v1.4.8 or later for DNS forwarding). Otherwise you need to enter the true Name Server for your ISP in each of the client machines.

4. With the dual ethernet setup, you can use the DHCP server within IPNetRouter to configure your client machines. This may be easier than individually and manually addressing each of the client machines on your private network. Using the DHCP

server within IPNetRouter is quite simple: just open the DHCP window and click the DHCP Server On checkbox and click Done. There are, however, many other options you can configure with the DHCP server. Please see the Help window in the DHCP window for more information.

Other servers such as mail and news servers should all be set to the standard values specified by your provider (@Home). You may, however, have to use FULLY SPECIFIED DOMAIN NAMES for Mail servers when accessed from your client machines (e.g. instead of just using "mail" for the SMTP mail server, you might have to fully specify the name like "mail.srst1.fl.home.com"). Also make sure the slaves each have a unique IP address and will use the gateway (192.168.0.1) as their default gateway. On a Mac use the TCP/IP or MacTCP Control Panel, on Win95 use the Network Control Panel.

If you need to determine the IP address of the default Name Server (DNS) which your ISP has you currently using, you can use the tools in IPNetMonitor to determine this IP address. From the gateway machine, launch IPNetMonitor and open the NSLookup Window (Name Server Lookup). From the NSLookup window, open the Ping window (Cmd-T), this will invoke a ping test to your default Name Server used by NSLookup. Your default Name Server address will be visible in the Ping window. You could then use this IP address in each of your client machines... but just using the IPNetRouter address with DNS masquerading is normally much easier.

If your gateway is configured using DHCP in the TCP/IP control panel, you should check the "DHCP aware" checkbox for the corresponding IP Interface in the IPNetRouter Interfaces window. This tells IPNetRouter to use your dynamically assigned DHCP address instead of the address saved with your configuration file when IPNetRouter restores your saved settings.

**IP Addressing**: I chose the example IP addresses from network 192.168.x.x because this network range is reserved for private Internets (see RFC 1918). In order to route IP datagrams between two networks, each network must normally have its own network number (the most significant part of the IP address logically ANDed with the network mask). If you wish to follow the suggested example, each machine on your ethernet LAN must have a unique address from network 192.168.0.x (192.168.0.1, 192.168.0.2, 192.168.0.3, etc.). The Macintosh running IPNetRouter (192.168.0.1 in this example) becomes the default router or gateway for the other machines on your ethernet LAN.

products

# IPNetRouter


Return to Getting Started

# IPNetRouter and PPPoE Configuration Section

## Contents



## IPNetRouter and PPPoE Client/Drivers

These directions assume that you normally connect to the Internet using a DSL/ADSL modem through a PPPOE driver such as Enternet (NTS), Sympatico (Enternet), or MacPoet. If you have a single ethernet configuration then you probably will not want to use MacPoet. You can download the Enternet client software and try that if you have problems with MacPoet. Read all instructions thoroughly before beginning an installation.

We strongly recommend that you download and install IPNetRouter 1.5 or later for use with PPPoE. The following instructions cover all previous versions of IPNetRouter but the

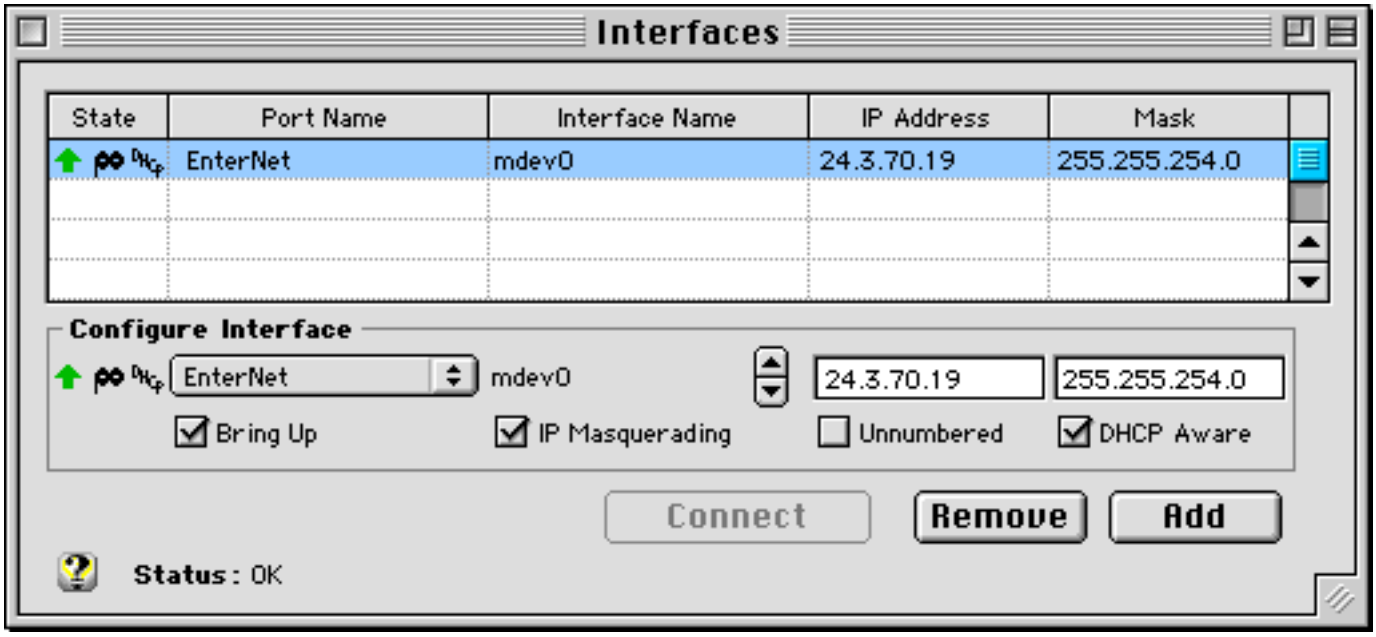
features available in 1.5 make it much easier to bring your LAN clients up for connection to the internet as it has a "Better PPPoE Routing" feature.

Although many users have dual ethernet configurations working on nubus Macs with PPPoE drivers, **we have few reports of single ethernet (one port) nubus Mac configuration working with IPNetRouter in this configuration. If you manage to get such a configuration to work, please let us know!**

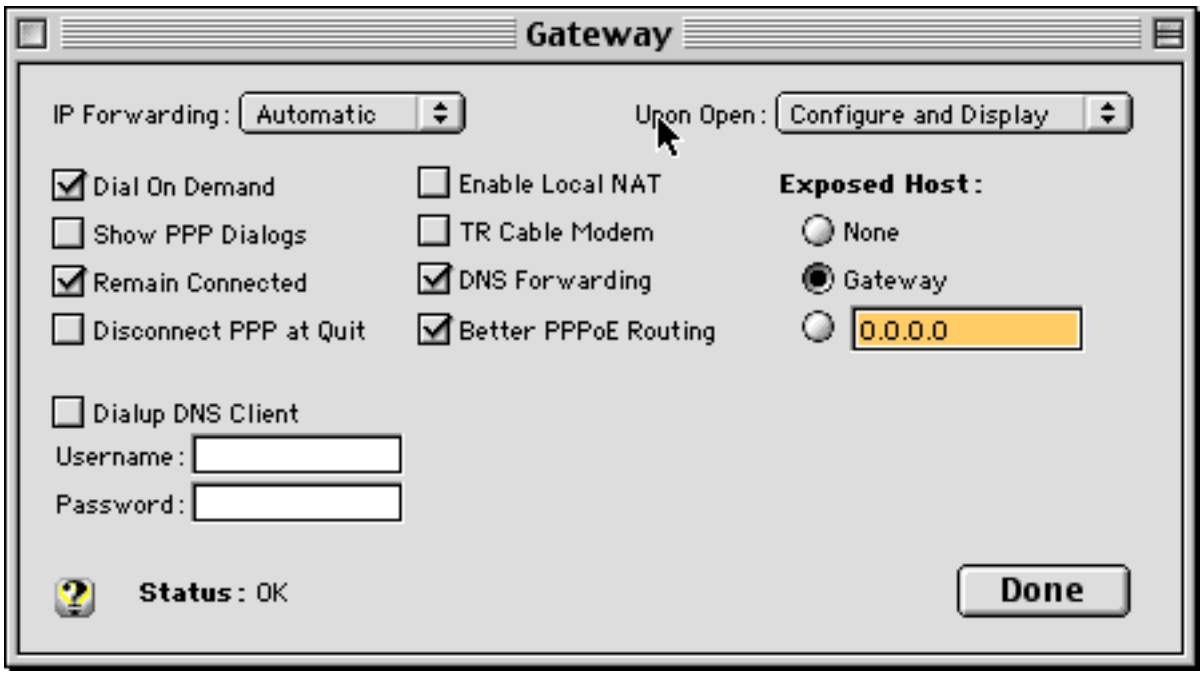
## Dual Ethernet and PPPoE installation instructions

The basic steps to configure IPNetRouter to work with PPPOE and a DSL/ADSL modem in this dual ethernet port setup are as follows:

1. On the gateway machine, use the TCP/IP control panel to configure the PPPOE driver for the DSL/ADSL modem as instructed by your service provider. **Make SURE that the "Load Only When Needed" option is UNCHECKED** (you must be in Advanced User mode to get to the TCP/IP options. Choose User Mode under the Edit menu and select Advanced). If you can already access the Internet from this machine via your DSL/ADSL modem and you are SURE that the "Load Only When Needed" option is UNCHECKED in the TCP/IP control panel, you're ready to proceed to the next step.
   IF you are using an older NuBus machine to run IPNetRouter, this DSL/ADSL modem connection MUST be made into the built-in ethernet port. With PCI bus machines, the cable/DSL/ADSL modem can be plugged into either ethernet port.
2. **Open your PPPoE client and connect to the internet before launching IPNetRouter.** The PPPoE driver must be loaded with an active connection before IPNetRouter is running. Verify that you can connect to the internet with a browser and then quit the browser. **If you do not do this first, you may experience problems configuring IPNetRouter, including crashing and hanging of your system.**
3. After connection with the PPPoE client, launch the IPNetRouter application. Verify that you can still connect to the internet with a browser and quit the browser before proceeding to the next step.
4. In the Interfaces window, click on the top row (with your public IP address on the ethernet port). With this interface selected, check the "IP masquerading" box, check the the "DHCP Aware" box, and click Add. You should see a little mask icon appear in the state column for the PPPOE interface and a DHCP icon. Notice you need to enable IP Masquerading on the interface that communicates directly with your DSL/ADSL modem through the PPPOE driver. Your Interfaces window should now appear similar to the following (your Port Name, IP address and mask values will vary depending upon the PPPOE driver you use and the IP and mask values assigned by your ISP):

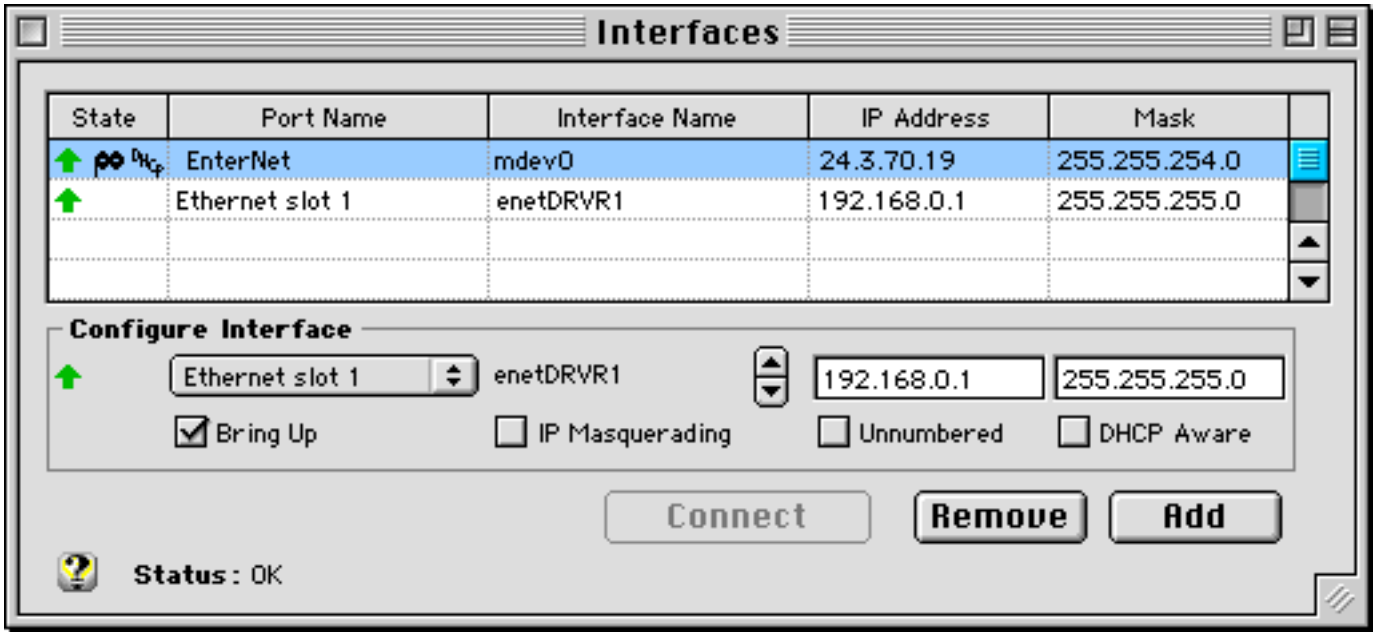



5. In IPNetRouter's Gateway window, make sure that IPForwarding is "Automatic", and that the following are checked:
   - "Remain Connected"--if the PPPoE connection times out while IPNetRouter is running, your system may hang if you try to reconnect.
   - "Better PPPoE Routing" (versions 1.5c12, c14 and later) or "Limit MTU for PPPoE" (1.5c8-11)




6. Next you need to create a 2nd IP interface on your other Ethernet port. Continue to work in the Interfaces window using the "Configure Interface" input area just below the main display grid. Select the other ethernet port from the popup menu under the Port Name column. The corresponding interface name is entered for you automatically (you should not need to change this). Enter a local IP Address and Mask. We suggest 192.168.0.1 and 255.255.255.0 respectively. Only the "Bring

Up" checkbox should be checked. Press Add. You should see all the values you just entered appear in a new row in the main Interfaces table.




7. Select SaveAs from the File menu to name and save your configuration. Launch the PPPoE client and then double click on this config file anytime you want to startup IPNetRouter with this configuration. Your gateway should now be working! If it doesn't work right away, try restarting once to give Open Transport a chance to rebuild your TCP/IP stack with the supplied Proxy module inserted.
8. See Slave Configuration below.

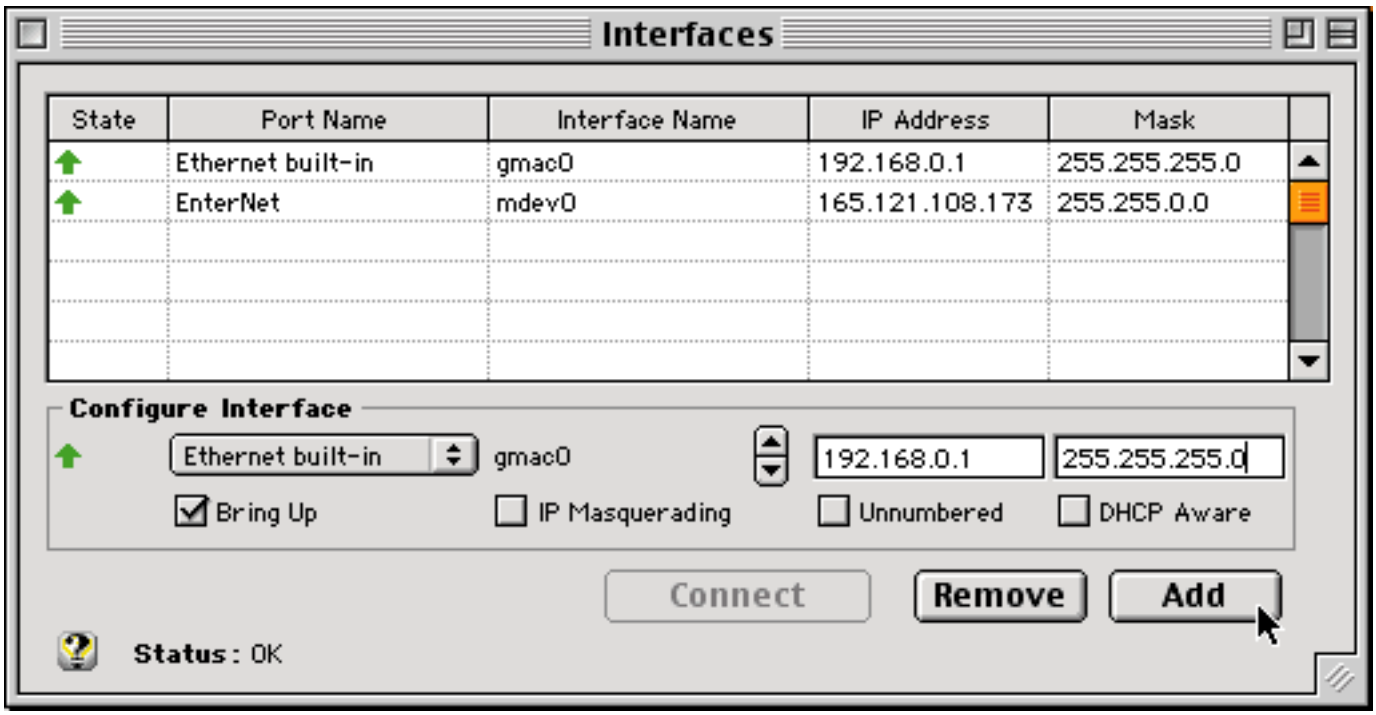
## Single Ethernet Configuration for PPPoE

For the PPPoE Single Ethernet configuration you will be following many of the steps used in the dual ethernet example above. This example works with Enternet 1.09 and presumably with Sympatico which is a derivative of the Enternet software. The following is for an iMac running OS 9 but should work on any PCI Mac and some nubus configurations:

1. Read the Introduction section. Carefully follow steps 1-5 for dual ethernet PPPoE configuration.

2. Verify that you can connect to internet with a browser and quit the browser.

3. To configure your LAN interface, create a second interface (Enternet should appear by itself initially) in the Interface window using the Ehternet built-in interface as shown in the diagram...




4. Follow steps 7-8 for dual ethernet PPPoE configuration.

## LAN Client ("Slave") Configuration With PPPoE

Now each slave machine on your LAN needs to be configured to take advantage of this new gateway. The general idea is as follows:

1. Each slave machine needs its own unique IP address from the same private sequence that the gateway is using. In the above example the gateway is at 192.168.0.1, so your slaves should each use a unique addresses like 192.168.0.2, or 192.168.0.X where X is anything from 2 up to 255.

2. All the slave machines need to be told to use the gateway machine (192.168.0.1) to access the Internet. So set the Default Gateway (router) parameter of all slaves to 192.168.0.1

3. You can use the IPNetRouter IP address as the Name Server address in each of the clients (e.g. 192.168.0.1). Just make sure that the DNS forwarding option is checked in the Gateway window of IPNetRouter. (you MUST be running IPNetRouter v1.4.8 or later for DNS forwarding). Otherwise you need to enter the true Name Server for your ISP in each of the client machines.

4. With the dual ethernet setup only, you can use the DHCP server within IPNetRouter to configure your client machines. This may be easier than individually and manually addressing each of the client machines on your private network. Using the DHCP server within IPNetRouter is quite simple: just open the DHCP window and click the DHCP Server On checkbox and click Done. There are, however, many other options you can configure with the DHCP server. Please see

the Help window in the DHCP window for more information. **You should never enable IPNetRouter's DHCP server feature in a single ethernet configuration with a cable/xDSL modem**--there is a limitation in the DHCP standard that prevents this from working correctly.

5. If you have a release of IPNetRouter prior to 1.5c8, you must use OT Advanced Tuner to set the MTU on each of your Mac client machines. Set the Maximum Transmission Unit (MTU) size to somewhere between 1390 and 1482 bytes on each of your client machines. Regedit can be used to do this on Windows clients. Skip having to do individual client configuration for MTU by checking the "Better PPPoE Routing" checkbox in the Gateway Window of IPNetRouter 1.5 or later. "Better PPPoE Routing" negotiates the MTU with your clients automatically. NOTE: The new feature was known as "Limit MTU for PPPoE" in revs 1.5c8-c11 and we highly recommend that you use the release version of 1.5 or later for your installation.

## Other Configuration Notes For PPPoE LANs

Email and Browser applications running on the client machines should have their servers (such as mail and news servers) set to the standard values specified by your provider. You may, however, have to use FULLY SPECIFIED DOMAIN NAMES for Mail servers when accessed from your client machines (e.g. instead of just using "mail" for the SMTP mail server, you might have to fully specify the name like "mail.srst1.fl.home.com"). Also make sure the slaves each have a unique IP address and will use the gateway (192.168.0.1) as their default gateway. On a Mac use the TCP/IP or MacTCP Control Panel, on Win95 use the Network Control Panel.

If you need to determine the IP address of the default Name Server (DNS) which your ISP has you currently using, you can use the IPNetRouter log window, the PPPoE client info window, or the tools in IPNetMonitor to determine this IP address. From the gateway machine, launch IPNetMonitor and open the NSLookup Window (Name Server Lookup). From the NSLookup window, open the Ping window (Cmd-T), this will invoke a ping test to your default Name Server used by NSLookup. Your default Name Server address will be visible in the Ping window. You could then use this IP address in each of your client machines... but just using your gateway's local LAN address with DNS forwarding is normally much easier.

**IP Addressing**: I chose the example IP addresses from network 192.168.x.x because this network range is reserved for private Internets (see RFC 1918). In order to route IP datagrams between two networks, each network must normally have its own network number (the most significant part of the IP address logically ANDed with the network mask). If you wish to follow the suggested example, each machine on your ethernet LAN must have a unique address from network 192.168.0.x (192.168.0.1, 192.168.0.2, 192.168.0.3, etc.). The Macintosh running IPNetRouter (192.168.0.1 in this example) becomes the default router or gateway for the other machines on your ethernet LAN. You can use a different network address in this range, 192.168.111.x or 192.168.3.x, for

example, instead of 192.168.0.x.

**See the Troubleshooting and OS 9 Compatibility page, readme and release notes documents included in your installation folder, and the FAQ page on our web site for more info on configuring your LAN with IPNetRouter.**

products

IPNetRouter


Return to Getting Started

# Connecting to the Internet using a One Way Cable Modem

One way or Telco Return (TR) cable modems use the cable system as their downstream connection but require a conventional dial-up PPP modem connection for outgoing data (upstream). This page offers a brief overview of how to configure IPNetRouter (TCP/IP) for use with these systems.

To connect a single host through a TR cable modem, you need multihoming so you can have both a PPP and Ethernet IP interface active at the same time. This is the same as connecting via PPP, except your Ethernet Interface is now being used for the downstream link. What IP address to use for your Ethernet interface is determined by your cable modem provider. Some use 192.168.0.1, others use 10.0.0.1 .

Getting a single host to work through a TR cable modem is the first step. Once this is working, the next step is to route this connection to the rest of your LAN. To get a single host to work through a TR cable modem, you can follow the instructions for **Connecting to the Internet using OT/PPP and Ethernet** except that your first Ethernet interface will be used to connect with your cable modem instead of your LAN. Here is a quick review of the steps required:

Run the IPNetRouter installer on the actual machine you wish to use it on. Be sure to use the correct version for your processor (PPC or 68K). The installer will place a copy of the IPNetRouter application on your hard drive, and a shared library named "OTModl$Proxy" in your Extensions Folder.

Next you must configure the TCP/IP Control Panel. You will need to create two separate TCP/IP configurations: one for your Ethernet; and a second for PPP (Remote Access). Your Ethernet configuration will probably be set to "Configure Manually" so that you need to specify an IP address, Subnet Mask, Router Address, and Name Server Address. I can't tell you what values to use for these since they depend on your Internet Service Provider. You will need to get this information from your cable modem ISP. Next you

will need to create a PPP (Remote Access) configuration. Press Cmd-K to bring up the configurations dialog, and create a PPP configuration named "IPNetRouter" (the name is important). This configuration may be set to "Configure via: PPP Server", but you will still need to specify a Name Server Address. If you have not connected via PPP (Remote Access) before and are not sure how this works, you may need to test this configuration. Once your PPP configuration is working, set the User Mode (under the Edit menu) in the TCP/IP Control Panel to Advanced, Choose Options, and uncheck "Load only when needed" for both your PPP and Ethernet configurations. The last step before finishing with the TCP/IP Control Panel, is to make your Ethernet configuration Active.

At this point, you are ready to test that your single host can successfully connect to the Internet using your TR cable modem. Launch IPNetRouter, in the Interfaces Window select PPP in the PortName popup menu, check "IP Masquerading", and press "Add". IPNetRouter should invoke PPP (Remote Access) to dial out, connect, and bring up a PPP interface in addition to the Ethernet interface you made active from the TCP/IP Control Panel. Try using an Internet application like your web browser to verify whether this is working (if not, see the Troubleshooting Tips and other instructions). Once this is working, the next step is to route this connection to the rest of your LAN.

To do this, you will create another IP interface on the same or another physical Ethernet to communicate with the rest of your LAN (as described in the instructions for a conventional 2-way cable modem). Of course this interface needs to have a different network number from the interface used to receive data from the cable modem. If your first Ethernet interface uses an IP Address in the range 192.168.0.x, you can't use an addresses from this range for your second Ethernet interface. You could use 192.168.100.x instead for example.

Up to this point, we have simply combined the instructions for using PPP and using a cable modem. Since we are sending upstream data via the PPP interface, we enable IP masquerading on this interface.

The tricky part is that we need to reverse the NAT (Network Address Translation) process for packets received on the first Ethernet interface as opposed to the PPP interface where masquerading has been enabled. IPNetRouter provides a switch called "TR Cable Modem" for this purpose (refer to the help text for the Gateway Window). It enables NAT for packets received on other interfaces.

Notice you may need to restart once to give Open Transport a chance to build your Ethernet link stream with the supplied OTModl$Proxy shared library. I recommend that you install IPNetMonitor available from our website and verify that you can monitor IP traffic on your Ethernet interface using the Monitor Window. This insures OTModl$Proxy has been successfully inserted into your Ethernet link stream.

The last step is to configure any hosts on your LAN to use the machine running IPNetRouter as their Internet Gateway. You do this by giving them IP Addresses in the same network range as the second Ethernet IP interface you created above, and setting

their Router Address to the IP Address of your second Ethernet interface on your gateway. You will also need to specify an appropriate Subnet Mask and Name Server Address for these machines (use the same Name Server Address as for your gateway machine that should have been given to you by your ISP).

That's it. You are ready to enjoy surfing the web from any machine on your LAN using your One Way Cable modem.

Peter Sichel
April 12, 1999

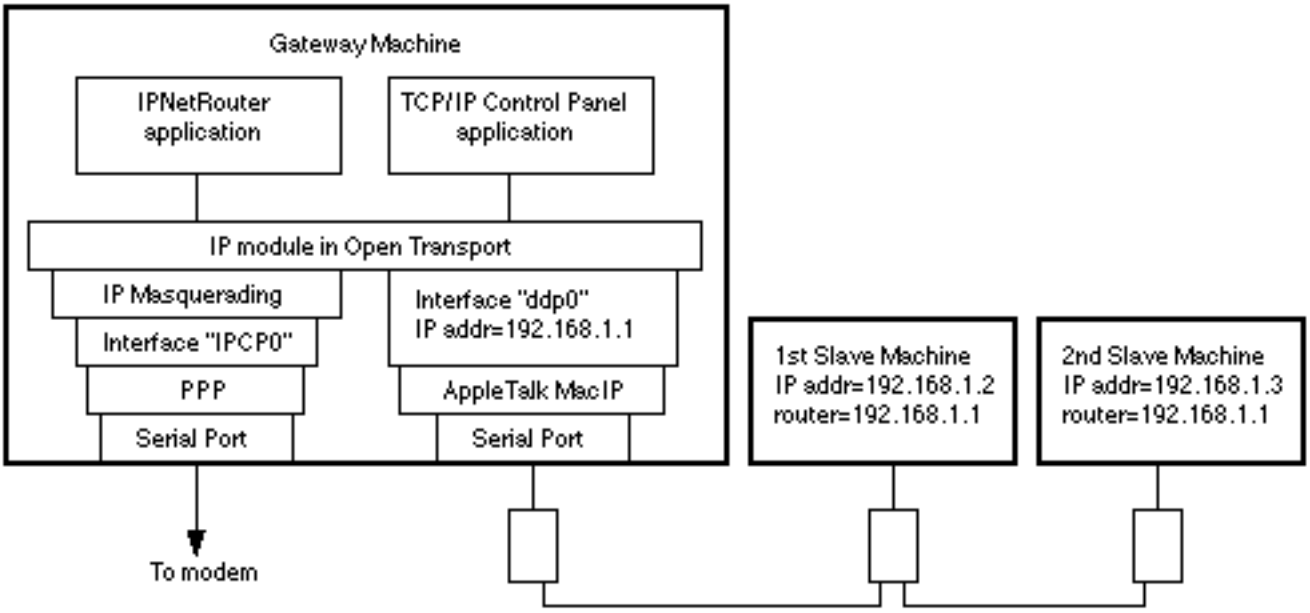
---

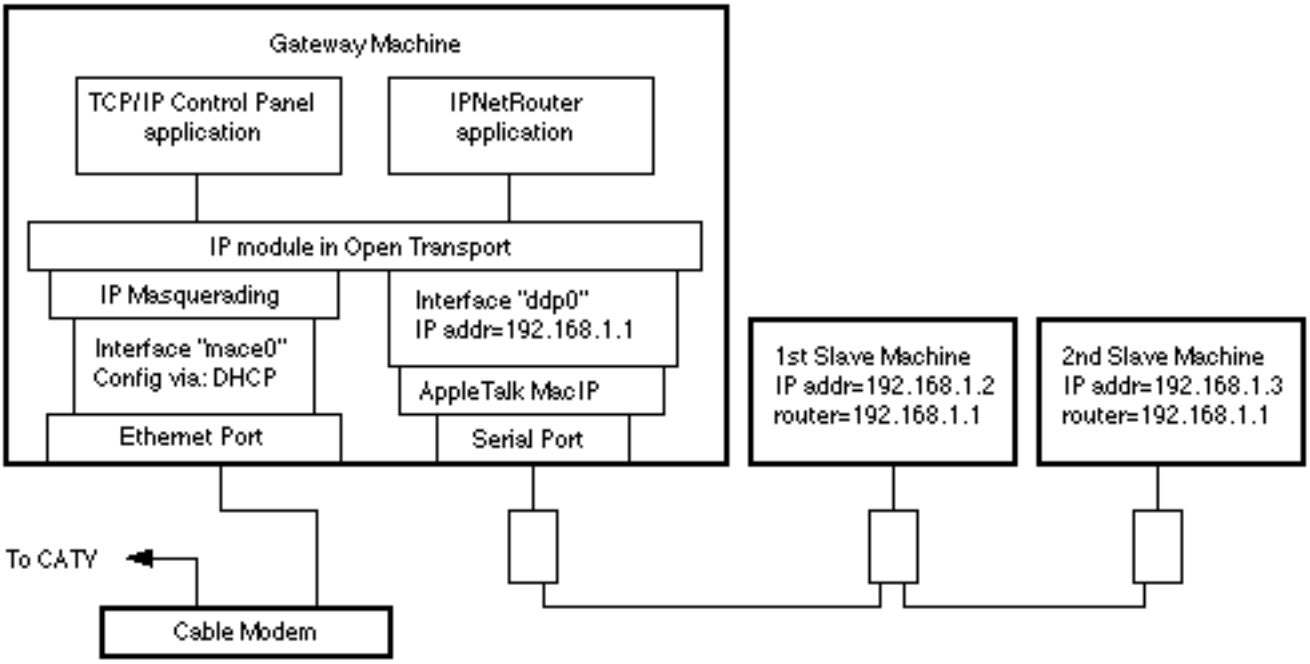
products

# IPNetRouter


Return to Getting Started

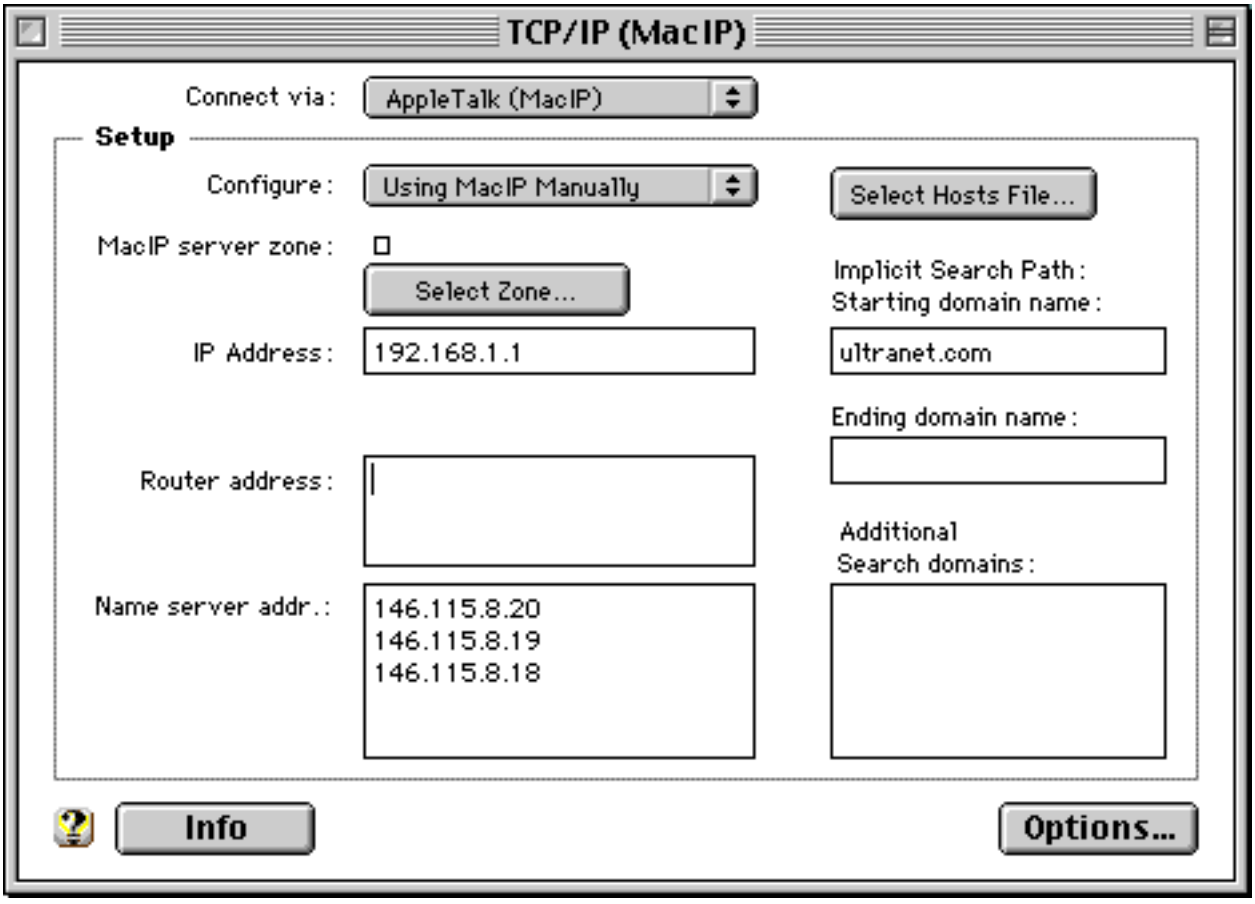
## Using LocalTalk (MacIP over AppleTalk)



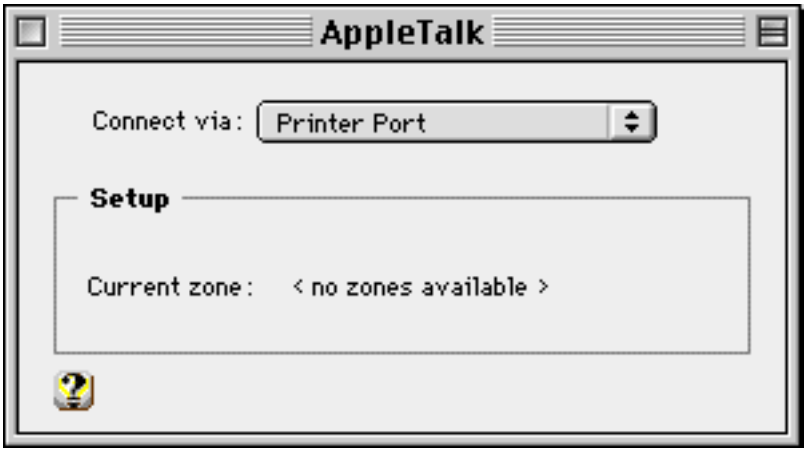



These directions assume that you will connect to your Internet Service Provider using OT/PPP or Ethernet and want to share this Internet connection with one or more machines via LocalTalk using MacIP (TCP/IP encapsulated in AppleTalk).

First, since IPNetRouter does not provide a full MacIP Gateway at this time, you must set each MacIP interface to "Configure: Using MacIP Manually" (do not select "Configure: Using MacIP Server") in the TCP/IP control panel.



Second, since we are encapsulating TCP/IP in AppleTalk to access machines on a LocalTalk network, you must set AppleTalk to connect to your LocalTalk network. If your LocalTalk network is connected to the printer port of the gateway machine for example, you would set the AppleTalk control panel to "Connect via: Printer Port".




At this point, there are two cases to consider:

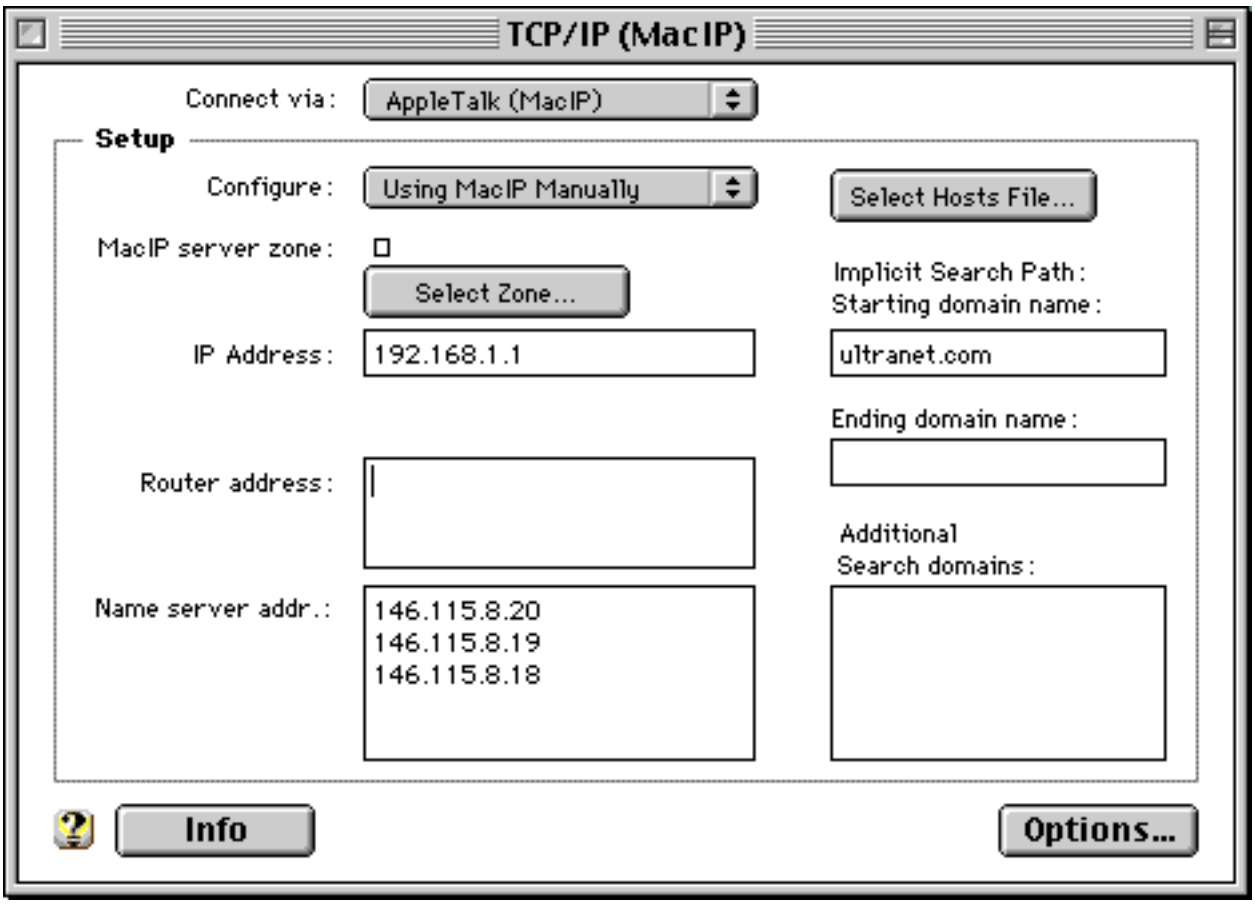
1. You need to configure a MacIP interface using the TCP/IP control panel.
2. You need to configure a MacIP interface using IPNetRouter.

The first case would arise if you are connecting to the Internet using PPP and do not want to have PPP as the primary interface selected in the TCP/IP control panel. This is important to prevent PPP from making TCP/IP unavailable when the PPP connection closes. In this case, you would select "Connect via: AppleTalk (MacIP)" in the TCP/IP control panel and configure this interface as described below. Refer to the instructions for "Connecting to the Internet using OT/PPP and Ethernet" for details on how to setup an OT/PPP interface using IPNetRouter.

The second case would arise if you are connecting to the Internet via Ethernet using a cable modem and your Ethernet interface needs to get its IP Address using DHCP. In this case, you must use the TCP/IP control panel to configure your Ethernet interface so you can specify "Configure: Using DHCP Server". Since the TCP/IP control panel is being used to configure Ethernet, you will need to configure your MacIP interface using IPNetRouter.

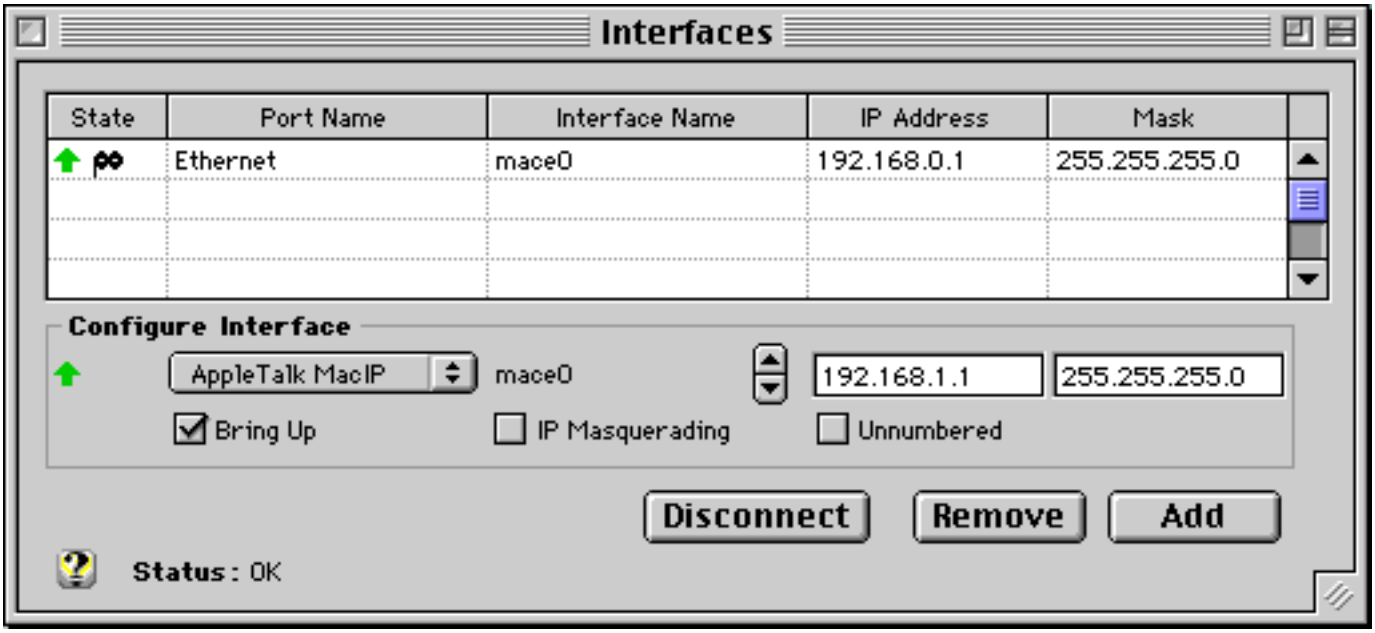
In either case, your LocalTalk network segment needs to have its own range of IP addresses with a unique network number. To follow the addressing scheme of our other examples, we can use IP addresses of the form 192.168.1.x where "x" is a number from 1-254, and use 255.255.255.0 as our network mask. We could assign the MacIP interface on the gateway machine running IPNetRouter 192.168.1.1 for example. Any other machines on our LocalTalk network would be assigned similar addresses ending in .2, .3, .4, etc. so that each machine had a unique IP address. Since the LocalTalk IP interface on our gateway machine is assigned 192.168.1.1, any other machines on our LocalTalk network would use this as their IP router address.

To configure our LocalTalk interface using the TCP/IP control panel, we could enter the corresponding information as follows.

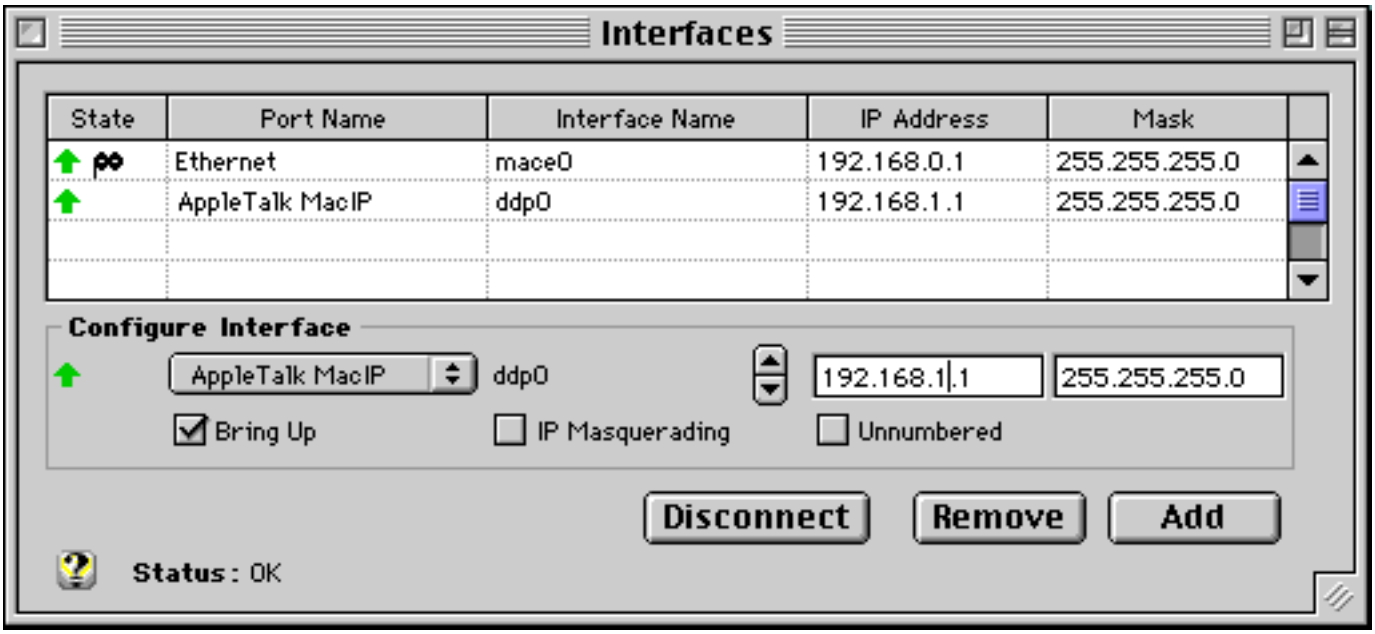



Notice the router address specified in the TCP/IP control panel normally needs to be an IP address on this network, but since we are actually connecting to the Internet via another IP interface (PPP), this router address would just need to be replaced with a "Default-Gateway" (router address) for our PPP interface. Therefore we can just leave the "Router address" empty for now.

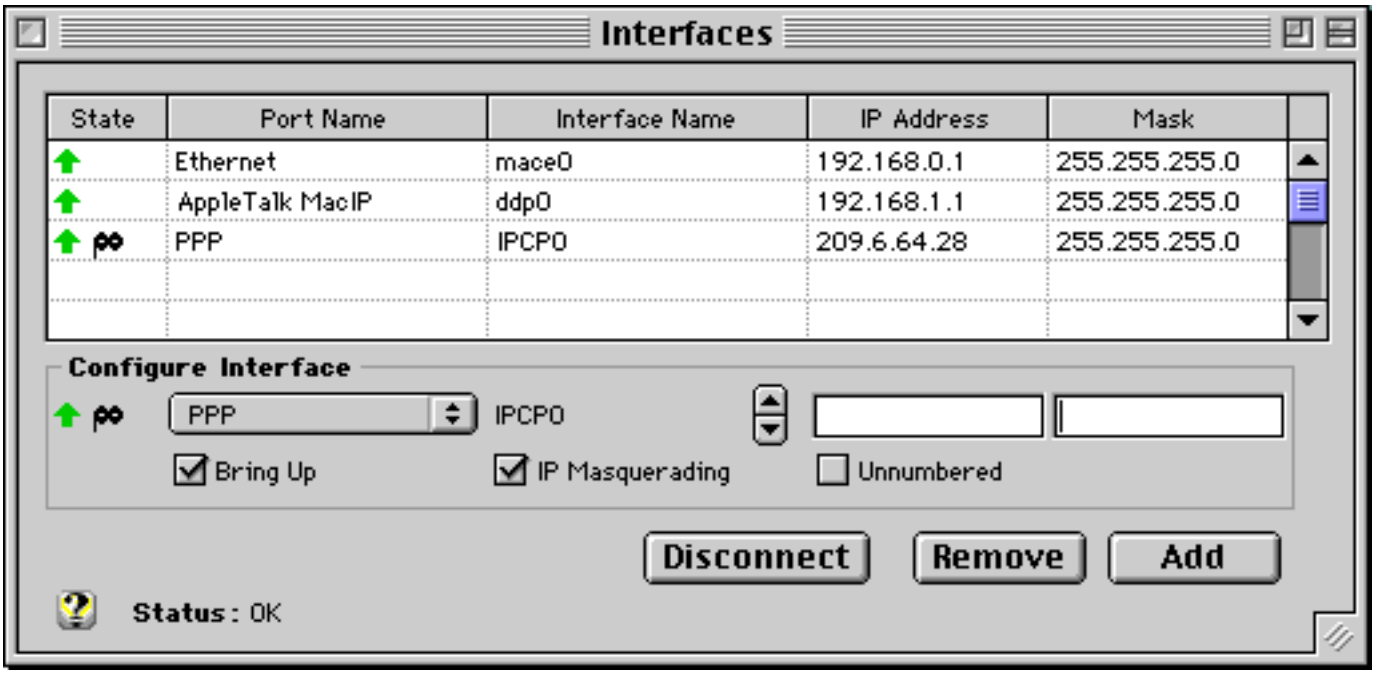
To configure our LocalTalk interface using IPNetRouter, we can select "AppleTalk MacIP" in the PortName popup menu, select a unique interface name (ddp0), and enter the desired IP address and subnet mask directly.

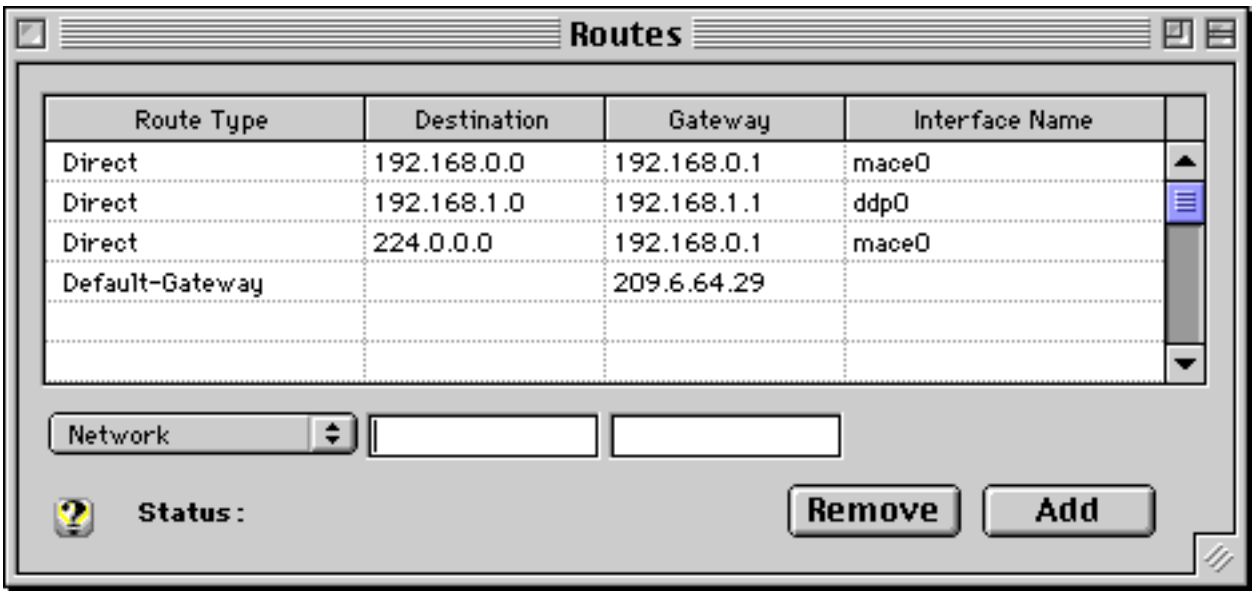


Then press "Add" to bring up this interface.




If you are connecting to the Internet using PPP or a second Ethernet interface, it is possible to have both a local Ethernet network and a LocalTalk network. In this case each IP interface on our gateway machine running IPNetRouter has a unique IP address. Following the previous examples, the Ethernet interface is at 192.168.0.1, and the LocalTalk interface is at 192.168.1.1 . Other machines on our Ethernet network will use 192.168.0.1 as their router address, while machines on our LocalTalk network will use 192.168.1.1 as their router address. The correct router address depends on which network a machine is connected to.






In the Routes window above, we have removed the router address (Default-Gateway) that was defined in the TCP/IP control panel (if any), leaving only one Default-Gateway for the PPP interface. This "pseudo gateway" was created automatically by IPNetRouter when the PPP interface was added (since the IP address and mask fields were empty). The pseudo-gateway address is adjacent to the PPP address in the Interfaces window.

Finally, each machine on your LAN (Local Area Network) needs to have a valid Name Server address. This can be the same for all your machines and simply point to one or more Name Servers provided by your ISP.

---


Return to Getting Started

# Using the IPNetRouter DHCP Server

The DHCP window allows you to configure the DHCP server built-in to IPNetRouter. In order to access the Internet, each computer on your LAN needs to be configured with four pieces of information (IP Address, Network Mask, Router Address, and Name Server Address). The DHCP server in IPNetRouter allows other computers on your LAN to get this information automatically from IPNetRouter by selecting "Configure via DHCP" in the corresponding TCP/IP control panel.

Using DHCP (Dynamic Host Configuration Protocol) avoids the need to manually configure each computer on your LAN before it can be used, and is especially useful if you have many computers on your LAN or frequently move computers from one network to another.

Before you can use DHCP to configure other hosts on your LAN, the DHCP Server itself needs to be configured. In most cases, you can simply press the "Use Defaults" button and enable the server by selecting "DHCP Server On". These settings assume your LAN will use IP addresses from the range 192.168.0.x and that your gateway running IPNetRouter will use 192.168.0.1 as its local address. [*WARNING* If you are connecting using a cable modem and single Ethernet, you must not enable DHCP on this Ethernet since DHCP requests are broadcast to all interfaces on that physical network.]

If your Local Area Network has additional requirements, you can use the DHCP Window to control what IP addresses to assign, to whom, for how long, and other operational parameters. The information in the DHCP Window is divided into four tables or panels which are briefly described below. For more information on DHCP, refer to RFC-2131, RFC-2132, or a good book on TCP/IP.

## STATUS TABLE

This table keeps track of Lease Bindings. What addresses are currently leased, to whom (hardware address or clientID), and when the lease expires. It also keeps a state value for each entry so it can track bindings that are not currently active (such as bindings that

have been offered, released, or expired). The intent here is to remember previously assigned leases so clients will always get the same lease whenever possible. Finally, this table keeps track of when each entry was last updated so the server can re-use the oldest unused entry first ("Least Recently Used"). This table should normally be viewed as "Read Only", but you can use the button on the right to remove an obsolete lease binding. You can also copy rows to the clipboard as text for use in other panels.

The Status Table is stored in a separate file in the Preferences Folder called "DHCP Server Status". Any time a lease is granted to a client, the corresponding entry is written out to this file (committed to non-volatile storage). If the server is stopped for any reason, it can recover all the current lease information from here. You can also move your DHCP Server to another machine by copying this file and the corresponding IPNetRouter configuration file.

## STATIC CONFIG TABLE

This table keeps track of any static address assignments the administrator wants to create. The fields are: (1) Network Interface - the IP address of which interface a request must arrive on [If a host moves from one network to another, we need to assign a different IP address depending on which network it is attached to. If a BOOTP relay agent is used, we match this network as well.]; (2) Lease Address - the address to grant; (3) What host or client this address is reserved for as specified by the Ethernet hardware address or ClientID. The ClientID will override the hardware address if specified in the table. The entries in this table can be edited in place by clicking in a cell. Use the buttons on the right to insert or delete entire rows. You can use Tab, Shift Tab, Return, and Shift Return to move the selected cell.

The Static Config Table is stored as part of your IPNetRouter configuration when you save from the file menu.

## DYNAMIC CONFIG TABLE

This table keeps track of ranges of IP addresses that can be automatically assigned to hosts as needed and later reclaimed. The fields are: (1) Network Interface - the IP address of which interface a request must arrive on [Each network interface that accepts DHCP requests will normally assign IP addresses from a different range.]; (2) Starting Lease Address - start of range used for address pool; (3) Ending Lease Address - end of range used for address pool. The entries in this table can be edited in place by clicking in a cell. Use the buttons on the right to insert or delete entire rows. You can use Tab, Shift Tab, Return, and Shift Return to move the selected cell.

There are no restrictions on how many address ranges can be defined to serve from multiple interfaces or multiple ranges on a single interface. Any IP address in a dynamic address pool that overlaps with an address in the Static Config Table will be reserved for static configuration and not dynamically assigned. The Dynamic Config Table is stored as part of your IPNetRouter configuration when you save from the file menu.

Some clients may use BOOTP, an older subset of DHCP. BOOTP clients are normally not eligable to receive dynamic IP addresses because BOOTP does not provide any mechanism to reclaim an expired address. To avoid requiring the network administrator to create a static assignment for each BOOTP client, IPNetRouter will try to assign BOOTP clients a dynamic address if no static configuration is found and display a warning message in the log window. In order not to lose addresses permanently, the server will try to reclaim "BOOTP dynamic addresses" after 60 days.

## LEASE OPTIONS TABLE

This table stores the other information that is given out to clients based on what network they are attached to. The fields are: (1) Network Interface - the IP address of which interface a request must arrive on [Each network interface that accepts DHCP requests can potentially give out different information.]; (2) Network Mask - the network mask for this network; (3) Routers - the router addresses for this network; (4) Name Servers - the name server addresses for this network; (5) Default Lease Time - lease time granted if none is explicitly requested; (6) Maximum Lease Time - the maximum lease time to grant regardless of what is requested; (7) Domain Name - A default domain clients can use for DNS lookups; (8) Enable DHCP on this interface - DHCP serving can be selectively enabled on a per interface basis.

The Network Interface popup menu lists the IP interfaces which have a lease options entry currently defined. It selects which row of the table to view in the rest of the panel. If you select a new interface, IPNetRouter will try to fill in appropriate values for the subnet mask, router, name server and default lease times. The Lease Options Table is stored as part of your IPNetRouter configuration when you save from the file menu.

## MORE ABOUT DHCP DATA AND EDITING

When you edit data in place, you are only editing a visible copy of that data on the display. You can undo any changes you make by pressing "Restore". To have your changes take effect, you must press "Apply". To save your settings to an IPNetRouter configuration file, you must still select "Save" from the File menu.

DHCP should be seen as a mission critical service since hosts on your LAN may not be able to use the Internet if your server is unavailable. DHCP is designed so that more than one server can be configured to provide the same information for reliability. Clients normally try to renew an existing lease and only the Server with a record of that lease will respond. If the existing lease cannot be renewed, the client may then try to obtain a new lease. The Server pings any new proposed lease address before offering it to a client to check that the address is not already in use. If the Server discovers an address conflict, it will mark the corresponding address in the Status Table as "In Use" or "Declined" and no longer try to assign it. You can see when the conflict was detected from the "Last Update" column in the Status Table. In order not to loose addresses permanently, the server will try to reclaim "In Use" or "Declined" addresses after 60 days.

Since the Status Table is kept as plain text in the file "DHCP Server Status", you can edit

this table manually if desired.

## Related References

DHCP and Mac OS

DHCP FAQ

---


Return to Getting Started

# Using an Unnumbered Interface

The "Unnumbered" check box may be used for point-to-point interfaces for which no unique local address is provided. If you have been assigned a block of IP addresses by your ISP for example, you would select "Unnumbered" to give your PPP interface an IP address from the same range assigned to your LAN (an "unnumbered" interface).

## Let's do an example

Suppose your ISP has assigned you 192.160.0.16-192.160.0.31, a range of 16 addresses. Your network number is 192.160.0.16/28 (prefix length 28) or your subnet mask is 255.255.255.240

Your gateway needs two IP interfaces:

1. An Ethernet interface on your LAN
2. A PPP interface connected to your ISP

For the Ethernet interface, we might use 192.160.0.17/28 (mask 255.255.255.240). For the PPP interface, we could use 192.160.0.17/24 (mask 255.255.255.0).

Because we are using an IP address from the same network range as our Ethernet LAN, we need to mark this as Unnumbered to tell Open Transport it is not unique. Normally in order to route between two IP interfaces they must have different network numbers yet we are assigning the same address to two different interfaces. Point-to-point interfaces are a special case because the far end of the link can already be identified so it doesn't have to have a unique IP address.

Next, since we specified our PPP address explicitly, we also need to specify a Default-Gateway (in the routes window).

Your ISP might have already told you a Default-Gateway or router to use, but you can also use a "pseudo-gateway" you make up yourself. The reason is that this gateway is just a place holder for the other side of the PPP link. Anything sent to this gateway is just

sent to the other side of the PPP link. It doesn't matter if there is really a physical gateway with this IP address or not. [When you tell PPP to get its IP address from the server, IPNetRouter sets up a pseudo Default-Gateway for you automatically.]

According to the standard, a gateway must be in the same subnet as your network interface. This is why we made the subnet mask for our PPP interface 255.255.255.0 (prefix length 24 instead of 28). We need to assign a gateway that is within this subnet, but not within your 16 address range (so IP can route to it).

A good choice would be 192.160.0.254 [We don't want to use the all zero or all one's host number]. To enter this default route in the routes window, specify the Route Type as "Default-Gateway", leave the "Destination" field empty (to match any address), and enter 192.160.0.254 in the "Gateway" field. Then press Add to update the actual Routes in the table above.

Finally, your ISP must configure their routers to route datagrams for your network range (192.160.0.16-192.160.0.31) to your gateway machine (192.160.0.17).

This sets up IPNetRouter for conventional subnet routing without IP masquerading (Network Address Translation). Since you are not using NAT, this setup is transparent to all TCP and UDP traffic.

---

You can also use the "Unnumbered" feature with IP masquerading to give your PPP and Ethernet interface the same IP address.

Suppose you have a static IP address from your ISP (just one), and want to run WebStar on your gateway machine. Since WebStar uses the IP address in the TCP/IP control panel, if you use a private address for your Ethernet interface (like 192.168.0.1), hosts on the public Internet won't see it.

We can fool WebStar into using the correct IP address by setting the TCP/IP control panel to use our static IP address for the Ethernet interface. Next we use IPNR to setup our PPP interface with the same IP address and set "Unnumbered" to tell OT it isn't unique. Here's the tricky part. Since we don't have public IP addresses to assign to the other machines on our LAN, we'll create a second IP interface on our Ethernet with a private IP address (192.168.0.1) as in the Getting Started examples.

IP will use this private IP interface to talk to our LAN, will use our PPP interface to talk to the public Internet, and we conveniently put our public IP address in the TCP/IP control panel where WebStar can find it.

---


Return to Getting Started

# Inbound Port Mapping

IPNetRouter provides a feature called "IP Masquerading" that allows you to connect multiple hosts to the Internet using a single user ISP account and single public IP address (that may be dynamically assigned). This saves you the cost of setting up multiple accounts with your ISP, or getting a block of globally unique IP addresses assigned that your ISP must route to (and will charge you for).

IP Masquerading works using a technique called "Network Address Translation" (NAT) with extensions for "Port Multiplexing" and "ICMP Translation". The basic idea is to disguise packets to or from your private LAN to make it appear to the rest of the Internet as if they are only communicating with your gateway machine. The Network Address Translator is implemented as a module that sits between your gateway machine and the public Internet. When the Translator sees a packet from your LAN that doesn't originate from the gateway, it changes the source IP address of that packet to the IP address of the gateway machine. In order to keep track of which host on your LAN the packet came from, it also assigns it an unused protocol port number on the gateway machine. Whenever a response is received at that port on the gateway, the Translator knows to reverse the process substituting the original IP address and port number on your private LAN.

The effect is to hide multiple hosts behind a single public IP address. Notice it is the action of sending a packet from the LAN to a host on the Internet that creates a return translation entry. Until a host on the LAN initiates contact, it is invisible to the public Internet. It has no public IP address and traffic addressed to the gateway machine is not forwarded to the LAN since there is no reverse translation entry. It passes through the NAT module unchanged so remains addressed to the gateway. IP Masquerading acts like a firewall allowing you to access the public Internet from inside your LAN, but not the other way around.

This arrangement is ideal for surfing the web from your LAN since each time your web browser requests a page, it creates a reverse translation entry allowing the requested

web page to be returned through the NAT gateway. Suppose you want to run a server on your private LAN and make it visible to the public Internet. You could run all your servers on the gateway machine which is visible to the public Internet, but this might not be the most convenient way to arrange your network. In order to allow servers on your private LAN to be visible to the public Internet, we need a way to create a reverse translation entry without initiating the connection from your private LAN. This is the purpose of "Inbound Port Mapping".

Inbound Port Mapping allows you to manually create a permanent (or static) translation entry that maps a protocol port on your gateway machine to an IP address and protocol port on your private LAN. TCP/IP uses the term "Endpoint" to refer to the combination of an IP address and protocol port. Every TCP/IP connection exists between two Endpoints. We use the term "Apparent Endpoint" to refer to the endpoint on your gateway machine that the rest of the Internet sees, and the term "Actual Endpoint" to refer to the Endpoint on your LAN that forms the local end of a connection.

To specify an inbound mapping, open the Port Mapping window with IP masquerading enabled. If you uncheck "Show Static Only" and press Monitor, you will be able to see the NAT table in real time (for any connections in progress). The window help describes how to add or remove your own mapping entries.

Suppose you have three machines and want to set up a web server on your LAN as follows:

192.168.0.1 (Gateway machine, IPNR)
192.168.0.2
192.168.0.3 (Web server)

The apparent address (what the Internet sees) is the public IP address of your gateway (the floating IP number assigned by your ISP). This number will be filled in automatically for you in the Port Mapping window. The Private address is the address of your web server on your private LAN (192.168.0.3). The port number is whatever port is used for your server. Web servers usually listen on port 80.

In the Port Mapping window, you would enter the following in the "specify entry" area:

Protocol: TCP
Apparent Endpoint
Address: leave as already filled in
Port: 80
Actual Endpoint
Address: 192.168.0.3
Port: 80
Static: checked

Then press the "Add" button to update the actual record in the NAT table.

# Using DNS from your LAN

If you set IP Forwarding to Automatic in the Gateway Window, the NAT configuration is as follows:

Internet<--->NAT<--->Gateway<--->LAN

In this configuration, only hosts on the public Internet can access a server on your LAN using its public IP address on the gateway. If the LAN tried to access your server via its public address, the packet would be delivered to the gateway without being translated.

If you check "Enable Local NAT" in the Gateway Window, the configuration becomes:

Internet<--->NAT<--->Gateway<--->NAT<--->LAN

In this configuration, if a host on your LAN tries to access a server on your LAN using its public (port mapped) IP address, the packet is translated before it gets to the gateway so the gateway will forward it back out the same interface it arrived from. This allows you to use the same DNS names to access your servers behind the gateway regardless of where you are. This feature is not available in cable modem and single Ethernet

configurations because NAT is disabled on the private network used to communicate with your LAN (so it can share the same physical Ethernet).

# For more Information

The basic concept of Network Address Translation (NAT) is described in RFC-1631 (a widely available public specification). IPNetRouter extends this by adding "Port Multiplexing". In addition to translating between private and public IP addresses, IPNR translates multiple private addresses to a single public address by assinging unique port numbers to each protocol stream. You can read more about this by searching for "Network Address Translation" and "Port Multiplexing" on the web.

---

Return to:

- Start of Guide
- IPNetRouter Troubleshooting Tips
- Sustainable Softworks home page

products

IPNetRouter

IPNetRouter.PPC

Return to Getting Started

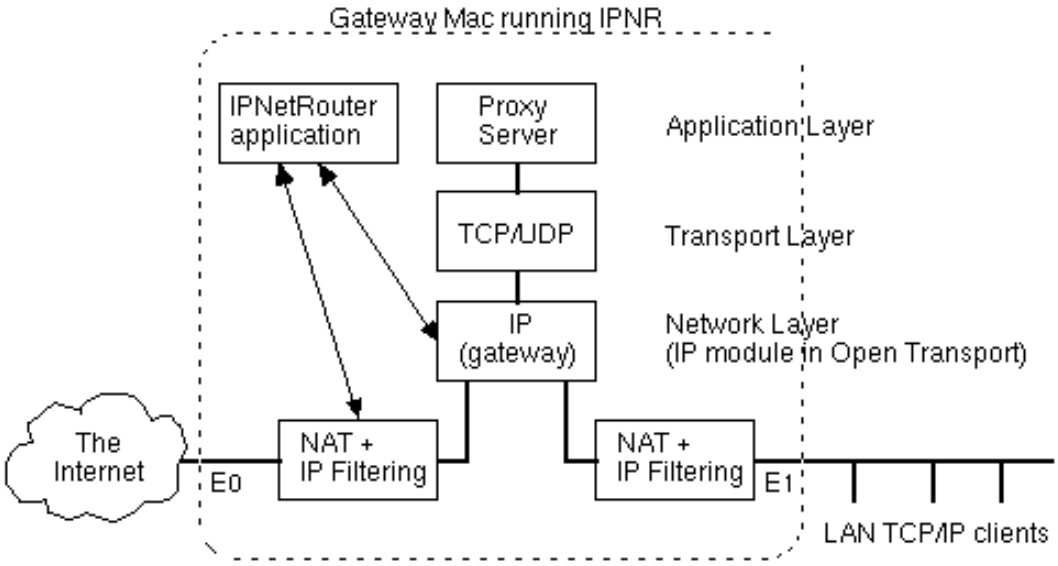
# Building Your Own Low Cost Firewall

We've all heard the term "firewall" to describe making a network more secure, but what does this mean? Aren't Internet firewalls supposed to be complicated and expensive? Not at all! This page explains the basic firewall concept and how you can use IPNetRouter to implement a completely "bullet-proof" firewall.

"Firewall" is actually a generic term for a mechanism that enforces an access policy. The basic building blocks of network firewalls are **IP routers with filtering** and **proxy servers**.

A "classic" firewall configuration uses two routers with a "Bastion Host" between them. One or more proxy servers on the Bastion Host selectively authenticate and log specific requests. Since IPNetRouter runs as software on a Macintosh, you can combine the routing, filtering, and proxy function in a single machine.

Consider a Macintosh with two Ethernet interfaces (E0 and E1). IPNetRouter provides full IP filtering on each Ethernet interface, and routing between them. Here's a simple block diagram:




IPNetRouter provides one kind of built-in proxy service: Network Address Translation (aka NAT or IP

Masquerading). NAT prevents access from the public Internet to your private LAN, while still allowing clients on your private LAN to access the public Internet. **By itself, and with NO filters or port mapping entries added, IPNetRouter completely protects your CLIENT machines from outside access.** (To understand how Network Address Translation works, refer to the Inbound Port Mapping page. ) For the gateway machine itself we strongly recommend that you do NOT run any servers or server type services (e.g. web sharing) UNLESS it is your intention to provide public access to these services.

In summary, just by running IPNetRouter as a NAT router, your network becomes very secure and you actually have quite a solid firewall in place. **There is, however, the possibility of unauthorized and remote use of your gateway machine as a router or some other machine pretending to be your router.** To prevent this we use packet filters.

---

## A Full Firewall for users with a Static IP Address

Let's assume that:

- We have a static IP address for our gateway machine.
- Only the gateway machine will have access to BOTH the Internet and our private LAN and it is running an email server.
- Our client machines will access the gateway machine to retrieve/send email but NOT have access to the Internet.
- There will be no outside access from the Internet to any of our client machines.
- On the gateway machine, our Internet connection is via the built-in ethernet port and our private LAN is connected to the slot 1 ethernet port.
- We want to be absolutely sure that ONLY our designated client machines will be able to access our gateway machine(e.g. prevent IP spoofing ).

To create a firewall for this situation we must use filters.

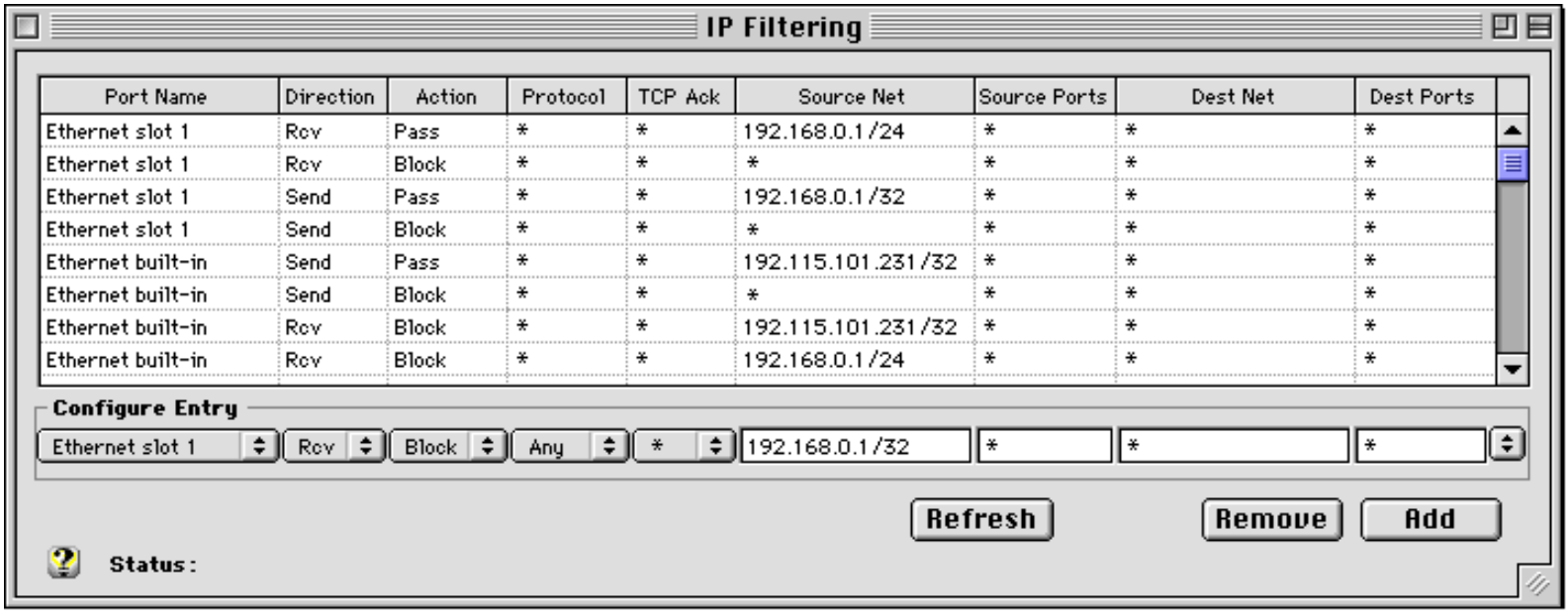
Starting from the block diagram above we would configure the IP filters as follows:

1. Set the filter on the Internet side to block any inbound packets whose source IP address matches the network number used for our LAN and the gateway itself. This insures any packets received by the gateway with the source IP address from our LAN must have come from our LAN (no IP spoofing from the public Internet is allowed).
2. Set the filter on the Internet side to block any outgoing packets that don't originate from the gateway itself (source IP address must match the IP address of the gateway).
3. Similarly, set the filter on the LAN side to block any outbound packets that don't originate from the gateway.
4. Set the filter on the LAN side to block any inbound packets that are not from our LAN (source IP address does not match the network number of our LAN). This prevents machines on our LAN from trying to use a different network number (IP spoofing), and also prevents packets from our LAN being forwarded to the Internet since the filter on the Internet side will only pass packets that originate from the gateway, not from our LAN.

With this setup, no traffic can be routed directly between the Internet and our LAN. It must pass through a server on the gateway and then be resent from the gateway itself (only the gateway is allowed to originate traffic in both directions).

By putting a mail server on the gateway, hosts on our LAN and hosts on the public Internet can both send and receive email from this server, but can never talk to each other directly. We can now post company information on our "Intranet" using web servers on the LAN. The public Internet will never be able to see these servers since they are behind our firewall.

Here's what this setup might look like using IPNetRouter. Suppose we have been assigned a static IP address of 192.115.101.231 by our ISP, and will use IP addresses from the range 192.168.0.x for our LAN. The filters described above can be specified as follows.




(NOTE: the list window above has been expanded to show all 8 filters at once).

The 192.115.101.231 address is used on E0 (Ethernet Built-in) attached to the public Internet.
We use 192.168.0.1 on E1 (Ethernet Slot 1) attached to our LAN.

Filters are invoked from TOP to BOTTOM. Hence the strategy is to first add a filter which BLOCKS all traffic on a specific interface, then add a PASS filter which will permit specific connections (datagrams). This is typically refered to as "punching a hole" into a firewall.

How does this work? Let's examine the filter order as seen by IPNetRouter:

| Filter Order (top to bottom) | Description | Result |
|---|---|---|
| Top | The first filter PASSes all packets RECEIVED on our slot 1 ethernet port which originated from our private subnet (only machines in our 192.168.0.1/24 subnet). | Net Result of Filters 1 & 2: Only client machines connected through the slot 1 port within our designated 192.168.0.1/24 subnet can use the IPNetRouter gateway. |
| 2nd | 2nd Filter: All other RECEIVED packets on this slot 1 interface are BLOCKed. | |
| 3rd | 3rd Filter: This filter PASSes all packets SENT from the gateway onto the private LAN. | Net Result of Filters 3 & 4: Only packets originating from the gateway machine can be SENT out to our client machines. |
| 4th | 4th Filter: All other packets SENT out of the gateway onto the private LAN to any other machine are BLOCKed. | |
| 5th | 5th Filter: This filter permits the gateway machine to SEND packets out to the Internet over the built-in ethernet port. | Net Result of Filters 5 & 6: Only packets originating from the gateway machine can be SENT out to the Internet. |
| 6th | Filter: All other packets SENT out of the gateway to the Internet are BLOCKed. | |
| 7th | This filter BLOCKs all packets from the Internet which claim to originate from the same IP address as our gateway. | Net Result of Filter 7: Our gateway has a unique, public IP address. No other packets received should have this same IP address...if so, BLOCK them. |

| | | |
|---|---|---|
| Bottom | This filter BLOCKs any packets received from the public Internet that claim to originate from our private LAN (192.168.0.1/24) | Net Result of Filter 8: Our private subnet is strictly reserved to be used by client machines connected through the Slot 1 interface. All external attempts to use the gateway from a machine on the Internet and having an IP address within our private subnet will be BLOCKed. |

Filtering is performed before Network Address Translation (NAT) for transmitted datagrams, and after NAT for received datagrams. This allows you to filter on the "Actual LAN Address" as opposed to the "Apparent Gateway Address".

## A More Practical Firewall for users with a Static IP Address

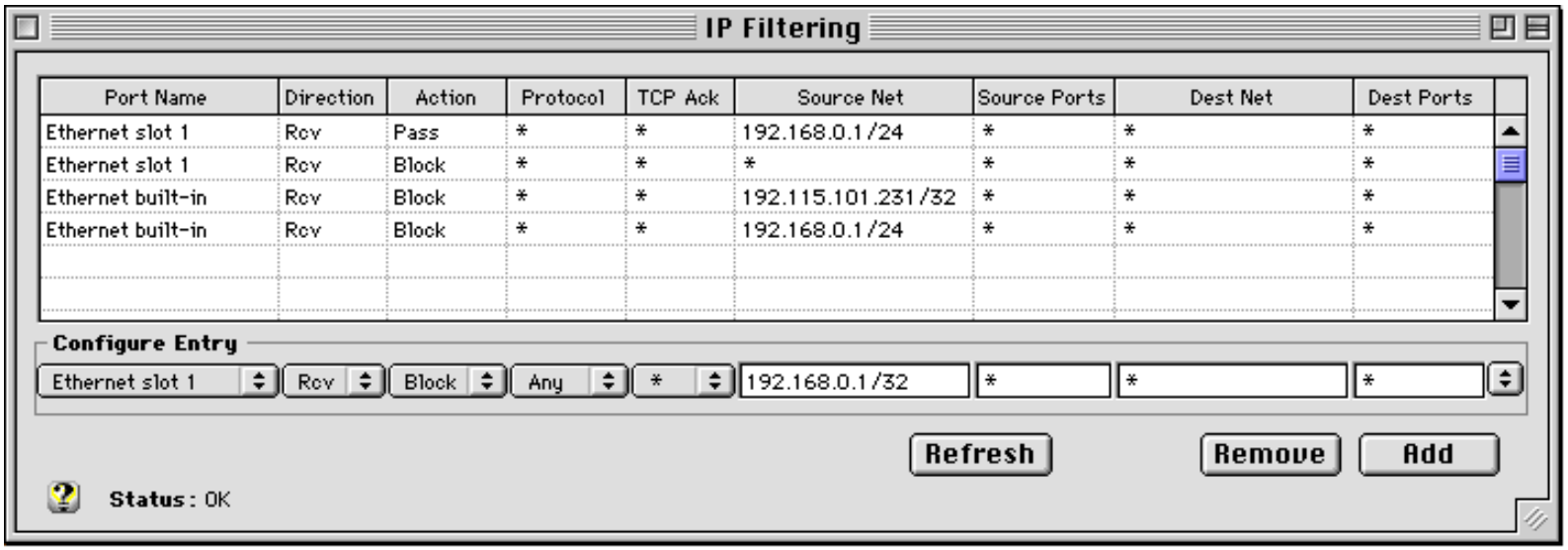
If you carefully followed the example above, you will note that this is an absolute firewall. Not only will it prevent outside users from getting in to your LAN, **but it will also prevent any client machines from getting out to the Internet!** This is the way most good network administrators start building a firewall:

1. First prevent all access.

2. Then permit any specific access on a case by case basis (commonly called "punching holes in a firewall").

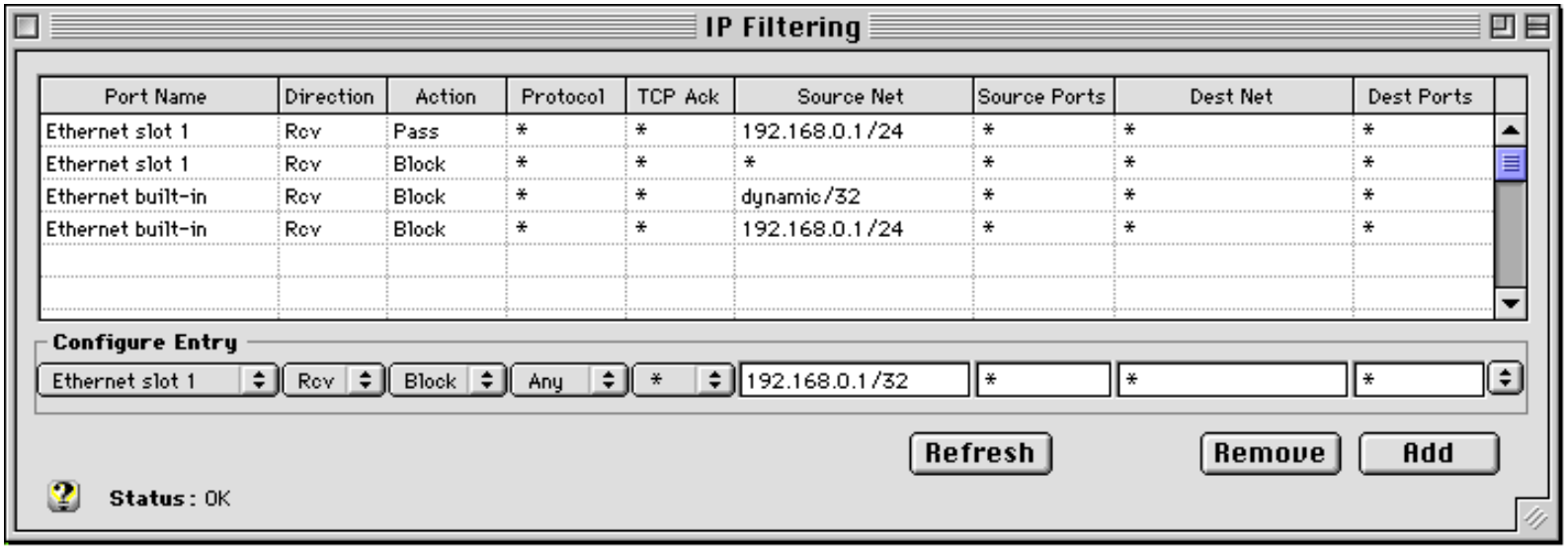
Most users of IPNetRouter, however, want all client machines to have access to the Internet, while preventing any intrusions into their client machines or unauthorized use of their router (i.e. spoofing). If you followed the example above, and understand the net results of invoking all of the eight filters, you will see that the way to let our client machines to access the Internet is to remove filter lines 3, 4, 5 and 6 of the above example.

This is what the IP Filtering window would look like (once again assuming the fixed IP address you have been assigned is 192.115.101.231):




## A More Practical Firewall for users with a Dynamic IP Address

Cable modem users are often assigned a dynamic IP address (through DHCP from their ISP), but still want the protection of a firewall while letting all their client machines access to the Internet. In this case, we simply replace the filter with the fixed IP address (second line from the bottom) with an entry for a dynamic IP address. To do this, just enter 0.0.0.1/32 in the Source Net edit box. Your IP Filtering window should appear similar to this:




## And if you really want to be a stickler....

There is one additional filter which can be added to the top of each of the above examples which will even further secure your private LAN. Most of the time this is not necessary, since most users of IPNetRouter are also the administrator of their own small private LAN. This filter simply ensures that no one else on the private LAN is claiming to be the gateway machine (which is not an easy thing to do anyway...since you would have to spoof packets). This filter would look like this:

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Ethernet slot 1 | Rcv | Block | * | * | 192.168.0.1/32 | * | * | * |

With this filter, all packets RECEIVED on the slot 1 interface and claiming to be from 192.168.0.1 (our IPNetRouter address) would be BLOCKed.

But, as mentioned previously, if you are in control of all the machines on your private LAN, there is very little use or need for this additional filter.

## Building a Firewall and using the DHCP Server within IPNetRouter

If you are using the DHCP server within IPNetRouter to dynamically assign IP addresses on your private LAN, then you have to add yet one more PASS filter to any of the above examples. The reason is that we are BLOCKING all incoming packets into the Slot 1 ethernet interface which are not originating from the local 192.168.0.1/24 subnet...and our client machines do not yet have an IP address in this subnet range (remember, these client machines are using DHCP to get an IP address assigned to them).

To remedy this, we add the following filter which permits incoming DHCP discover and request messages:

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Ethernet slot 1 | Rcv | Pass | UDP | * | * | 68 | * | * |

This filter PASSes all UDP protocol port 68 messages (DHCP negotiation messages) which are received on our internal ethernet interface so that the DHCP server within IPNetRouter can handle them.

## Order of NAT, Filtering, and Routing

```
                Proxy instance              Proxy instance
Internet<--->NAT-Filtering<--->IP<--->Filtering-NAT<--->LAN
```

Consider a packet travelling from the Internet to the LAN (left to right). Following the diagram above the order of processing is:

1. Packet originates from Internet
2. NAT on Internet side of gateway (reverse IP masquerading)
3. Filtering on Internet side of gateway

4. Routed by IP
5. Filtering on LAN side of gateway
6. NAT on LAN side of gateway (no operation).
7. Packet delivered to LAN

For packets from the LAN to the Internet, the order is reversed. Following the diagram from right to left:

1. Packet originates from LAN
2. NAT on LAN side of gateway if Local NAT is enabled
3. Filtering on LAN side of gateway
4. Routed by IP
5. Filtering on Internet side of gateway
6. NAT (IP masquerading) on Internet side of gateway
7. Packet sent on to destination on Internet

The Single Ethernet case is similar except that Local NAT is not available. Specifically, the instance of the Proxy module enabled
for IP masquerading is now also processing packets between IP and the LAN. It does not translate datagrams to or from our "Private Network" so that IP masquerading will work correctly in this configuration.

# It's a Wrap

Firewalls don't have to be complicated or expensive. A firewall is simply a mechanism for enforcing an access policy. The first step in choosing a firewall is to consider what access policies you wish to enforce. Using IPNetRouter, you can turn an existing Mac into a rock solid firewall router...for a lot less $ than any other commercial hardware based firewall router.

Just remember: first BLOCK...then PASS (and you only have to enter PASS filters when you want to punch holes in your Firewall).

There's lots of information about "firewalls" available on the web. Visit your favorite web search engine for more information.

---


Return to Getting Started

# DHCP and Mac OS

Apple has extended DHCP somewhat to accomodate wireless networks where a client might roam in and out of range of a DHCP server. If a client is unable to contact a DHCP server, it will try to auto-assign itself an IP address from the 169.254.x.x range, and then try again to contact the DHCP server every few minutes. Since a client can assign itself a temporary IP address, it may contact a DHCP server to request a new IP address while claiming to be bound to a different address. This creates a few possible conflicts:

- The server my try to respond to the bound address (ciaddr) the client said it has. There is no guarantee the response will be routed to the correct network.
- In Mac OS 8.6, the server may try to respond via hardware unicast. Since the client is already bound to a different IP address, it will not receive the response. The DHCP Server in IPNetRouter works around this by broadcasting the response if the first unicast response was not received.
- In Mac OS 9, the client will always request a broadcast response to avoid these conflicts. The DHCP standard however is ambiguous on how to respond to a DHCP DISCOVER or REQUEST when 'ciaddr' is non-zero. Some DHCP servers may misinterpret the request.

Some other differences in the latest DHCP client:

- When requesting or renewing an IP address, the Mac OS 9 client requests an explicit long lease time to insure any previous lease binding will be extended. Some DHCP servers do not respond properly to this request. They may reject it for example, instead of offering or granting a shorter lease time.
- The time out values used for the DHCP client have been reduced to avoid long delays while trying to reach a DHCP server. If there is no response within a few seconds, the client will assume the server is currently unavailable.
- Since the existing DHCP client runs synchronously, other processing comes to a halt while the DHCP client waits for a response. Reducing the time out interval helps eliminate the long pauses, but makes the Mac client more likely to regard a

busy DHCP server as unavailable and therefore auto-assign itself an address (169.254.x.x) which is not functional on the attached network. This should be fixed Mac OS 9.0.4 (OT2.6.?).

Previous known issues:

- Prior to Mac OS 8.6, the Mac DHCP client would release an assigned address when it was shutdown, and then try to request this address again upon restart. Some DHCP servers do not keep track of released lease bindings and would not respond causing excessive startup delays.
- The Mac DHCP client would not request an explicit lease extension (address time) when attempting to renew a previous binding. Some DHCP servers that followed the RFC literally would use the remaining time on the previous lease binding so the lease was never actually renewed.

Apple has posted some Tech Info Library (TIL) pages with more information here:

<http://til.info.apple.com/techinfo.nsf/artnum/n25049>

<http://til.info.apple.com/techinfo.nsf/artnum/n58372>

DHCP is a subtle protocol that is still evolving along with various implementations. As the developer of a DHCP server (in IPNetRouter) and DHCP client simulator (DHCP Test in IPNetMonitor), I hope documenting these issues will help resolve some of the conflicts.

You can use the DHCP Lease tool in IPNetMonitor (from my website) to explicitly force your Mac to get a new DHCP lease (easier than futzing with the TCP/IP control panel). There's also a tool to test your DHCP server so you don't have to guess if there is a problem accessing the DHCP server at your ISP.

For more information on DHCP, refer to RFC-2131, RFC-2132, <http://www.dhcp.org/>, or a good book on TCP/IP.

- Peter Sichel

---


Return to Getting Started

# IPNetRouter Guide to AirPort

Apple's AirPort 1.1 software includes a "preview" of the Software Base Station feature. The Software Base Station (Access Point or AP) uses a version of IPNetRouter supplied as a Faceless Background Application (FBA). "AirPort AP" in the Extensions folder is the application, and "AirPort AP Support" is the proxy module corresponding to OTModl$Proxy [use Conflict Catcher to get information about these files]. Notice that installing the AirPort software will remove OTModl$Proxy and replace it with a possibly older version named "AirPort AP Support".

AirPort AP (Apple FBA version of IPNR) is configured using the AirPort Setup Utility. Turning on the Software Base Station writes out an IPNR configuration file named "AirPort AP Configuration" in your Preferences Folder, and then sends an Apple Event to launch AirPort AP. Any changes you make to the Software Base Station writes out a new "AirPort AP Configuration" file. To use IPNetRouter to extend your AirPort configuration, you have two choices:

1. Quit the AirPort AP FBA and launch the UI version of IPNetRouter to modify your IP network configuration directly.

2. Modify AirPort AP to use a different configuration file that you specify (STR# resource 131 Option Settings, "ConfigFileName=Airport AP Configuration"). [Note this technique is for registered IPNR users who wish to integrate their use of IPNR with AirPort. Proceed at your own risk since the AirPort Setup Utility could produce unexpected results.]

To verify whether your AirPort AP configuration succeeded, you can examine the "AirPort AP.log" file in your Preferences Folder.

## Hardware versus Software Base Station (Access Point)

The AirPort Hardware Base Station is running a version of KarlBridge, which was developed by Doug Karl at Ohio State <http://www.karlnet.com/news/199912/199912-Airport.html>.

The AirPort Software Base Station uses a version of IPNetRouter developed by Peter Sichel <http://www.sustworks.com>.

What are some of the differences? That is, why would you choose one over the other?

1. The hardware AP is $299 whereas the software AP is included with Apple's $99 AirPort PC card but requires an AirPort capable Mac. The full UI version of IPNR is $89 if purchased separately.

2. With the hardware AP, everyone else doesn't lose their connection if you have to reboot your Mac.

3. The software AP does not support either AppleTalk or transparent bridging mode. It does IP routing with NAT and DHCP.

4. The software AP (IPNR) allows you to connect two Ethernet LANs via wireless. Hardware base stations cannot connect to each other in this way because they only operate in Base Station mode (there is no way to set Computer-to-Computer mode).

5. The software base station supports Trace Route (NAT with ICMP translation), can work with PPTP, and has various other implementation differences. It has the potential to offer higher performance and more flexibility (multiple Ethernets, PPPoE, LocalTalk, or any other Data Link Provider).

Apple will no doubt continue to enhance both products.

# Base Station versus Computer-to-Computer mode

Apple's Hardware AP operates in Base Station mode only, whereas the AirPort software in combination with IPNetRouter can set the radio to either Base Station, or Computer-to-Computer mode (802.11 "ad-hoc" mode).

Base Station mode is similar to a cellular network and provides relaying and power saving features. The Base Station can buffer traffic for low power clients allowing them to periodically turn on their radio and ask the base station for any traffic that may be waiting. By switching on the radio only long enough to receive any waiting traffic, clients can save precious battery power by not operating the radio all the time.

In Computer-to-Computer (ad-hoc) mode, the radio listens for incoming traffic all the time. Computer-to-Computer mode is intended for setting up ad-hoc limited range LANs

when no base station (or cellular network of base stations) is available. As a side effect, it may offer better network performance, but consumes more power.

By setting the AirPort Software to use Computer-to-Computer mode along with IPNetRouter, it is possible to connect multiple Ethernet LANs via AirPort wireless. AirPort wireless is just another data link provider logically equivalent to Ethernet.

Since Apple has not published the commands to switch the radio between Base Station and Computer-to-Computer mode, the only way to select Base Station mode is to use the AirPort Setup Utility (and then optionally modify the configuration or switch over to running the UI version of IPNetRouter).

---

SUSTAINABLE Softworks

products


# IPNetRouter Troubleshooting Tips





---

## First Check!

Quit IPNetRouter before doing any changes to the TCP/IP control panel.

**1. You MUST make sure that TCP/IP is NOT set to "Load Only When Needed".**

Open the TCP/IP control panel on the machine running IPNetRouter. Beneath the "Edit" menu select "User Mode...". Choose "Advanced". Click "OK". You will then see an "Options" button in the TCP/IP control panel. Click it. Make sure that TCP/IP is "Active" and the "Load only when needed" checkbox is **UNCHECKED**.

If you are connecting with PPP (Remote Access), you must uncheck "Load Only When Needed" for your PPP configuration in the TCP/IP control panel as well (even if it is not Active).

After making any of the above changes, restart your machine. Relaunch IPNetRouter. You just might be surprised to find out that everything works!!

**2. IF you are running IPNetRouter with a modem or ISDN terminal adapter (connecting with PPP):**

Open the TCP/IP control panel. Your settings **MUST** display the **TCP/IP configuration FOR THE LAN**.

Typically these settings will list this machine's address at 192.168.0.1, have the router address BLANK, and have the correct DNS servers listed for your particular ISP.

If you open your TCP/IP control panel and you see that your PPP TCP/IP settings are displayed, you must change these so that the TCP/IP configuration for your LAN is displayed (made active). Do this after quitting IPNetRouter.

---

## General Information

You may need to launch IPNetRouter from the application once while pressing and holding the Option key down. This tells IPNetRouter to rebuild your link stream with the supplied Proxy module included. Normally this occurs automatically, but may be necessary if you have replaced your TCP/IP Preferences File or determine IP masquerading is not working.

For Mac OS 8.1 or earlier, you may need to restart once after trying to start the Monitor or Router to give Open Transport a chance to build your Link Stack with the supplied STREAMS module included.

Do not change the settings in your TCP/IP control panel while IPNetRouter is running. Quit IPNetRouter first, make the necessary changes in the TCP/IP control panel, and then re-launch IPNetRouter.

You need to uncheck "Load only when needed" in the TCP/IP Control Panel so TCP/IP will remain loaded at all times. Set the User Mode to advanced (under the Edit menu) and click on the Options button to access this setting.

To avoid possible conflicts, configure IPNetRouter before running other TCP/IP software on your gateway machine. This means you'll need to download and print out the instructions so you don't need to have your web browser running while you try to configure IPNetRouter.

If you have previously installed or used a Vicomsoft product, please see the FAQ page for more info.

---

## Understanding the IPNetRouter User Interface

IPNetRouter is mostly a utility for sending commands to the IP Module in Open Transport.

You specify the parameters of the command in the Configure Interface box, and then send the command by pressing the Add or Remove button. The Interfaces table shows the actual IP interfaces the IP module knows about. When you click on a line in the table, it copies the current information from that line to the Configure Interface box below to save you the trouble of retyping it, thus making it easier to modify an existing interface.

It is the parameters in the Configure Interface box that control what happens, not what line of the table is selected. The IP module identifies which interface you are trying to modify by the interface name field in the Configure Interface box.

There is help available for each window by clicking on the question mark symbol in the lower left corner.

See the IPNetRouter download page for more info on our new web configurator application. If you are having problems understanding how the IPNetRouter interface should be changed to work on your Mac, this often proves to be an easier way to get everything up and running.

---

# Identifying Problems and Getting Help

TCP/IP networking has many components which can be intimidating when something doesn't work. The key to troubleshooting network problems is to recognize the sequence of components and look at each one individually to see what isn't working.

You can use the tools in IPNetMonitor to verify whether each leg of your network is functioning correctly. From the gateway machine, you should be able to ping hosts on both your LAN, and the public Internet. From a client machine, you should be able to ping the gateway, and hosts on the public Internet. You can use the Monitor tool in IPNetMonitor to see your ping datagrams leaving one machine and arriving at another. Red bars indicate transmit data while green bars show receive. On the gateway, you can see data being received on one interface (green) and being forwarded out another (red). [IPNetMonitor is distributed as 21-day trialware.]

Check that you have specified the correct Name Servers on each machine and that you can ping to them. If your ISP didn't give you any Name Server addresses (because they want you to use DHCP), you can use IPNetMonitor to find your default Name Server. Once your gateway is connected, launch IPNetMonitor and press Cmd-L to open the NS Lookup window. Then press Cmd-T to begin a ping test to your default Name Server (used by the NS Lookup window). The IP address of your default Name Server will appear at the top of the Ping window.

IPNetRouter (IPNR) is mostly a utility for sending commands to the IP module in Open Transport. When you save an IPNR configuration, IPNR writes a plain text file that lists the commands needed to restore this configuration. Each line generally corresponds to a row in the Interfaces, Routes, Port Mapping, or Filter windows.

Here's an example of an IPNR configuration file for a PPP dialup connection:

```
#forwardingAutomatic
+interface\Ethernet\mace0\192.168.0.2\255.255.255.0\
+interface\PPP\IPCP0\0.0.0.0\0.0.0.0\masquerading\
+route\Direct\192.168.0.0\192.168.0.2
+route\Direct\224.0.0.0\192.168.0.2
+route\Default-Gateway\0.0.0.0\0.0.0.0
#end
```

Notice you don't need to copy down the window contents or send a screen snapshot to describe your configuration, select File->Save to create a configuration file, and then drag-and-drop your configuration file into your email editor. If you are still having trouble and would like me to help, please include your IPNR configuration as above and the output of the IPNR Log Window. This will allow me to see your entire IPNetRouter setup.

For me to find "a problem", I need to know specifically what didn't work. The IPNR configuration file showing the commands to be executed along with the IPNR Log Window output that shows the result of each command usually provides the information I need.

If a saved IPNR configuration isn't working, try recreating your desired configuration (per the instructions on our web site) and Saving it only after you have tested to see that it works. Saving a configuration that doesn't work will only perpetuate the problem.

If you have questions about configuring other software such as Windows-95, Windows-NT, or TIMBUKTU, check the IPNetRouter FAQ and then try posting your question to our NetTalk mailing list. Other users are likely to have experience with the same issues. TCP/IP networking is too vast a subject for one person to know everything.

You can edit IPNetRouter saved configuration files using any text editor such as SimpleText. BBEdit Lite is particularly convenient because it allows you to edit a file without changing its file type and creator (so it remains an IPNetRouter document). If your editor does change the file type and creator, you can drop the resulting file on the IPNetRouter

application to invoke it explicitly and then do a Save As to store this configuration as an IPNetRouter document.

---

## Using PPP with IPNetRouter

You must not have PPP selected as the "Active" interface in the TCP/IP Control Panel.

If IPNetRouter is interrupted during a PPP connection attempt, IPNetRouter may be unable to restore the currently selected configuration in the TCP/IP control panel. In this case, you may need to manually select the desired configuration in the TCP/IP control panel to restore normal operation.

## Configuring TCP/IP to use PPP

In the TCP/IP Control Panel, select "Configurations" under the File menu. You must have at least two configurations:

1. A PPP configuration named "IPNetRouter". This configuration must not be active, but you still need to have it so OT/PPP can find its configuration in the TCP/IP Preferences File. This configuration must be set to allow PPP to connect automatically when launching TCP/IP applications (otherwise the PPP configurator will fail to build a PPP link stack). Notice in Mac OS 8.5 the "PPP" control panel has been renamed "Remote Access."
2. An Ethernet (or MacIP) configuration with some other name (not "IPNetRouter"). This configuration should be made Active.

In both configurations above, you must unselect "Load only when needed" in the TCP/IP Control Panel. You want TCP/IP to remain loaded at all times.

---

## Cable/DSL Modem and Single Ethernet

When using a "Single Ethernet" configuration, your LAN is directly attached to the cable modem network through your hub. It is possible someone else on the cable modem network will try to use the same IP addresses suggested in the getting started example on our web site. If this happens, you could see the following message:

> Another device on your TCP/IP internet, which has the physical address 00 10 XX XX XX XX, is currently using the same IP address (dd.dd.dd.dd). Your TCP/IP network interface has shut down.

Notice the IP address reported (dd.dd.dd.dd) may be the address specified in your TCP/IP control panel even though it is the LAN IP address (192.168.0.1) that is in conflict. The warning dialog doesn't realize your machine can have more than one IP address.

To work around this problem, you need to choose a different sequence of IP addresses for your LAN. If your LAN is currently using 192.168.0.1, 192.168.0.2...192.168.0.n for example, you could change the 3rd octet to create a different set of addresses: 192.168.117.1, 192.168.117.2...192.168.117.n.

Alternatively, you can change to a Dual Ethernet configuration to isolate your LAN from the cable modem network. This makes your private IP addresses truly private (no one else will see them).

---

## Using Two Ethernets on NuBus Macs

If you have a NuBus Ethernet card in addition to "Ethernet built-in", you should use the TCP/IP control panel to configure Ethernet built-in, and IPNetRouter to configure any remaining NuBus cards. If you try to use the TCP/IP control panel to select your NuBus Card, multihoming may not work due to a limitation of the Ethernet driver on these machines.

If your machine doesn't have built-in Ethernet, put the cable modem Ethernet interface in the lower NuBus slot (towards the outside of the case first).

## Other Common Ethernet Problems

One major source of problems is bad or no wiring. Check all cables and ports to make sure everything is connected properly. Some ethernet cards will not be recognized by the OS unless they are actually connected to another powered ethernet device via an ethernet cable. Check to make sure you do not have a crossover cable in a regular port when that would not be appropriate. Verify that you can get link lights and that ethernet over Appletalk works. Next, verify that TCP/IP can be used without IPNetRouter. If both are true but you cannot get IPNetRouter to route IP please review the rest of this section.

Some ethernet cards and drivers are not compatible with one another. In some cases, two cards from the same manufacturer (eg Asante, MacSense) may conflict with one another. A simple solution in this case is to install drivers and cards from two different manufacturers. IPNetRouter 1.5 fixes an incompatibility with some cards (eg Farallon) that have spaces in their driver names. Apple broke most ethernet

drivers in OS 9. Some nubus cards work better without their drivers or with older drivers, depending on the Macintosh OS system you might be running them on. Many older scsi-ethernet solutions only work on certain 68k machines. A few cards do not work well with particular machines or in certain slots. Search our nettalk archives for information on a particular manufacturers cards and solutions found by users. Make sure that you have the latest available drivers and that they are compatible with your gateway machine's OS. Verify with the card manufacturer that your machine should work with IPNetRouter in the configuration you are trying to setup.

Some hubs and switches are not compatible with some cards and/or Macintosh built-in ethernet implementations. See Apple's Tech Support Area at their website for various tech notes and software that can be used to resolve these problems.

---

## PPPoE (Enternet, MacPoet, Sympatico)

IPNetRouter can be configured with PPPoE clients. However, you must do it in a very specific order and there are some other limitations, depending on which implementation you have, etc. See the PPPoE setup page for details.

---

## Network Time Server Issues

Apple's Network Time Server, included in the Date & Time control panel starting in OS 8.5, may not work correctly unless you use large time intervals (10+ hours) between synchs.

Also, this feature may cause your gateway to continuously try to reconnect if it is set anywhere on the LAN to synch frequently.There are several solutions for related issues if you require more precise time or are experience trouble with a particular network time server.

We encourage you to search our Nettalk archives on the terms "time troubles" (don't include the quotes) as an exact phrase to see a host of interesting information about times servers, alternatives to Date & Time control panel, etc.

---

## Problems With MacPPP

MacPPP was installed in the very early releases of Open Transport. Various versions of MacPPP were also available as shareware. None of these versions is supported by IPNetRouter. Apple replaced MacPPP with OT/PPP. IPNetRouter is designed to work with OT/PPP. See FAQ page for info about FreePPP and ARA.

---

## "Thread Manager Required"

Message appears on 68k OT 1.1.1 or earlier system, even after Thread Manager is present. Install OT 1.1.2 and the installer will work correctly! This is probably a bug in ASLM.

---

## MacOS 9.0 Compatibility

MacOS 9.0.x introduces Open Transport 2.5 & 2.6 and other significant changes that effect IPNetRouter. The main changes are:

1. Apple changed ethernet significantly in MacOS 9, adversely effecting 3rd party ethernet drivers, especially in dual ethernet configurations.
2. Airport Software is included as part of the MacOS.
3. Apple Remote Access Server is now included with MacOS 9. This includes the ARA Client that has been present since MacOS 8.5.
4. MacIP is no longer supported over LocalTalk.
5. The Multiple Users control panel was added.
6. Software Updates is not compatible with IPNetRouter on the routing Mac.
7. New File Sharing implementation.
8. Various bug fixes and additions, especially for DNS, DHCP, and PPP dialup handling.

How does this affect IPNetRouter?

- We recommended that you use IPNetRouter 1.5 or later with OS 9. IPNR 1.4.7 or later is supported. As always, be sure to read the accompanying readme and release notes documents carefully before running IPNR.
- If you have ANY 3rd party ethernet cards installed, verify with the manufacture that you have the latest MacOS 9 compatible drivers installed if you encounter any problems with IPNetRouter. Apple

broke the ethernet drivers of most 3rd party vendors in OS 9; you MUST have Mac OS 9 compatible drivers for the card(s) to function properly. These drivers are only available from the card manufactures. If there is no MacOS 9 updates available for your particular card, you may not be able to get it to function properly with IPNR on OS 9. The best solution if there are no compatible drivers available is to revert to an earlier version of the Mac OS that does support a non-Mac OS 9 compatible driver or to buy a Mac OS 9 compatible card. NOTE: This is a very good reason not to upgrade nubus Power Macs to OS 9 if you are dependent on ethernet cards for your routing unless you are certain the card is supported in OS9. ALSO NOTE: Madge PCI Token Ring cards have been shown to work on OS 9 with IPNR and built-in Ethernet. See the other ethernet troubleshooting sections for more info on general ethernet issues.

- The following Mac OS 9 System software updates are required for IPNR to function correctly with versions of MacOS 9 prior to the release of 9.0.4:
  1. You SHOULD update your version of Open Transport to OT 2.6.1. This update is available from Apple's website.
  2. You MUST update any installed Airport software to version 1.1 or later. 1.0.x is NOT compatible with the IPNR software. This problem occurs even if you do not have any Airport hardware installed--you MUST update installed Airport software for IPNR to function correctly in ALL cases. This update is available from Apple's website.
- When feasible, upgrade to Mac OS 9.0.4 if you already have an earlier version of OS 9 installed.
- Treat Airport cards as if they are Ethernet ports for purposes of configuration with IPNetRouter. You can remove the Airport AP application and associated files from your system folder if you wish to use IPNetRouter for routing TCP/IP over an installed/configured Airport card on the gateway Mac. The Airport AP application included in MacOS 9 was version 1.4.8.0 of the IPNR Faceless Background App (FBA). Note, this application and associated files must be present in the system to use Apple Airport utilities for configuring your card as a basestation, etc. Retain a copy of these files if you do decide to removing them from the system. See the Airport section of our website for more info on using Airport with IPNetRouter.
- Mac OS 9 comes with Apple Remote Access (ARA) Server installed by default. See the IPNetRouter FAQ page for more info regarding ARA and IPNR.
- Don't upgrade to OS9 if your router depends on MacIP over

Localtalk to support your LAN. Sadly, Apple neglected to support MacIP over Localtalk in OS 9. See Apple's Technote for the poop.

- OS 9 comes with a new Multiple Users control panel. If your machine is configured for more than one user you may have a problem running IPNetRouter. This problem occurs because each user has a seperate system preference folder. There are two workarounds to getting IPNetRouter to work on machines with multiple users.
  If you are using Multiple Users on your IPNetRouter gateway, install the latest version of our Faceless Background App of IPNetRouter. It runs independently of the Multiple User API interfaces. You can download the FBA from the IPNetRouter download page elsewhere on this site. Note: You still have to configure the Faceless Background App using a regular copy of IPNetRouter first.
  The alternative is to install IPNetRouter for each user. This is not recommended due to complications of licensing, complexity of installation, etc.
- See the FAQ page for information regarding Software Updates and related installer compatibility issues with IPNetRouter.
- File Sharing in OS 9 is a an implementation of Open Door Network's ShareWay IP software. It now supports file transfer via TCP/IP. If you wish to Filesharing over IP on both the local LAN and the internet, map the AFP port to the appropriate interface on your gateway using IPNetRouter's port mapping feature. Which interface this is may differ depending on your gateways configuration.

---

## MacOS 8.5 Compatibility

MacOS 8.5 introduces Open Transport 2.0 with a number of changes. The main changes are:

1. OT/PPP is replaced with Remote Access (the full ARA 3.0 client that supports AppleTalk over PPP and ARAP).
2. A new much faster Ethernet driver for Ethernet built-in.
3. Improved Finder copy code.
4. Various bug fixes and additions for developers.

How does this affect IPNetRouter?

- Previous versions of IPNetRouter may not complete PPP connections successfully with Remote Access. You should

download IPNetRouter 1.3 or later to work around this problem.

- Some 3rd party Ethernet drivers are not compatible with Mac OS 8.5. The Farallon driver for the PN593-C/PN893 (Etherwave/EtherMac) will fail under System 8.5 due to a problem with Apple's ENet driver. The solution is to remove the Apple Enet extension.

---

## Installing Over Another Version of IPNetRouter

The standard installer is designed to install a newer version over a previous one. If you have any trouble, the following steps will insure the application is installed correctly.

1. Find any previous version of OTModl$Proxy in your Extensions folder and move it to the trash.
2. Remove any previous version of the IPNetRouter Application.
3. Run the correct IPNetRouter installer for your machine (PPC or 68K) to install the desired version of IPNetRouter.
4. Restart your computer (this step is very important). The installer only "suggests" you restart because it isn't always necessary, but you should do this if you are not sure.

   I recommend against using the "uninstaller" if you are just going to re-install the application. The uninstaller is for removing IPNetRouter completely and makes it harder to re-install. If you do use the uninstaller, here's how to complete the re-install process.
5. Hold down the Option key when you launch the IPNetRouter application the first time after re-installing. Continue pressing the option key until the application comes up. This forces Open Transport to rebuild your link stream.
   - or -
   Alternatively, configure IP masquerading, and then Restart your Mac yet again.

The purpose of step (5) is to give Open Transport a chance to rebuild your link stream with the Proxy module inserted. You can verify this was successful if you are able to use the monitor window in IPNetMonitor to monitor data on the interface you will be using to masquerade.

Back to the top

---

SUSTAINABLE Softworks

products

# IPNetRouter Overview

IPNetRouter is the simple low cost way to share your cable modem, xDSL, or dial-up internet connection with your entire LAN. With IPNetRouter, you can use multiple IP interfaces at the same time (such as Ethernet and OT/PPP for example), and specify additional routes for communicating with more than one IP gateway.

IPNetRouter is a complete IP router and firewall solution including a built-in DHCP Server, NAT with inbound port mapping, and IP filtering to setup your own firewall. Other features include support for PPPoE and Quicktime streaming. IPNetRouter's OT native design offers superior performance easily outperforming other low cost hardware and software solutions.

System Requirements

- MacOS 7.5.3 and Open Transport 1.1.1 or later (OT 1.1.2 for 68k)
- Macintosh PowerPC or 68030 or higher.
- A Macintosh capable of connecting to the Internet using the TCP/IP protocol suite.

---

## Use our IPNetRouter Configuration Application NEW

(strongly recommended for first time IPNetRouter users)

Our IPNetRouter Configuration Application is designed to simplify IPNetRouter setup for basic Internet configurations. This includes PPP, cable modem, and DSL/ADSL Internet connections (with or without

PPPoE).

**Important Note:** Use of the IPNetRouter Configuration Application is optional, but STRONGLY recommended. It provides users with guided step-by-step instructions for configuring IPNetRouter. Please read the IPNetRouter Configuration Application instructions before use.

SUSTAINABLE Softworks

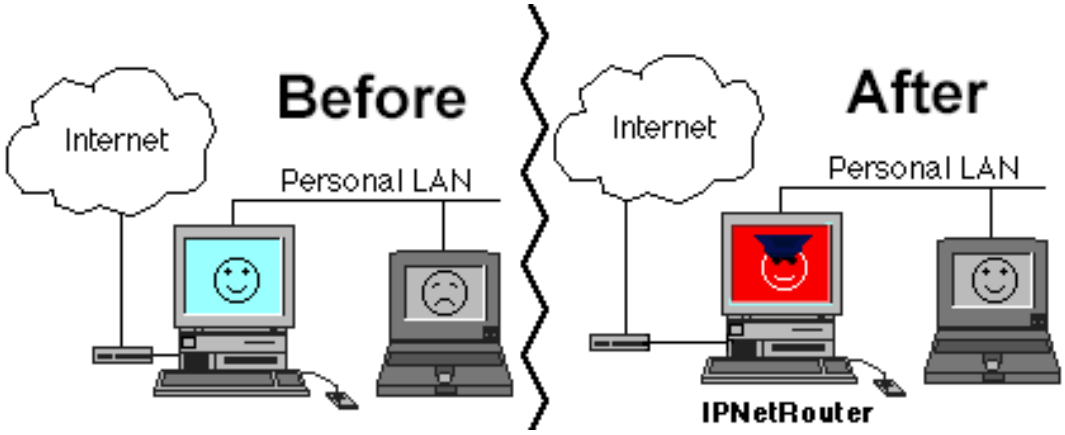
products



# IPNetRouter Description and Specification

## Introduction

IPNetRouter is Macintosh router software which permits you to share your Internet connection. This includes PPP, Cable Modem, DSL, ADSL, and PPPoE Internet connections. Designed from the ground up to take full advantage of Open Transport, IPNetRouter offers superior performance with greatly reduced cost and complexity.




As shown above, before installing IPNetRouter you have one machine directly connected to the Internet and one or more machines networked via a personal LAN (Local Area Network). Unfortunately, the other machines cannot access the Internet with this setup.

After installing IPNetRouter, the PowerBook and any other computers on the LAN have full Internet access. At the same time IPNetRouter becomes an Internet sentry (firewall) and prevents unauthorized access to machines on your LAN. This is because IPNetRouter enables Open Transport to forward packets between the Internet and the Personal LAN through a mechanism called "Network Address Translation" (aka "NAT"). With both NAT and the packet filtering capabilities of IPNetRouter, you can build a very solid firewall which prevents unauthorized access from the Internet to any of your machines on your personal LAN.

Since IPNetRouter understands network addresses that are independent of the LAN hardware, it can connect networks using different physical media. That is, you could have a private LocalTalk network, a private ethernet network, a private wireless network and IPNetRouter would be able to make intelligent decisions about which packets to forward to/from the Internet and how to transfer packets from one network to another. This capability is called "multilink multihoming" and permits your Macintosh to have several different IP addresses on several different physical networks...a feature most often associated with much more complicated UNIX based computers.

## System Requirements

- MacOS 7.5.3 and Open Transport 1.1.1 or later (OT 1.1.2 for 68k)
- Macintosh PowerPC or 68030 or higher.
- A Macintosh capable of connecting to the Internet using the TCP/IP protocol suite.

# Technical Description and Specification

IPNetRouter consists of the following major components:

- A utility for sending commands to the IP module in Open Transport
- A STREAMS module that performs Network Address Translation and IP Filtering
- Support for PPP dial-on-demand
- Support for saved configuration files
- A graphical user interface
- A DHCP Server

The following sections examine each of these components in detail.

## Utility for Sending Commands to IP

This component sends IOCTL commands to configure and bring up additional IP interfaces based on the following parameters:

- The *user port name* or physical *port name* (from the OT Port Registry)
- The *interface name*
- The *IP address*
- The *network mask*
- A set of boolean values that specify whether to bring the interface up or down; enable IP masquerading on this interface; designate the interface as unnumbered; and prevent a saved IP address from overriding a DHCP assigned address.

This component builds additional Link Stacks (the chain of modules between IP and a Data Link Provider (such as Ethernet) to support multilink multi

homing. Any Link Stacks created by IPNetRouter will be dismantled when the program is quit.

In general, IPNetRouter can work with most Open Transport compatible Data Link Providers such as Ethernet, Fast Ethernet, Token Ring, PPP and others. In the case of PPP however, dial on demand support is more limited (see below).

IPNetRouter supports MacIP (IP in AppleTalk) as a client. This means IPNetRouter can build a Link Stack that will add or remove the AppleTalk wrapper for MacIP allowing LocalTalk connected devices to access TCP/IP services via IPNetRouter. IPNetRouter does not assign IP addresses to other MacIP hosts at this time (as a traditional MacIP gateway), it is necessary to configure each MacIP host manually.

This component enables or disables how the host forwards IP datagrams. IP_FORWARD_NEVER (0) to inhibit forwarding (RFC 1122 compliance); IP_FORWARD_ALWAYS (1) to always forward; IP_FORWARD_AUTOMATIC (2) to forward only if the number of IP interfaces on the system is greater than 1 (SunOS 4.1.x compatible).

IPNetRouter supports static routing. It does not participate in RIP exchanges or other routing table update protocols at this time.

## STREAMS module that performs Network Address Translation (NAT) and IP Filtering

An Open Transport STREAMS module (shared library) named OTModl$Proxy is stored in the Extensions folder and inserted in any IP Link Stacks immediately below the IP module itself. We will refer to this module as the "Proxy" module through out the rest of this specification.

The Proxy module performs Network Address Translation (RFC-1631) with extensions for Port Multiplexing and ICMP translation. Port Multiplexing allows an entire LAN to hide behind a single public (globally unique) IP address by assigning unique port numbers to each actual endpoint. ICMP translation allows diagnostic tools like Trace Route to work through the translator by keying off the "triggering datagram" returned in ICMP response packets.

The use of a single public address with Port Multiplexing means that protocols that do not include a protocol port number (ICMP and GRE for example) are limited to a single host at a time.

The NAT function translates IP address and port information in the IP header, TCP header, and UDP header. Certain protocols also embed IP address and port information within the datagram body itself. Among the most heavily used protocols, FTP does this when configured to use "PORT" mode as opposed to "passive" (PASV) mode. The Proxy module recognizes PORT commands on the standard FTP control connection (port 21) and

translates them as a special case to provide ftp masquerading. Specialized gaming or streaming protocols may not work with NAT without additional support.

GRE packets are passed unmodified to support PPTP for Window-NT VPN.

The NAT function is normally enabled based on the physical port (from the OT Port Registry). To facilitate using cable modems with a single Ethernet (two IP interfaces on the same physical port), IPNetRouter can designate one IP interface as a "Private Network" so it will not translate datagrams to or from this IP interface even though it is translating other datagrams on the same physical port.

The Proxy module currently supports masquerading on up to four Interfaces at a time (one translation table) using a single public address with a maximum of 500 simultaneous connections (the translation table can hold 500 entries).

The NAT table can be edited under user control to support Inbound Port Mapping. See the Port Mapping page at <http://www.sustworks.com/products/ipnr/gettingstarted/port_mapping.html>.

The Proxy module also supports IP Filtering on a per port basis. See the Building Your Own Low Cost Firewall page at <http://www.sustworks.com/products/ipnr/gettingstarted/firewall.html>. The module includes built-in filtering for Windows-95 and Windows-NT NetBIOS DNS queries to prevent spurious connection attempts (UDP datagrams from source ports 137, 138, and 139).

## Support for PPP dial-on-demand

Open Transport 2.0 and earlier as supplied by Apple assumes there can only be one IP data link active at a time. If that data link is temporary in nature such as PPP, Open Transport will initiate a connection when an application on that machine tries to open an IP endpoint. Since IPNetRouter allows multiple data links to be active and might receive IP traffic for a PPP interface from another host, the existing mechanism does not provide the desired capability.

IPNetRouter implements dial-on-demand as follows. The IPNetRouter application monitors the state of the PPP interface and notifies the Proxy module whether that interface is up or down. While the interface is down, IPNetRouter will periodically poll the Proxy module to determine if it has seen any IP traffic for an on demand interface. When an instance of the Proxy module that is pushed above a PPP data link provider detects IP traffic for an interface that is disconnected, it sets a flag to notify the IPNetRouter application the next time it polls. The IPNetRouter application then launches a connection attempt for that interface.

Since IPNetRouter needs to monitor the state of the PPP connection and

initiate a connection attempt when there is traffic for the corresponding interface, IPNetRouter requires some PPP specific code to carry out this task. It is for this reason the dial-on-demand feature only works with certain PPP implementations. IPNetRouter supports dial-on-demand with OT/PPP and FCR PPP (a.k.a. LinkUPPP or LeoMLP) at this time.

When IPNetRouter initiates a PPP connection attempt, it is possible the connection attempt will fail (line is busy, modems fail to connect, ISP is offline, etc...). In this event, IPNetRouter will retry the connection attempt five times. The first 3 tries tell PPP to initiate a connection attempt. The next three tries will remove the PPP interface and then add it back. If all 6 tries fail to connect, IPNetRouter will wait 10 minutes and then try again to launch a connection attempt repeating the process above.

It is possible for other PPP implementations to support dial-on-demand directly without oversight from IPNetRouter. In this case, PPP would initiate a connection when it receives IP traffic, and inform IPNetRouer of the negotiated local IP address possibly using the command file interface described below.

## Support for saved configuration files

IPNetRouter allows you to save the current networking configuration including any interfaces, routes, port mapping, filters, or other gateway settings you have specified. To save the current configuration you can choose Save under the IPNetRouter File menu. To restore this configuration at a later time, you can open the corresponding configuration file from the Finder by double-clicking or placing it in the Startup Items Folder. IPNetRouter configuration files contain human readable text to simplify debugging or controlling the application directly from stored configuration files.

The format of IPNetRouter configuration files is largely self evident. Consider the following example:

```
                                    #forwardingAutomatic
+interface\ \lo0\127.0.0.1\255.0.0.0\
+interface\Ethernet\mace0\192.168.0.2\255.255.255.0\
+interface\PPP\IPCP0\0.0.0.0\0.0.0.0\masquerading\
+route\Direct\192.168.0.0\192.168.0.2
```

```
+route\Direct\224.0.0.0\192.168.0.2
+route\Default-Gateway\0.0.0.0\0.0.0.0
+map\icmp\209.6.64.35:0\209.6.64.35:0\1\
#end
```

Keywords corresponding to individual settings begin with the pound character "#" in the first column. Commands to the IP module begin with a plus "+" or minus "-" character. Command parameters are delimited by the backslash "\" character. The configuration ends with a line containing "#end". Any line beginning with a character other than "#", "+", or "-" will be treated as a comment and ignored.

IPNetRouter process the commands in order trying to make the current networking configuration match the configuration contained in the file. Any existing interfaces or routes not listed in a saved configuration will be removed. If the keyword "#addOnly" is seen, IPNetRouter will not try to match the entire configuration, but simply execute any commands in the file. If the keyword "#delete" is seen, IPNetRouter will delete the configuration file after processing the commands therein.

"+interface" specifies to bring up an interface. "-interface" specifies to remove an interface. "+route" adds a route, "-route" removes a route. "+map" and "+filter" entries work similarly.

IPNetRouter recognizes the Open File Apple Event which can be used to programmatically change the networking configuration. The application creator signature for IPNetRouter is "IPnl". IPNetRouter can be configured as a faceless background application to hide the default user interface described below.

Keyword commands:

```
                                    #forwardingNever
#forwardingAlways
#forwardingAutomatic
```

#configureOnly
#configureDisplay
#auto
#addOnly

#remainConnected
#remainconnectedOff
#disconnectAtQuit
#disconnectAtQuitOff
#showPPPdialogs
#showPPPdialogsOff
#dialOnDemand
#dialOnDemandOff
#enableLocalNAT
#enableLocalNATOff
#TRCableModem
#TRCableModemOff
#PPPName=<name-of-TCP/IP-config>

#disconnect
#connect
#delete
#quit

Interface parameters

\masquerading
\unnumbered
\dhcp
\nogateway
\mtu=

## Graphical User Interface

In Macintosh fashion, the IPNetRouter user interface consists of a menu bar with standard commands and a set of windows that allow you to edit application data thereby controlling the application and networking configuration. Individual windows can be invoked from the Window menu. To allow a single IPNetRouter configuration document to represent the entire network configuration, the Routes Window, Gateway Window, Port Map Window, and IP Filtering Window are children of the Interfaces Window and correspond to different views of a single document. Closing the Interfaces Window closes the corresponding document and will automatically close its dependent windows.

Each IPNetRouter window contains a help button in the lower left corner that describes the window controls in more detail.

Many of the windows follow a similar model for sending commands to the IP module in Open Transport. A table along the top of the window shows the current configuration. You specify the parameters of the command in an entry area below the table, and then send the command by pressing the Add or Remove button. The Interfaces table for example shows the actual IP interfaces the IP module knows about. When you click on a line in the table, it copies the current information from that line to the Configure Interface box below to save you the trouble of retyping it, thus making it easier to modify an existing interface. It is the parameters in the entry area that control what happens, not what line of the table is selected. The IP module identifies which interface you are trying to modify by the interface name field in the Configure Interface box.

If you try to send commands that don't make any sense, the IP Module will not accept them and no changes will appear in the table.

A status line along the bottom of the window provides a summary of any unexpected conditions or errors encountered while trying to perform a command. The IPNetRouter Log window records what the program is doing and will often provide additional information that would not fit in the status area. The contents of the log are also written to the file "IPNR.log" in your Preferences Folder. The log file is automatically re-used each time you launch the IPNetRouter application. In the event of a system crash that precludes reviewing the log window, it is important to rename or copy the IPNR.log file before re-launching IPNetRouter.

## DHCP Server

The DHCP window allows you to configure the DHCP server built-in to IPNetRouter. In order to access the Internet, each computer on your LAN needs to be configured with four pieces of information (IP Address, Network

Mask, Router Address, and Name Server Address). The DHCP server in IPNetRouter allows other computers on your LAN to get this information automatically from IPNetRouter by selecting "Configure via DHCP" in the corresponding TCP/IP control panel.

Using DHCP (Dynamic Host Configuration Protocol) avoids the need to manually configure each computer on your LAN before it can be used, and is especially useful if you have many computers on your LAN or frequently move computers from one network to another.

Before you can use DHCP to configure other hosts on your LAN, the DHCP Server itself needs to be configured. In most cases, you can simply press the "Use Defaults" button and enable the server by selecting "DHCP Server On". These settings assume your LAN will use IP addresses from the range 192.168.0.x and that your gateway running IPNetRouter will use 192.168.0.1 as its local address. [*WARNING* If you are connecting using a cable modem and single Ethernet, you must not enable DHCP on this Ethernet since DHCP requests are broadcast to all interfaces on that physical network.]

If your Local Area Network has additional requirements, you can use the DHCP Window to control what IP addresses to assign, to whom, for how long, and other operational parameters. The information in the DHCP Window is divided into four tables or panels which are briefly described below. For more information on DHCP, refer to RFC-2131, RFC-2132, or a good book on TCP/IP.

**STATUS TABLE**

This table keeps track of Lease Bindings. What addresses are currently leased, to whom (hardware address or clientID), and when the lease expires. It also keeps a state value for each entry so it can track bindings that are not currently active (such as bindings that have been offered, released, or expired). The intent here is to remember previously assigned leases so clients will always get the same lease whenever possible. Finally, this table keeps track of when each entry was last updated so the server can re-use the oldest unused entry first ("Least Recently Used"). This table should normally be viewed as "Read Only", but you can use the button on the right to remove an obsolete lease binding. You can also copy rows to the clipboard as text for use in other panels.

The Status Table is stored in a separate file in the Preferences Folder called "DHCP Server Status". Any time a lease is granted to a client, the corresponding entry is written out to this file (committed to non-volatile storage). If the server is stopped for any reason, it can recover all the current lease information from here. You can also move your DHCP Server to another machine by copying this file and the corresponding IPNetRouter configuration file.

**STATIC CONFIG TABLE**

This table keeps track of any static address assignments the administrator wants to create. The fields are: (1) Network Interface - the IP address of which interface a request must arrive on [If a host moves from one network to another, we need to assign a different IP address depending on which network it is attached to. If a BOOTP relay agent is used, we match this network as well.]; (2) Lease Address - the address to grant; (3) What host or client this address is reserved for as specified by the Ethernet hardware address or ClientID. The ClientID will override the hardware address if specified in the table. The entries in this table can be edited in place by clicking in a cell. Use the buttons on the right to insert or delete entire rows. You can use Tab, Shift Tab, Return, and Shift Return to move the selected cell.

The Static Config Table is stored as part of your IPNetRouter configuration when you save from the file menu.

**DYNAMIC CONFIG TABLE**

This table keeps track of ranges of IP addresses that can be automatically assigned to hosts as needed and later reclaimed. The fields are: (1) Network Interface - the IP address of which interface a request must arrive on [Each network interface that accepts DHCP requests will normally assign IP addresses from a different range.]; (2) Starting Lease Address - start of range used for address pool; (3) Ending Lease Address - end of range used for address pool. The entries in this table can be edited in place by clicking in a cell. Use the buttons on the right to insert or delete entire rows. You can use Tab, Shift Tab, Return, and Shift Return to move the selected cell.

There are no restrictions on how many address ranges can be defined to serve from multiple interfaces or multiple ranges on a single interface. Any IP address in a dynamic address pool that overlaps with an address in the Static Config Table will be reserved for static configuration and not dynamically assigned. The Dynamic Config Table is stored as part of your IPNetRouter configuration when you save from the file menu.

Some clients may use BOOTP, an older subset of DHCP. BOOTP clients are normally not eligable to receive dynamic IP addresses because BOOTP does not provide any mechanism to reclaim an expired address. To avoid requiring the network administrator to create a static assignment for each BOOTP client, IPNetRouter will try to assign BOOTP clients a dynamic address if no static configuration is found and display a warning message in the log window. In order not to lose addresses permanently, the server will try to reclaim "BOOTP dynamic addresses" after 60 days.

**LEASE OPTIONS TABLE**

This table stores the other information that is given out to clients based on what network they are attached to. The fields are: (1) Network Interface - the IP address of which interface a request must arrive on [Each network

interface that accepts DHCP requests can potentially give out different information.]; (2) Network Mask - the network mask for this network; (3) Routers - the router addresses for this network; (4) Name Servers - the name server addresses for this network; (5) Default Lease Time - lease time granted if none is explicitly requested; (6) Maximum Lease Time - the maximum lease time to grant regardless of what is requested; (7) Domain Name - A default domain clients can use for DNS lookups; (8) Enable DHCP on this interface - DHCP serving can be selectively enabled on a per interface basis.

The Network Interface popup menu lists the IP interfaces which have a lease options entry currently defined. It selects which row of the table to view in the rest of the panel. If you select a new interface, IPNetRouter will try to fill in appropriate values for the subnet mask, router, name server and default lease times. The Lease Options Table is stored as part of your IPNetRouter configuration when you save from the file menu.

**MORE ABOUT DHCP DATA AND EDITING**

When you edit data in place, you are only editing a visible copy of that data on the display. You can undo any changes you make by pressing "Restore". To have your changes take effect, you must press "Apply". To save your settings to an IPNetRouter configuration file, you must still select "Save" from the File menu.

DHCP should be seen as a mission critical service since hosts on your LAN may not be able to use the Internet if your server is unavailable. DHCP is designed so that more than one server can be configured to provide the same information for reliability. Clients normally try to renew an existing lease and only the Server with a record of that lease will respond. If the existing lease cannot be renewed, the client may then try to obtain a new lease. The Server pings any new proposed lease address before offering it to a client to check that the address is not already in use. If the Server discovers an address conflict, it will mark the corresponding address in the Status Table as "In Use" or "Declined" and no longer try to assign it. You can see when the conflict was detected from the "Last Update" column in the Status Table. In order not to loose addresses permanently, the server will try to reclaim "In Use" or "Declined" addresses after 60 days.

Since the Status Table is kept as plain text in the file "DHCP Server Status", you can edit this table manually if desired.

## Resource Options

IPNetRouter supports a number of resource switches that can be used to further customize the application. These are stored in 'STR#' resource 131 named "Option Settings". The setting names and their defaults are listed here:

**SleepTime=6**

Parameter passed to WaitNextEvent

**LogFileName=IPNR.log**

Name of IPNR log file in Preferences folder

**ConfigFileName=Router Config**

Name of config file in Preferences folder to load automatically at application startup.

**PPPName=IPNetRouter**

Name of configuration in TCP/IP control panel to temporarily activiate for PPP connection.

**IPNetRouterDials=1**

Selects whether IPNetRouter will always tell PPP to dial before building a PPP link stream. Setting this to zero (0) allows PPP to decide when to dial on demand.

**DynamicBOOTP=1**

Allow BootP clients to receive an IP address from the dynamic address pool (like DHCP clients) instead of requiring a static configuration for each client.

**AuthorizationName=wireless-authority**

Name used to determine whether a wireless base station is permitted on this network.

**DHCPPingCheck=1**

Allows you to control whether the DHCP Server will ping a new address to verify it is not in use before offering it to a client.

**MapBroadcast=1**

Allows you to control whether IPNetRouter will automatically create a port mapping entry to forward directed broadcasts to your private LAN. This can be used to help locate services on your LAN using NSL/SLP.

---

SUSTAINABLE Softworks

products


# Introduction



IPNetRouter is the simple low cost way to share your cable modem, ADSL, or dial-up internet connection with your entire LAN. With IPNetRouter, you can use multiple IP interfaces at the same time (such as Ethernet and OT/PPP for example), and specify additional routes for communicating with more than one IP gateway. IPNetRouter is a complete solution including a built-in DHCP Server, NAT with inbound port mapping, and IP filtering to setup your own firewall. IPNetRouter's OT native design offers superior performance with greatly reduced cost and complexity.

A companion application IPNetMonitor provides a suite of integrated Internet tools which allow Macintosh users to verify router operation and performance.

**IPNetRouter Features**:

- OT native IP multihoming and routing
- Allows multiple users to connect to the Internet simultaneously using a single modem and dial-up account.
- Dial on demand with OT/PPP or FCR PPP (a.k.a. LinkUPPP or LeoMLP).
- Perfect for cable modems, ADSL, and multiple Ethernets.
- Connects Macs using LocalTalk via MacIP.
- IP Masquerading (RFC-1631 NAT) with in-bound port mapping.
- Built-in modern DHCP Server (RFC-2131, RFC-2132).
- IP filtering (firewall router). Built-in Windows-95, Windows-NT NetBIOS filtering.
- Passes PPTP transparently for Windows-NT VPN.
- Macintosh user interface makes these powerful networking tools more accessible.
- One low price: US$89 for unlimited users through a single

gateway!

System requirements:

MacOS 7.5.3 and Open Transport 1.1.1 or later

# Installation and Use

To install or remove the software, simply run the supplied Installer.

The Installer will place the "OTModl$Proxy" shared library file in your Extensions Folder (needed for IP masquerading & filtering). If you decide to remove this file from your Extensions Folder, please read the important note below.

> Note: IPNetRouter actually modifies the currently selected configuration (Cmd-K) in the TCP/IP Preferences file in your System Folder. Once modified, it is critical that the OTModl$Proxy module remain in your Extensions folder since OT won't be able to build this configuration without it. OTModl$Proxy is not really an extension. It doesn't patch anything or modify the operating system. It's a shared library that is loaded and linked by Open Transport when the corresponding protocol stack is created. You can undo the change to any configuration in your TCP/IP Preferences by: (1) Using the supplied Installer to uninstall IPNetRouter; (2) Creating a new TCP/IP configuration (by copying an existing one); or (3) Using ResEdit to edit your TCP/IP Preferences File and remove the 'crpt' resource IPNetRouter added.

A guide to "Getting Started with IPNetRouter" is available on the Sustainable Softworks website. To keep the download size small, this information is not included as part of the basic download package. Allow me to emphasize that TCP/IP networking is a lot more complex than most other software on your Macintosh. While we have tried to keep it simple, you will almost certainly need to read and follow the instructions on our website to make it work.

Help is also available on each of the various windows by clicking on the question mark symbol in the lower left corner. This is probably the best way to familiarize yourself with more advanced features of the software since you can experiment with the window while you read the description.

The Release Notes describe the latest features and additions.

# Installing Over Another Version of IPNetRouter

The standard installer is designed to install a newer version over a previous one. If you have any trouble, the following steps will insure the application is installed correctly.

1. Find any previous version of OTModl$Proxy in your Extensions folder and move it to the trash.
2. Remove any previous version of the IPNetRouter Application.
3. Run the correct IPNetRouter installer for your machine (PPC or 68K) to install the desired version of IPNetRouter.
4. Restart your computer (this step is very important). The installer only "suggests" you restart because it isn't always necessary, but you should do this if you are not sure.
5. Hold down the Option key when you launch the IPNetRouter application the first time after re-installing. Continue pressing the option key until the application comes up. This forces Open Transport to rebuild your link stream. Alternatively, configure IP masquerading, and then Restart your Mac yet again.

I recommend against using the "uninstaller" if you are just going to re-install the application. The uninstaller is for removing IPNetRouter completely and makes it harder to re-install. If you do use the uninstaller, step (5) above may be necessary. The purpose of step (5) is to give Open Transport a chance to rebuild your link stream with the Proxy module inserted. You can verify this was successful if you are able to use the monitor window in IPNetMonitor to monitor data on the interface you will be using to masquerade.

SUSTAINABLE Softworks

*products*

# IPNetRouter Installation and Use



IPNetRouter.PPC

To install or remove the software, simply run the supplied Installer.

Your system must meet the minimum system specifications (MacOS 7.5.3 or later and Open Transport 1.1.1 or later. MacOS 7.6.1 or later is recommended). If you attempt to install with a system which does not meet these specifications, not all the necessary components will be installed.

It is important to use the supplied installer to uninstall the program. Manually dragging the OTModl$Proxy shared library out of your Extensions folder can disable your TCP/IP configuration. If you do manually remove the OTModl$Proxy extension, please drag your existing TCP/IP preferences file to the trash and reconfigure your TCP/IP control panel.

A guide to "Getting Started with IPNetRouter" is available on the Sustainable Softworks web site. To keep the download size small, this information is not included as part of the basic download package.

Help is also available on each of the various windows by clicking on the question mark symbol in the lower left corner. This is probably the best way to familiarize yourself with more advanced features of the software since you can experiment with the window while you read the description.

The Release Notes describe the latest features and additions.

SUSTAINABLE Softworks

products


# IPNetRouter Release Notes



**Aug 18, 2000** (1.5.1)

DHCP Server: send responses to DHCP Server port (67) if via Relay Agent.

DHCP Server: ignore clients that match a static config with address=0.

DHCP Server: do not assign remaining lease time if less than 2 minutes.

DHCP Server: change default User Message for ACK response to empty.

Fixed bug in storing registration keys with a bad date.

**July 31, 2000 (1.5)**

Fix bug in accepting 68K only long hash key.

**July 14, 2000** (1.5)

Released as version 1.5

Update DHCP help text

Allow editing in Log Window

**July 2, 2000** (1.5c16)

Port Mapping: fix bug in mapping port ranges uniquely.

DHCP Server: include customizable user message in ACK responses.

Support 68K only registrations.

Help Window: provide translated text for German, Spanish, and French.

Help Window: add search website button.

**June 18, 2000** (1.5c15)

Proxy (OTModl$Proxy 1.9.4): fix possible leak for copied messages with ref_count>1.

Inserting Proxy module: don't give up if fixing an invalid configuration fails.

**May 29, 2000** (1.5c14)

Proxy: fixed bug that could mis-locate receive datagrams.

Gateway: change DNS Forwarding to default to off for better stability.

Gateway: change Remain Connected to ask PPP for the remote address to ping.

Dialup DNS Client: stop after 5 retries if connection fails.

IP Filtering: revert to block invalid or unrecognized datagrams for better stability.

IP Filtering: add filter action "NoDial".

Config File: add filter headings as a comment.

Gateway: restore "Better PPPoE Routing" so that MSS Clamp can be turned off if needed for compatibility (default to on).

**May 17, 2000** (1.5c13)

Allow FBA to run in trial mode.

Add date field to registration data.

Accept extended registration data as a single hex string.

Port Mapping Window: remember "Show Permanent Only" and monitoring state.

Proxy module: determine FastPath Header Size from response to DL_IOC_HDR_INFO instead of scanning header.

Save & Restore: fixed bug in determining buffer size, issue a log message if buffer overflows.

Gateway: Set MSS Clamp for PPPoE based on Interface MTU. This feature modifies TCP connection request packets to limit the MSS to MTU-40 to insure TCP segments will pass through a PPPoE connection that does not correctly handle oversize datagrams. Remove "Better PPPoE Routing" since it is now automatic.

**April 26, 2000** (1.5c12)

DHCP Server: set "DHCP Server On" checkbox correctly after wake from sleep.

DHCP Server: include customizable user message in NAK responses.

IP Filtering: pass unrecognized datagrams as default.

Gateway Window: save Exposed Host address upon Tab, Enter, or Return.

Gateway Window: add built-in Dialup DNS Client.

**April 5, 2000** (1.5c11)

Filter Window: increase max table size to 100 rows.

Filter Window: add ports for "dhcp-client" and "dhcp-server".

Gateway Window: changed "Limit MTU for PPPoE" to "Better PPPoE Routing".

DNS Forwarding: fixed bug that would save and restore an old DNS Forwarding entry.

Retry updating routes data if report size exceeds buffer.

About Box: update support URL.

**Mar 9, 2000** (1.5c10)

Gateway: Fix Exposed Host bug when set to Gateway.

Gateway: Skip Exposed Host translation in Single Ethernet configurations.

**Mar 8, 2000** (1.5c9)

Gateway: Fix bug that didn't display Exposed host after restore from a settings file.

Gateway: Fix bug that sometimes reset the Exposed Host address when the window is opened.

About Box: Update URL links and Copyright.

Dialing: update connection retry processing to work better with Mac OS 9.

**Mar 6, 2000** (1.5c8)

Gateway: Add "Limit MTU for PPPoE" setting.

Gateway: Add "Exposed Host" settings.

Interfaces: recognize interface names with an embedded space such as "108,Farallon Enet0".

DHCP: remove trailing null from Domain Name reported by DHCP Server (Option 15).

DNS Forwarding: translate the from address in DNS responses to make the gateway appear as the DNS server to any clients.

DNS Forwarding: disable extra NAT lookups when DNS forwarding is turned off.

**Feb 18, 2000** (1.4.8.2)

Fixed bug in Local NAT feature.

**Feb 4, 2000** (1.4.8.1)

Fixed restore when IPNR is set to "Configure only".

**Jan 28, 2000** (1.4.8)

Gateway Window: Add support DNS forwarding and DNS Deferral.

Gateway Window: The Local NAT function is enabled for DNS Forwarding.

DHCP: use DNS addr 0 to mean the currently configured DNS server for the gateway.

DHCP: change default for DNS to the gateway addr, 0.

IP Filtering: For ICMP datagrams, use Source Ports for ICMP Type and Dest Ports for ICMP Code. Match values of zero (no wildcards).

Increase memory available to IPNR application by 200K. Minimum Heap Size 800K, Preferred Heap Size 1000K.

Update copyright notices to year 2000.

Numerous structural changes to improve robustness.

- Use a separate RestoreThread to apply saved settings. Hold off opening any TCP/IP endpoint for 15 seconds (FBA version).

- Wait for routes data to stabilize before configuring routes (pause 1 second, check if PPP is connecting and wait for it to finish, update routes data from IP module and repeat if report will not fit buffer).

- Log each time DNS deferral is enabled or disabled.

- Make PPP retry delay a configurable option (PPPRetryDelay=5). Default to 5 seconds.

- Extend data buffer cushion for reading routes from 512 to 2048 bytes.

- Allow launching a PPP connection from the "kPPPLowerLayerDownEvent" state.

- When PPP connects, wait to get a DNS address before setting up DNS forwarding.

- Log if no DNS address available when trying to set up DNS forwarding, or forwarding a local DNS query.

- Do not try to restore interfaces if program is quiting.

- Make DNS forwarding default to on when #addOnly and no previous setting.

- Make use of NSL (Network Services Location protocol) a configurable option as STR# 131 (UseNSL=0). Default to off at this time.

- Report actual OT error if unable to read current Interfaces or Routes data during restore. Continue with restore since new settings are "#addOnly".

**Dec 17, 1999** (1.4.7)

General: Fix ASLM pool corruption conflict by reverting Proxy module to bump reference counts instead of using "writer buddy".

PPP: Fix bug in restoring PPP interface with correct IP address when the PPP link stream was not created by IPNR.

**Dec 12, 1999** (1.4.6)

General: for OT2.5 or later, use the 'blip' (below IP) resource instead of 'crpt' so IPNetRouter can coexist with other software that uses the 'crpt' resource.

General: fix bug in opening log file so application will not freeze if another copy is already running.

Register "service:router://x.x.x.x" with NSL 1.0 or 1.1.

DHCP: skip writing out the status file and log entry if DHCP server was not running when the application quits.

DHCP Lease Data: use DNS addr "0.0.0.0" to mean currently configured DNS of gateway.

TRCableModem: restore setting each time masquerading is enabled.

**Nov 17, 1999** (1.4.5)

DHCP: skip static configs in dynamic address pool.

Port Mapping: allow specifying protocol "Any".

Port Mapping: add option to automatically map directed broadcasts ('STR#' 131 "MapBroadcast=1"), default to on.

Subnet Calc: convert decimal input to a dotted quad.

PPP: do not remove and rebuild link stream when finishing connection under OT2.5

**Nov 4, 1999** (1.4.4)

Fix TR cable modem support: wait two seconds and try again if PPP returns the "Port Off Line" address (169.254.255.254) after connecting.

**Oct 25, 1999** (1.4.3.3)

PPP dial-on-demand: work around fragile IPCP in Mac OS 9.

**Oct 21, 1999** (1.5c7)

DHCP Server: add option to disable DHCP ping checking (STR# resource 131 "DHCPPingCheck=1").

DHCP Server Lease Options pane: fix Apply to apply all unsaved changes.

Routes Window: allow specifying destination Prefix Length (network mask).

Routes Window: update status after each operation.

Routes Window: work around bug in Mac OS 9 to display "Direct" routes correctly.

General: check saved window size against window size limits before restoring.

PPP: log name of PPP configuration used when changing the primary interface.

**Oct 15, 1999** (1.5c5, also released as 1.4.3)

DHCP Server: Fix endpoint recovery when certain errors occur (so server will not go deaf).

General: Detect if the user selects a previously unknown interface in the TCP/IP Control Panel and abort restoring the previous IPNR configuration.

**Oct 12, 1999** (1.5c4)

Interfaces Window: Do not display the internal loopback interface "lo0". Sort interfaces by "PPP", "MacIP", module name, device number, and interface number to work with Mac OS 9 (OT 2.5/Mentat 3.5).

DHCP: Restart DHCP Server after any unexpected endpoint notification.

Application Startup: Change FBA to more closely follow the same startup sequence as the normal UI version.

**Oct 5, 1999** (1.5c3)

Fixed bug in restoring IP masquerading in single Ethernet configurations when the gateways DHCP lease expires.

**Oct 1, 1999** (1.4.2)

Released as 1.4.2

**Sep 29, 1999** (1.5c2)

DHCP Server: Only match the ClientID if the ClientID field is specified at the server, otherwise try to match the 'chaddr' (client hardware MAC address). This allows the server administrator to override whether the ClientID field (which the user can change) will be used.

DHCP Server: Ignore leading null byte of ClientID inserted by many clients so names entered in the Static Config pane will match names entered in the TCP/IP control panel.

DHCP Server: Add log messages to indicate why a static config address could not be used.

DHCP Server: Try to assign a dynamic address to BOOTP clients if no static config is found.

DHCP Window: Allow editing in place with a single click.

Dial-On-Demand: Reset PPP controller when all six retries are exhausted to improve recovery.

General: Check TCP/IP Preferences File directly to avoid network setup when the Proxy module is already configured. This reduces launch time and works around network setup being out of sync with the TCP/IP Preferences File.

General: Ignore empty data for processing "#quit" directive.

Masquerading: Create a permanent mapping entry for directed

broadcasts.

Masquerading: Update "TR Cable Modem" setting to find the secondary NAT address under Mac OS 9.

**Sep 23, 1999** (1.5c1)

DHCP Server: Do not log DHCP messages received on an interface that is not enabled for DHCP.

DHCP Server: Increase suggested heap size by 100K to support more clients with the default memory partition..

DHCP Client: Detect new DHCP address if lease expires while IPNR is running.

NAT: As NAT table fills, try to age out open but inactive TCP connections more agressively. 24 hours for 1-127 entries, 12 hours for 128-255 entries, 4 hours for 256-500, 2 hours for >500 entries.

**Sep 14, 1999** (1.4.1)

Port Mapping: fix save and restore for protocol "GRE"

Fix possible crash when closing ARP window.

DHCP Server: Ignore all requests that arrive on an interface that is not enabled for DHCP.

**Sep 6, 1999** (1.4)

Release as version 1.4

**Sep 5, 1999** (1.4c23)

Only close streams to Proxy module if necessary for MacIP since this re-initializes the module. Reload filter table if module is re-initialized.

**Sep 2, 1999** (1.4c22.2)

Close control streams from CPortMapAction and CFilterAction when plumbing a new link stream to avoid OT errors when using MacIP.

Do not skip first entry of interfaces table when rebuilding configuration upon wakeup from sleep (this entry may not be the loopback device under OT2.5 (Sonata)).

DHCP: Do not skip first entry of interfaces table when matching a ServerID (this entry may not be the loopback device under OT2.5 (Sonata))

DHCP: Respond via Broadcast if previous response via Hardware Unicast was not successfully received.

Log File: Reset logical end-of-file upon open to avoid showing any previous text.

**Aug 26, 1999** (1.4c22)

Close control streams to Proxy driver when plumbing a new link stream to avoid OT errors when using MacIP.

DHCP: Fixed handling of DISCOVER and REQUEST messages when 'ciaddr' is not zero as used by Mac OS when self configuring IP.

DHCP: Broadcast response if requested regardless of 'ciaddr'.

DHCP: show correct Unicast IP address in log when 'ciaddr' is not zero.

Fix to restore masquerading if provider closes with new recovery from 1.4c21.

**Aug 19, 1999** (1.4c21)

Improved recovery when switching configurations in the TCP/IP control panel while IPNetRouter is running (this now works).

Converted some ALRTs to Log Messages to avoid crashing the FBA if certain exceptions occur.

Changed to not push OTModl$Proxy above MacIP to avoid possible conflict with Enable Local NAT.

**Aug 15, 1999** (1.4c20)

Add "#PPPName=<configName>" command to specify which PPP configuration in the TCP/IP Preferences File IPNetRouter should use.

Fixed access beyond end of block when reading DHCP status.

Fixed improper deletion of OT Config when PPP fails to connect.

Added 'STR#' resource 131 with option settings (SleepTime=6, LogFileName=IPNR.log, ConfigFileName=Router Config, PPPName=IPNetRouter, IPNetRouterDials=1).

Restructured the logging function to write out status messages to a log file as well as displaying them in the Log Window. The default log file is named "IPNR.log" in the Preferences Folder.

**Aug 2, 1999** (1.4c19)

DHCP: Several bug fixes: (1) Use time on existing lease if any for Discover/Request sequence or Request from client in INIT-REBOOT state; (2) Extend lease time if client is RENEWING or REBINDING

even if client doesn't explicitly request more time; (3) Send NACK in response to verify (Request from client in INIT-REBOOT) if requested address is not valid for this network; (4) Allow static configs to override existing lease assignments; (5) Do not send address time option for DHCP inform; (6) Reclaim Client_ID lease bindings for IP addresses that are no longer in the dynamic pool so these clients will get a new valid lease; (7) Added delete button to DHCP Status pane to manually remove old lease bindings.

Filter Window: Changed "pass" filter action to ignore datagrams that don't match since it is easy to block such datagrams with other filters. This allows pass filters to open holes in a set of "block" filters without blocking datagrams that don't match any filter.

**July 27, 1999** (1.4c18)

Fixed "stale handle reference" when reading data from IP module.

DHCP: fixed bug in block size when writing out DHCP Status.

DHCP: extend lease time if client is RENEWING or REBINDING even if client doesn't explicitly request more time.

Fixed dispose handle bug when opening Gateway Window.

**July 18, 1999** (1.4c17.2)

Gateway Window: when dial on demand is unselected, open PPP before trying to connect so PPP can decide when to dial for the first time. Press the Option key while adding a PPP interface to temporarily override this behavior.

DHCP: fixed bug in log of DHCP message type when the type is unknown or unspecified.

**July 10, 1999** (1.4c17)

Gateway Window: add checkbox to disable using OT Network Setup.

Allow Gateway options to be set or cleared using #addOnly.

Improve error handling when unable to open a PPP control endpoint.

Do not generate pseudo Default-Gateway for non PPP interface.

**July 1, 1999** (1.4c16)

Interfaces Window: Allow masquerading to be enabled on a single interface when multiple interfaces are defined on the same port.

NAT: Do not translate other interfaces from the same network on a single port. This is to allow multiple public IP addresses on the gateway

machine.

Filter Window: change protocol port popup to only set the destination port.

General: remove Default-Gateway safely when Provider Is Closing.

**June 28, 1999** (1.4c15)

Gateway Window: Add checkbox to configure for "TR Cable Modem".

General: Do not allow Private Network that overlaps primary interface.

General: Suppress log message when IP address is not available from Remote Access.

**June 20, 1999** (1.4c14)

General: changed KOTProviderWillClose processing to be more reliable when DHCP lease expires.

**June 15, 1999** (1.4c13)

Proxy module: add support for QuickTime 4 Streaming (RTSP).

Proxy module: fix support for port ranges and additional public addresses.

Port Mapping: allow specifying port ranges as "A-B" where "A" and "B" are protocol port numbers.

General: improve how threads are terminated.

General: updated to CW-Pro Release 4.

General: use "Import Weak" for Appearance Lib.

Startup: open the settings document "Router Config" in the Preferences Folder if found (especially useful for FBA version).

**May 21, 1999** (1.4c12)

Proxy module: fix bug that could lose port ranges for permanent mapping entries.

Proxy module: fix bug in performing NAT on LAN interfaces (Enable Local NAT).

Proxy module: designate "Proxy" and "Proxym" (the module and driver part of OTModl$Proxy) as install buddies instead of modifying reference counts to prevent unintended unloading.

DHCP Server: allow lease times to be specified in minutes (to facilitate

testing).

DHCP Server: reduce grace period for expired leases from 5 to 2 minutes.

General: Shorten recovery time when Provider Closes unexpectedly (DHCP lease expires), wait one second and then open an endpoint to block until a new lease is established.

General: Use OT Configuration Database if available instead of modifying TCP/IP Preferences File directly to insert Proxy module. Perform this operation at application startup before opening any endpoints since it may rebuild the TCP/IP configuration (pressing Option during startup will force the stack to rebuild).

General: always terminate threads the same way to avoid a possible race condition when quiting the application while PPP is connected.

General: make version number and version strings more consistent.

General: Fix PowerPlant LInterruptSafeList to work with future MacOS version.

General: Update CIPNumberField.cp to work with Metrowerks IDE 3.3.

**April 26, 1999** (1.4c9)

General: improve recovery when provider closes.

DHCP Server: fix renewal to recognize ciaddr.

DHCP Server: catch improperly configured address before offering to client.

DHCP Server: automatically resume serving after network error.

**April 12, 1999** (1.4c8)

Fixed bug in Enable Local NAT for one way cable modems.

IP Filtering: allow filtering on both source and destination within a single rule.

IP Filtering: allow filtering on TCP ACK bit.

ARP: fixed bug in displaying Interface Names that contain comma.

DHCP Server: changed "Save" button to "Apply".

**April 3, 1999** (1.4c7d)

DHCP Server: fixed bug that prevented server from recognizing multiple Ethernets.

DHCP Server: offer a different IP address if a requested address is not available.

DHCP Server: ping previously assigned addresses that have been released or expired.

DHCP Server: allow a released binding to be renewed.

DHCP Server: fixed bug in handling Client IDs.

DHCP Server: allow copy from DHCP Status Table.

Changed command shortcuts to use Cmd-R for ARP and Cmd-H for DHCP.

**April 3, 1999** (1.4c7c)

DHCP Server: add workaround for bug in IP_RCVIFADDR to support dual Ethernets.

DHCP Server: refresh Network Interface popup when lease data settings change.

DHCP Server: write DHCP Server Status in 512 byte blocks.

DHCP Server: change Interface popup in Lease Data to only show existing entries.

DHCP Server: don't erase existing settings for #addOnly.

DHCP Server: fix editing dynamic config range.

DHCP Server: save and restore "Use Verbose Logging" checkbox.

DHCP Server: fix possible freeze when saving.

ARP Window: fix bug in initial data displayed.

Port Mapping: changed the meaning of "static" to specify a fixed apparent address. This allows mapping additional fixed IP addresses to hosts on your LAN. Manually configured NAT table entries that do not age are now called "permanent".

Port Mapping: protocol port zero is now interpreted to mean map all ports.

**March 30, 1999** (1.4c7)

DHCP Server is now working (Yay!)

IP Filtering: do not dial on demand for transmitted FIN segments (to prevent improperly terminated connections from redialing).

**March 2, 1999** (1.4c6)

Fixed bug introduced in 1.4c5 that caused Port Map entries to dispear when PPP reconnected.

IP Filtering: filtering is performed before Network Address Translation (NAT) for outgoing packets, and after NAT for incomming packets (so LAN clients can be distinguished).

Added DHCP Window. Server implementation is in progress ("enable" just logs DHCP requests).

**February 20, 1999** (1.4c5)

Fixed bug where inbound port mappings could be lost the first time IPNR is launched.

Fixed bug where secondary NAT address (for one-way cable modems) could be lost the first time IPNR is launched.

Enhanced logging to show each command in full.

Changed the way IPNR reads from IP to improve robustness.

Allow interfaces and routes table to hold up to 200 entries.

Added ARP tool (still in progress). Allows IPNR to support proxy ARP.

**February 17, 1999** (1.4c4)

Fixed recovery when DHCP lease expires while connected.

Holding the Option key while selecting "Interfaces" from the Window menu will remove and then restore all interfaces (as if the DHCP lease had expired).

Improved error checking when reading from IP module.

Changed support address in the About Box to "support@sustworks.com"

Fixed error message when unable to age the NAT table.

**January 29, 1999** (1.4c3)

Interfaces Window: fixed to remember Default-Gateway when changing a PPP configuration.

**January 24, 1999** (1.4c2)

Command Files: added parameter to set interface MTU (\mtu=xxx).

NAT: do not age TCP entries unless they have been closed from at

least one end.

**January 22, 1999** (1.4c1)

Fixed bug in Local NAT for TCP connections (OTModl$Proxy 1.6.3).

Enhanced recovery when DHCP lease expires while connected.

**January 17, 1999** (1.3.2)

Masquerading: removed alert before modifying TCP/IP Preferences File from 1.3.1 below.

**January 14, 1999** (1.3.1)

Interfaces Window: Fixed bug in remembering Unnumbered setting.

Gateway Window: Enable NAT on local interface is now a separate selection.

Port Mapping: fixed bug in restoring port ranges.

Masquerading: notify user when restart may be required for masquerading to work.

Installer: always update OTModl$Proxy and recommend restarting.

**January 4, 1999** (1.3)

Released as version 1.3

**January 2, 1999** (1.3c15)

Changed to remove and then restore interfaces when DHCP lease expires while connected.

Restore previous behavior of always using PPP negotiated IP address when available.

Reset 21-day trial period for all users.

**December 30, 1998** (1.3c14)

Fix the way DHCP setting is saved.

Don't remove interfaces if provider closes to continue working when DHCP lease expires.

Preserve port mappings when using #addOnly command file.

Override PPP negotiated address if an IP address is explicitly specified.

Changed to enable NAT on local interfaces more reliably including

single Ethernet configurations.

**December 15, 1998** (1.3c13)

NAT: do not override static port map entries.

Port Mapping: allow a single table entry to map a range of ports.

Port Mapping: added support for Telco Return cable modems.

Make all command verbs case insensitive.

**November 22, 1998** (1.3c12)

Fixed bug in restoring port map settings for PPP connection.

Ask to save changes at quit only if user has changed settings.

Add option to disconnect PPP at quit in Gateway window.

Added "#connect" and "#disconnect" command verbs.

**November 5, 1998** (1.3c11)

Treat any unrecongized transport protocol as a security wrapper.

Fixed bug in restoring port map settings.

Ignore empty routes data.

**October 23, 1998** (1.3c10)

Updated to work with Mac OS 8.5.

Fixed bug in setting up masquerading when using two interfaces on a single Ethernet.

**October 21, 1998** (1.3c9)

Ensure PPP link stream is built at system task time, retry if first attempt fails.

Fixed DHCP processing.

Fixed crashes and instability problems that resulted from restructing below.

Allow IP masquerading on multiple interfaces (up to four).

Remember which windows were open when closing a document.

Include short version string in IPNR saved configuration file.

Interpret "-interface" in config file to remove the corresponding interface if any.

Allow TCP NAT entries to age out promptly when FIN packets arrive out of order.

IP Filtering: fix bug in matching a single port number.

Move all user modifiable data to separate Data Objects to simplify the design and prepare for making IPNetRouter a scriptable faceless background application.

Restore: Fixed bug when restoring "#addOnly" routes.

Restore: Add or remove filter entries to match config unless "#addOnly".

**September 15, 1998** (1.3c7)

Improve error recovery during PPP connection attempt. If all retries fail, wait 10 minutes and start again.

Fixed bug that could result in duplicate interface names when more than 32 registerd ports.

Add logging to show User Port Name, Actual Port Name, and Interface Name.

When restoring settings, if no Interface Name is given, use the corresponding Interface Name based on the port name.

Rescan port registry before each restore settings operation.

**August 31, 1998** (1.3c6)

Fix DHCP feature to not over write new routes.

Filter Window: restore filters in same order as saved.

Filter Window: avoid interaction with protocol port name popup in Port Map window.

**August 14, 1998** (1.3c5)

Fix DHCP feature to not over write existing interface.

Filter Window: fix NetBIOS DNS querry filtering to reject invalid IP datagrams.

Filter Window: allow IP address 0.0.0.1 to represent a dynamically assigned address.

Filter & Interface Window: allow user or actual port names.

Restore Interfaces: added "noGateway" parameter to allow restore of

dynamic PPP interface without creating a Default-Gateway.

Improve error recovery when PPP connection attempt fails.

**August 10, 1998** (1.3c4)

Improve error reporting when updating display tables.

Fixed bug in save and restore filter port number.

**July 31, 1998** (1.3c3)

Fixed bug in save and restore filter settings.

**July 30, 1998** (1.3c2)

Fixed problem in configuring Link Stack for MacIP.

IP Filtering: added protocol port name and ICMP type popup menus.

Port Mapping: added protocol port name popup menu.

**July 29, 1998** (1.3c1)

Added support for "IP Filtering".

Perform NAT on datagrams from LAN when IP Forwarding is set to "Always".

**July 15, 1998** (1.2.1)

Fixed bug when reconnecting an unnumbered interface with a duplicate IP address.

Changed NetBIOS filtering to delete NetBIOS DNS querries sent to the on demand interface.

**July 6, 1998** (1.2)

Released as version 1.2

**July 4, 1998** (1.2c6)

Fixed bug in remembering static translation entries.

Fixed bug in reclaiming existing port map entry for ICMP or GRE packets.

Changed "Show PPP Dialogs" to not enable connection reminders.

**July 2, 1998** (1.2c5)

Fixed bug in starting masquerading when no document is open.

Fixed bug in restoring an unnumbered interface when "Show PPP

dialogs" is selected.

Added support for GRE packets to allow PPTP through the gateway.

**June 30, 1998** (1.2c4)

Minor changes to NAT module.

Restructured Save and Restore to be more reliable.

Command-period or Connection->Reset now aborts a connection attempt in progress.

**June 26, 1998** (1.2c3)

Restore static translation entries every time masquerading is started.

Don't force modal connections.

**June 25, 1998** (1.2c2)

Added support for "inbound port mapping".

Improved error reporting when restore fails.

Force modal connection during restore so connection completes before trying to restore routes.

**June 16, 1998** (1.2c1)

Improved retry algorithm for PPP connections.

Tested and improved support for FCR LinkUPPP.

**June 14, 1998** (1.1)

Fixed to save current Gateway settings when creating a new document.

Released as V1.1

**June 11, 1998** (1.1c1)

Fixed to save and restore host routes.

Fixed to allow PPP to redial if line is busy.

Fixed to identify masquerading correctly when two interfaces use the same IP address.

Report source IP address of on demand traffic.

Allow dial on demand to be turned off.

**June 10, 1998** (1.0.2)

Fixed to restore PPP interfaces with static IP address.

Fixed to retry building PPP link stack if connection attempt fails.

**May 31, 1998** (1.0)

Released as V1.0 (Yay!)

[End of Release Notes]

SUSTAINABLE Softworks

products


# IPNetRouter - Frequently Asked Questions

Will your router work on the same machine as MacDNS and AIMS?

Will IPNetRouter work on 68K machines?

What is the performance like on older machines?

Do you plan to provide both dial out PPP and dialin PPP services?

Can Filemaker Pro Server use more than one IP on the gateway machine?

Does IPNetRouter "multihome" Appletalk?

- What is MacIP?
- Since IPNetRouter does not multihome Appletalk, how can I get my Appletalk printer to be shared on a routed LAN?

How come Software Updates and some installers do not work with IPNetRouter on the gateway machine?

When is a good time to download IPNetMonitor?

How do I get x-windows to display on a LAN client?

Can I dial into AOL and use IPNetRouter to route to my private LAN?

Like to know how to get Realplayer to work behind NAT? Click here!

About Apple Remote Access (ARA) and OT/PPP

IPNetRouter Parameters You Can Change with ResEdit

Do you have any plans to integrate S/WAN into the router?

Can I get IPNetRouter to work with VPN/TunnelBuilder/PPTP?

How does your router compare with Vicomsoft's Gateway?

- How does IP addressing work on the gateway machine?
- Where is the "mirror port"?

How does your router work with OT1.3 and the IP Secondary Address file?

I still can't get it to work, what should I do? See the Troubleshooting page on our web site. Or, you might want to see IPNetRouter's online guide.

---

**What's the latest status of IPNetRouter** (08-Aug-2000)

1.5 was released at MacWorld Expo NY 2000. New features include "Better PPPoE Routing", Dynamic DNS support, and an exposed host feature, along with numerous bug features and other improvements. There is a cheaper 1.5 license now available for a 1.5 68k only key.

OS X support is in the preliminary investigation stages. We will make an announcement on the nettalk and netannounce mailing lists when more info is available. Apple has officially delayed OS X release until 2001.

See the current IPNetRouter download page and release notes for more information about the latest available versions.

A new configurator application is also now available that works with IPNetRouter 1.5. Please see the IPNetRouter download page on our site for more info.

**Is there an upgrade or downgrade path for the new 68k only license?** (25-Jul-2000)

Contact us via our web support page if you would like a license upgrade path for the 68K only license. If you eventually plan to upgrade your gateway machine to a PPC platform purchase the higher priced "universal" license. No matter what, this will cost you less in the long run.

If you already have the full universal license you may not "downgrade" to a 68k license, but your older key will continue to work on both PPC and 68k.

If you replace a 68k machine with a PPC gateway, you can always purchase the cheaper 68k license and transfer the older PPC/68K license to the new machine for use there. The 68k machine can then be used as another router/gateway or sold. Keep the 68k machines in service! We hope the lower 68k only price will encourage everyone to do so rather than throwing these perfectly good machines out.

**Is there an easier way to configure IPNetRouter?**

Yes, we now have a web configurator application that you can use if the IPNetRouter Interface Window is too confusing. It won't solve any hardware problems you might be experiencing but it should get you up and running quickly if IPNetRouter's user interface is causing confusion.

**Is there a way to suppress the initial dial-out of the PPP connection until it is actually needed?**

Not for Mac OS 8.6 and earlier.

I have tried to work around this, but there isn't a good solution without modifying the way the OT/PPP configurator works. In order to have IPNetRouter detect whether there is PPP traffic, the IP module in Open Transport needs to have a corresponding interface that it can route traffic to. When this interface is defined by building the corresponding link stack (the chain of protocol modules from IP to the data link provider). The PPP configurator runs and tries to establish a connection. If you set PPP to not connect automatically when opening TCP/IP applications, the PPP configurator fails to create the corresponding link stack.

For Mac OS 9 (OT2.5.x), you can suppress the initial dial out by unselecting "Dial on Demand" in the gateway window, and using ResEdit to change 'STR#' resource 131 "Option Settings" from "IPNRDials=1" to "IPNRDials=0". I plan to make this easier in a future version.

**IPNetRouter won't disconnect, or keeps trying to reconnect. What can I do to prevent this?**

There are two reasons IPNetRouter might remain connected or keep trying to reconnect:

1. "Remain Connected" is selected in the Gateway window.
2. Some process on your network is generating traffic for your PPP interface. The IPNetRouter Log Window and tools in IPNetMonitor can help you track this down. A prime source of this problem is the Network Time Server feature present in MacOS 8.5 or later Date and Time control panel, either on clients or the gateway machine.

**The first time a client on my LAN tries to access the Internet when the PPP connection is down, it reports failure before the PPP connection can be reestablished. Is there anything I can do to prevent this?**

A clever work around for this problem is to enter the IP address of your Name Server(s) more than once in the TCP/IP control panel. This way when the Local Resolver times out after trying each Name Server the first time, it will proceed down the list and try again giving your router a chance to finish connecting to the internet before giving up.

**I get an error that another machine is using the same IP address, what can I do?**

If you connect with a cable modem using the "Single Ethernet" configuration your LAN is directly attached to the cable modem network. It is quite possible someone else on the cable modem network is already using the IP addresses I suggested in the getting started examples. In this case you can either choose different IP addresses, or switch to a "Dual Ethernet" configuration to isolate your LAN from the cable modem network.

**Will IPNetRouter work with more than two physical ethernet interfaces?**

Yes. You can use multiple ethernet cards in machines that support them. See the Troubleshooting page for more info on Ethernet card conflicts.

**Can I use IPNetRouter to share my cable modem connection with VPC?**

IPNetRouter is an Open Transport native IP router. That is, it enables Open Transport to perform routing.

VirtualPC bypasses Open Transport completely and talks directly to the Ethernet hardware so it can run the PC networking stack. The best way to look at VPC is as a separate stand alone PC that shares your Macs hardware.

To use VPC through IPNetRouter, you would need to run VPC and IPNetRouter on separate Macs so VPC can connect through your gateway that runs IPNetRouter.

How to configure VirtualPC for IPNetRouter link

**How do I configure my Windows 95 machine to use my Mac as its gateway?**

You will have to be somewhat familiar with the Windows95 Network control panel. Here is basically what you need to do:

1. Open the Windows Network control panel.
2. Select the TCP/IP line for the Windows Network card.
3. Open the Properties of this device. You should see several Tabs (IP Address, Gateway, and DNS configuration are the ones we are interested in).
4. Goto the IP Address tab. Give your machine an IP

Address within the range of your Private LAN (for example, 192.168.0.2).
5. Goto the Gateway tab. Enter the IP Address of the machine running IPNetRouter (typically 192.168.0.1). Click the Add button. This address should now be added to the list of Installed Gateways.
6. Goto the DNS Configuration tab. Enter a host name (any name, without spaces, for this machine, such as "Win1", will do). For the domain, enter the domain of your Internet service provider, such as mindspring.com. Then go to the DNS part and enter your DNS server addresses (as provided to you by your ISP). For each address, click the Add button.
7. Click the OK button. This will take you back to the Network control panel. Click the OK button again. You may have to insert your Windows CDROM (or diskette) and restart Windows.

This should do it! (You'd think!)The only other thing you may have to check is that your Internet connection is setup (on the Windows machine) to use the Ethernet card and NOT a dial up connection. There is another Windows control panel to let you select how you want to connect to the Internet.

**How do I configure IPNetRouter to work with Timbuktu?**

In order for Timbuktu Pro to work through a firewall (IP masquerading), UDP port 407 and TCP ports 1417 through 1420 must be open. Timbuktu Pro uses UDP port 407 for connection handshaking and then switches to the TCP ports for Timbuktu Services: Control (1417), Observe (1418), Send (1419), and Exchange (1420). Chat, Notify, and Intercom use Dynamic TCP ports.

You can setup these ports using the Port Mapping window in IPNetRouter. See Inbound Port Mapping for details.

**Does IPNetRouter work with FreePPP?**

IPNetRouter does not support dial on demand with FreePPP. You must use OT/PPP or FCR PPP (a.k.a. LeoMLP) for auto connection to work.

Notice that Apple's current Open Transport implementation does not support "dial on demand", but rather dials when a client opens a TCP/IP endpoint causing PPP to be configured. With IPNetRouter, the client opening a TCP/IP

endpoint might not be on the same machine as IPNetRouter, so IPNetRouter needs to detect traffic for your PPP interface and instruct PPP to dial out if it is not already connected. IPNetRouter currently knows how to do this for OT/PPP and FCR PPP. FreePPP requires a different API which IPNetRouter has not yet implemented.

**Does IPNetRouter work with ISDN cards from Hermstedt that use LeoTCP?**

Yes, but you must obtain LeoTCP/OT version 3.03 or later (version 3.01 will not work).

**Will your router work on the same machine as MacDNS and AIMS?**

I don't see any reason why not. The router is Open Transport native so it does not require a dedicated server. It uses the existing OT protocol stacks.

**Will your router work on 68K machines?**

Yes, IPNetRouter requires a minimum of a 68030 CPU, 7.5.3 or later and a minimum of Open Transport 1.1.2 or later.

**What is the performance like on older machines?**

IPNetRouter itself is very fast by virtue of being Open Transport native. Routing is handled by OT within the kernel. Even a Mac IIsi (68030) can provide good performance depending on your network configuration.

Older machines have other limitations however. Trying to use LocalTalk and a serial port at the same time will cause performance to suffer on machines that do not have DMA serial ports (non AV 68Ks). On these machines, LocalTalk disables interrupts for so long that your serial port can lose characters causing frequent retransmissions. The maximum serial port speed is 56K.

Routing between two IP interfaces on a single Ethernet can also reduce performance (Cable modem with single Ethernet for example). In this configuration, the Ethernet must transfer each packet twice so the second transfer has to wait for the first transfer to get out of the way.

Older machines typically do not have built-in Ethernet which can be another limiting factor.

**Do you plan to provide both dial out PPP and dialin PPP services?**

The router is actually a User Interface to configure forwarding in Open Transport, a routing protocol module to handle routing table updates, and a STREAMs module to provide proxy services (Network Address Translation) to allow multiple hosts to hide behind a single public IP address (single user ISP account).

This design allows it work with any OT compatible data link provider (such as PPP). At this time, Apple's PPP Control Panel (Remote Access) provides fully functional dial-out capability for IPNetRouter.

For dial in capability, you must run ARA server on a client machine. ARA cannot be run on the same machine as IPNetRouter due to a conflict with the way the TCP/IP control panel is used by both applications.

With MacOS 9, the ARA personal server is included. Hence you can create a "mini-ISP" dial-in service using a client machine and have your remote machine connect to this client via PPP (Remote Access). This remote machine will then have direct access to all machines on your LAN as well as to the Internet through IPNetRouter which is functioning as a gateway to this LAN.

**Can Filemaker Pro Server use more than one IP at one time on the gateway machine?**

Unfortunately, Filemaker Pro Server was designed only to recognize the IP address in the active TCP/IP control panel configuration. It cannot bind to more than one. As far as we know, the FMP developers never considered the dual NIC case. For a shared server/gateway Mac, the only workaround we know of is to point your Appletalk interface at one NIC and use the active TCP/IP configuration for the other.

**Does IPNetRouter "multihome" Appletalk?**

IPNetRouter does not effect your Appletalk network in any way. Whatever Apple permits Appletalk to do, it will do. What does this mean? This means that whatever interface is selected in the Appletalk control panel is the one on which Appletalk will be active. IPNetRouter does not use Appletalk, nor can it manipulate the Appletalk transport layer. If you are experiencing problems getting your Appletalk printer to work, it is extremely unlikely that it has much to do with IPNetRouter. The same goes for

Filesharing over Appletalk.

**What is MacIP?**

MacIP is Apple's way of transporting TCP/IP packets over Appletalk. While IPNetRouter can send IP packets to a MacIP interface, IPNetRouter cannot send Appletalk packets themselves anywhere. The Appletalk Control panel determines where Appletalk packets are sent.

**Since IPNetRouter does not multihome Appletalk, how can I get my Appletalk printer to be shared on a routed LAN?**

If you have an Appletalk capable printer, you can use Apple's printsharing in many instances. You may also find Apple's unsupported Localtalk Bridge software to multihome Appletalk between an ethernet and a localtalk interface handy. Farallon and Asante make Localtalk to Ethernet converters to put a localtalk printer on an ethernet network.

**How come Software Updates and some installers do not work with IPNetRouter?**

Due to limitations in Apple's OS, IPNetRouter must run as an application. This makes IPNetRouter incompatible with any installer or updater that requires all other applications to be quit during software installation. Fortunately, this incompatibility does not effect client machines behind your IPNetRouter gateway. You can test whether an installer or updater may be incompatible with IPNetRouter by running it on a client machine first--if the Finder Desktop is lost during software installation, it is not one you should run on your IPNetRouter gateway without first quitting IPNetRouter.

On your IPNetRouter gateway, you can workaround most problem installers and updaters by quitting the IPNetRouter application and running the installer or updater as you normally would. After you have verified that the software has been configured properly, you may launch IPNetRouter once again and go back to using your Macintosh as a router.

**When is a good time to download IPNetMonitor?**

If you are having problems getting your IPNetRouter configuration to work, we sometimes suggest downloading IPNetMonitor to test what and how far your gateway or

client can "see" on your local IP network. IPNetMonitor is very often useful in troubleshooting partial connectivity problems. Read the troubleshooting page for when this might be appropriate.

For one-way telco cablemodem/DSL users, we highly recommend downloading and familiarizing yourself with IPNetMonitor. If you mention to your ISP that you have a tool that can scan address ranges, traceroute, and bind DHCP client addresses they may be more willing to help you troubleshoot over the phone. It also saves time in seeking tech support from us as we often request that you install this to help figure out what's right and wrong with your IPNetRouter one-way telco configuration.

**How do I get x-windows to display on a LAN client?**

X-windows uses ports 6000-6063 to communicate with display procs. If you map these ports to a LAN client, you should be able to get that client to work as an x-display terminal, provided you tell your remote unix host that the x-display IP address is your public gateway IP.

There is probably a way to get this to work for multiple clients on your LAN by mapping a subset of the above ports to different IPs, although we have not figured out how to do it as yet. If you figure out how, please submit an info support ticket explaining how to setup the remote display host. We'd love to share that info with other users!

**Can I dial into AOL and use IPNetRouter to route to my private LAN?**

No, IPNetRouter supports standard OT/PPP negotiation. AOL 5.0 and earlier uses a propietary implementation of PPP to negotiate connectivity to AOL's servers. Consider getting a service provider that uses industry standard connectivity if you would like to use IPNetRouter.

**About Apple Remote Access (ARA) and OT/PPP**

IPNetRouter supports ARA 3.x and OT/PPP for connecting your gateway to the internet.

OT/PPP 1.0.x is a derivative of ARA 3.0. You can download OT/PPP 1.0 from here. 1.0.1 was included with 8.1. ARA 3.x is available for installation as a client install on 8.5 and later MacOS installation CDs. Apple no longer appears to

be selling ARA 3.0 software but you may be able to find copies in a local software store--3.0.x runs on all machines supported by OT 1.1.2. Both OT/PPP and ARA 3.x are scriptable although 3.x has many more supported commands. Versions of ARA before 3.0 do not support PPP and are therefore not useable by IPNetRouter as a dialup client.

**Do you have any plans to integrate S/WAN into the router?**

I've been following S/WAN with some interest as it seems like an ideal feature to include in a router making it easy to secure a stub network. I would like to include it, but I'm focused on doing other things first.

**Can I get IPNetRouter to work with VPN/TunnelBuilder/PPTP?**

Yes and no. IPNetRouter supports transparent IPSec client tunnelling. In many cases, you can use at least a single client behind IPNetRouter's gateway to connect to a remote service. IPNetRouter does not support PPTP/VPN connection as an interface directly on the gateway machine. We have had reports of TunnelBuilder 5.0.9a working as a gateway interface but have not yet grokked a clear method for doing so. Regrettably, you may have to get a non-VPN/TunnelBuilder/PPTP connection to the internet first if you require IPSec support for a client machine routed to by IPNetRouter.

In order to get a secure client machine connection to work, you may have to map certain ports in IPNetRouter. The most recent releases of IPNetRouter have an exposed host feature that you can easily use to test whether port mapping is a likely solution for a VPN client you may be experiencing problems with. Some IPSec software, Interport's Mac client software for instance, can be configured to easily work behind any standards compliant NAT router, which IPNetRouter happens to be. IPNetRouter's Gateway window help "?" button provides more information on how to use the exposed host feature to map all IP ports from your gateway to a single client quickly permitting verifying whether port mapping is a solution to your IPSec client's connection problems.

We are aware that many users want IPNetRouter to support PPTP negotiation directly on the gateway. Unfortunately, PPTP is not an industry standard protocol

but a proprietary protocol controlled by Microsoft. This makes PPTP a moving target with regard to technical implementation and licensing. Although many ISPs and businesses have chosen the PPTP path for security and user monitoring purposes, we currently have no plans to implement PPTP support in IPNetRouter. We support standard IPSec transparency. We have also considered implementing other methods of secure tunnelling on the gateway but have nothing to announce at this time.

**How does your router compare with the VICOM Gateway (www.vicomtech.com)?**

IPNetRouter is OT native, so it is much faster, and more compatible with other OT software. It's also simpler and less expensive.

Open Transport is based on Mentat/TCP, the same networking infrastructure used by Solaris. Mentat/TCP is already capable of forwarding (routing) within the kernel, but Apple hasn't enabled this yet. What I have written is a UI to configure interfaces and routing, and a proxy module to allow multiple hosts to hide behind a single dynamically assigned IP address. [Along with a set of IP tools I've already released]

In contrast, the existing VICOM product has to replace the network stack with its own PPP, PPPOE, and IP network code.

**How does IP addressing work on the gateway machine?**
**Where is the "mirror port"?**

IPNetRouter doesn't need a "mirror port" because it is Open Transport native. VIG has its own IP stack which does routing and then presents one of its router ports to Open Transport as if it were a simple driver.

With IPNetRouter, the IP module in Open Transport does the routing and talks directly to the driver for each physical port on your machine.

**How does your router work with OT1.3 and the IP Secondary Address file?**

It may help to realize that IPNetRouter is mostly a configuration utility for sending IOCTL commands to the IP module in Open Transport. It is similar to the TCP/IP

control panel, but exposes more features of Open Transport allowing you to configure and bring up additional IP interfaces.

The Configure Interface and Configure Routes window in IPNetRouter query the IP module in Open Transport to show the actual interfaces and routes that IP knows about. Based on this design, it doesn't matter what other programs may have configured IP interfaces. IPNetRouter simply shows the current configuration and allows you to modify it.

OT1.3 or later does not actually provide full IP "multihoming", the ability to be homed on more than one IP network as a user configurable feature. It provides what Apple has dubbed "single-link multihoming", the ability to have more than one IP address on a single physical network link. IPNetRouter is not restricted in this way and allows you to configure IP interfaces on different physical network links. You can use IP over two Ethernet interfaces for example, or Ethernet and dial-up PPP at the same time.

So to answer the question, it works just fine thank you.

Back to the top

---

SUSTAINABLE Softworks

*products*

# IPNetRouter Pricing

IPNetRouter.PPC

• **IPNetRouter**

Overview

Descriptions and Specifications

**Readme**

**Download**

**Installation Guide**

Getting Started with IPNetRouter

Release Notes

Troubleshooting

FAQ

**Pricing/License**

| | |
|---|---|
| Single Gateway Machine running IPNetRouter 1.x for PPC and 68K (unlimited LAN clients) | $89 |
| Site License (any number of gateway machines running IPNetRouter) | $3000 |
| (NEW) Single Gateway Machine Running IPNetRouter 1.5 or later 68K only (unlimited LAN clients) | $49 |

Upgrades to the latest version within the same major release are free to all previously registered users, except purchasers of the 68k only license introduced in 1.5. (e.g. v1.0.0 through v1.9.9). Purchasers of 1.5 or later 68K only licenses must currently buy a separate license for the PPC version.

## Register on-line to receive your key within minutes

Register on-line at the Sustainable Softworks WWW Registration Page to receive your key within minutes, or via Kagi

## Special student discount available

As part of our ongoing support for educational organizations, users having an email address ending with the .edu suffix ("edu" domain) or ".k12." organization identifier are only charged $62.30 (33% off the regular commercial registration fee) when registering via our WWW registration server.

Please contact us if you do not have such an educational suffixed email address yet have proof of affiliation to an educational institution.

Use of this Software ("IPNetRouter") is subject to the terms of the LICENSE AGREEMENT below.

This Software is licensed, not sold, to you by Sustainable Softworks for use only under the terms of this License. Sustainable Softworks retains ownership of the Software itself and reserves any rights not expressly granted to you. This Software including all screens and displays is Copyright 1996-2000 by Sustainable Softworks.

# LICENSE AGREEMENT

1. License.

   This license allows you to: (a) Use one copy of the Software on a single computer at a time for each licensed copy registered to you. (b) Make copies of the Software in machine readable form solely for backup purposes. (c) Use an unregistered demo copy during a single trial period of up to 21 days. (d) Redistribute the complete unregistered demo package including all documentation, copyright, and license information.

2. Restrictions.

   You may not reverse engineer, modify, or create derivative works based upon the Software or any part thereof.

3. Termination.

   This License is effective until terminated. You may terminate this License at any time by destroying the Software and related documentation and all copies thereof in your possession. This License will terminate immediately without notice from Sustainable Softworks if you fail to comply with any provision of this License. Upon termination you must destroy the Software and related documentation and all copies thereof in your possession.

4. Release of Liability.

   While reasonable effort has been made to ensure this Software operates substantially as described, Sustainable Softworks cannot guarantee proper operation in every possible configuration. For this reason, the Software is provided "AS IS". You assume full responsibility for evaluating the Software and determining its appropriateness for any use. In no event will Sustainable Softworks be liable to you for any damages arising out of the use of this software.

That's it. Enjoy!

Peter Sichel

Sustainable Softworks
13 Fieldside Drive
Cumberland, RI 02864

SUSTAINABLE Softworks

products

# IPNetRouter Feature and Price Comparison with other Solutions

• IPNetRouter

Overview

Descriptions and Specifications

Feature and Price Comparison

Readme

Download

Installation Guide

Getting Started

Release Notes

Troubleshooting

FAQ

Pricing/License


IPNetRouter is a great product, but how does it compare with other solutions? This page is designed to help you answer this question.

## Side-by-side Comparison of IP Routing Solutions

Last updated 16-Aug-2000

| Product/ Feature | **IPNetRouter** 1.5 | **Vicom SurfDoubler** 6.6.3 | **Vicom SoftRouter/ Internet Gateway** | **Hardware Routers** |
|---|---|---|---|---|
| Price | $89.00<br>$49 for 68k (1) | $29 or $39 | $99.00 for 5<br>$148.00 for 10<br>$249 for unlimited users | Lowest end starts at about $100 |
| Number of Users | Unlimited | 2 or 3 | | Typically unlimited, may cost more $ (2) |
| Free Updates | Yes (1) | No | No | Varies. Usually only free if the manufacture is at fault |

| | | | | |
|---|---|---|---|---|
| Open Transport Native (best performance) | Yes | No | No | Only expensive hw routers can route IP as well as a Mac! |
| Mac Hardware Supported | Mac 68030 through current Macs | Mac 68040 through PPC for gateway and client | Mac 68040 through PPC | May have problems with some Macs |
| RAM Useage | 2MB | 5MB | 6MB | (Requires browser application for configuration) |
| MacOS | 7.5.3 and later | 7.5.3 and later | 7.5.3 and later | (Varies. Some hw routers require a PC to do a firmware update.) |
| A Full IP Router | Yes | No | Yes | Usually more $ to get it right |
| DHCP Server | Yes | No | Costs extra | Usually more $ to get it right |
| IP Filtering | Yes | Not on a per IP interface basis | ? | Usually more $ to get it right |
| Rollover NAT Interface | No | No | Costs extra | Usually more $ |
| Dynamic DNS | Yes | No | No | No |
| PPPoE (PPP over Ethernet) | Yes, MSS clamp set to MTU-50 (more compatible) | Yes, but MTU only -40 | Yes (MTU only -40) | Usually more $ to get it right |
| Wireless Routing | Install Airport card, get wireless LAN router (3) | Yes, but limited to one sublan per port | Yes, but limited to one sublan per port | Starts at $299+ (Most don't fully support PPPoE routing as yet) |

| | | | | |
|---|---|---|---|---|
| Gateway dial-in server | No | No | Costs extra | Usually more $ |
| Single Ethernet port gateway with NAT | Yes | No | No | not usually necessary on hw routers |
| Faceless Background Version? | Yes | No | ? (Can't figure out from their website) | n/a |
| Download from the web in an emergency? | Yes | Yes | Yes | No |
| Other | Supports 1gigabit ethernet,Token Ring, MacIP, PPP, ISDN, IPSec/PPTP, ARP configuration, one-way telcos | Limited number of users and features, have to pay more to expand network | Only unlimited license ($249) gets you out from under the gun and even than you have to pay for updates | More $ to get 100mb ethernet and other interfaces (If your Mac has to be on, it might as well be the router!) |

Table Footnotes

1. We offer a 50% discount on licenses to educational institutions. An IPNetRouter $89 license is good for both 68k and PPC machines and the PPC version continues to receive frequent free updates; active development has ended for the 68k version as of August 2000. The price for the 68k only license was lowered accordingly. Sustainable Softworks has not charged for updates to IPNetRouter 1.x since its debut in 1997.

2. Many cheaper routers are limited to only one sublan of 254 users or even only 32 users on that single sublan. IPNetrouter is not limited in this manner. This limitation is caused by poor or slow performing hardware or NAT firmware.

3. Apple licensed IPNetRouter as part of its Airport software router package. IPNetRouter permits more sophisticated IP configuration of an Airport installed Macs network interfaces than Apple's Airport 1.2 release.

# What Users are Saying about IPNetRouter

## Other Reviews

Brindley Network Consulting
Internet Sharing Software, Macworld, January 2000, page 51
Apple Wizards

# Our Perspective

Common sense tells us that no single solution is best for everyone. We would like to take this opportunity to comment on some of the strengths and weaknesses of IPNetRouter.

First, IPNetRouter is fundamentally a utility for configuring TCP/IP that lets you access the features Apple left hidden. It tries to put a Macintosh GUI on UNIX like functionality. If you are familiar with TCP/IP networking, IPNetRouter is powerful and flexible. If you are new to TCP/IP networking, the lack of guidance can be intimidating. The instructions on our website are intended to help. We also have a web configuration application that can help you overcome the interface to a large degree.

IPNetRouter doesn't cost more per user like some Internet Sharing solutions. We don't charge for bug fix updates every 6 months.

IPNetRouter allows you to choose which features you want. You are not forced to use NAT (Network Address Translation) to use IP multihoming, IP filtering, DHCP, or configure IP routing. IPNetRouter works with Ethernet, Fast Ethernet, Gigabit Ethernet, PPPoE (PPP over Ethernet), Token Ring, LocalTalk, and other Data Link Providers. There's no need to replace your router hardware when switching among dial-up, ISDN, Cable Modem, xDSL, or AirPort Wireless.

As a small Mac focussed company, we know our product and try to be responsive. Our DHCP Server is tested extensively with the Mac. Our Internet tool suite IPNetMonitor works seamlessly through our NAT gateway. IPNetRouter is updated frequently and supports QuickTime Streaming.

Although many people assume hardware routers are faster, performance is not as obvious as it first appears. Compare a Cisco router at $1500 with IPNR on a contemporary Mac. The Cisco box has a 68030 at 30 MHz, and perhaps .5MB of RAM. Contrast this with a 300 MHz RISC processor, 100 MHz system bus, .5 MB backside Cache, 64 MB RAM, 6 Gig HD, Monitor, Keyboard, and Mouse. Open Transport (configured using IPNR) can route at or near wire speed between 100Mbps FAST Ethernets. We're not aware of any hardware routers that can do this for under $3,000. Low cost hardware solutions simply cannot keep up. Some can't even take full advantage of a DSL or Cable Modem connection. The key to OT performance is to respond to a network interrupt and pass pointers to a STREAMS message up the stack and back down again in less time than it takes to send a 1500 byte Ethernet packet.

Since routing occurs at interrupt time, it is not affected by other applications.

We also believe that by using your Macintosh for quick and efficient networking, rather than purchasing an unnecessary hardware router, you help reduce landfill and keep Macs in the limelight. And, its always nice to see the look on people's faces when you explain to them that your Mac is saving the planet by running rings around NT or Cisco routers and is way easier to configure than a linux machine!

# Some Tradeoffs

There's no retail packaging or printed manual (we're trying to be earth friendly). The manual can be downloaded in PDF format for easy printing.

IPNetRouter is usually a set it and forget it solution. It can be extremely stable on a well configured Mac (running for months without intervention), but we also know Macs can and do crash. We do offer a Faceless Background version which can prevent less knowledgeable users from changing IPNetRouter parameters and eliminate the clutter of having the router always available in the process menu.

IPNetRouter does not offer detailed security logging or remote management (SNMP). IPNetRouter is not a PPP dial-in server although another machine on your private LAN could be configured for such and used in conjuction with it. IPNetRouter does not offer web caching or content filtering. Other software and hardware can be used to supply this functionality if you think you'll need it.

We do charge a modest fee for IPNetRouter licenses. If you continue to use it after the 21 day free evaluation period is up, please purchase a key.

Back to the top

---

SUSTAINABLE Softworks

*products*


IPNR Config PPC v1.0.0

# Download

• **IPNetRouter**

Overview

Descriptions and Specifications

**Readme**

**Download**

**Installation Guide**

Getting Started with IPNetRouter

Release Notes
Troubleshooting

FAQ

**Pricing/License**

IPNR Config PPC Release v 1.0.0

IPNR Config 68K Release v 1.0.0

---

# IPNetRouter Configuration Application

Our IPNetRouter Configuration Application is designed to simplify IPNetRouter setup for basic Internet configurations. This includes PPP, cable modem, and DSL/ADSL Internet connections (with or without PPPoE).

After running the application, and following the directions provided, each client machine on your local area network (LAN) should be able to access the Internet throught the machine running IPNetRouter (gateway machine).

(This product is still undergoing development. We greatly appreciate your feedback regarding its use. Please address specific comments to ipnr_config_app@sustworks.com).

## Here is how it works:

1. You run the IPNetRouter Configuration Application and click the first button circled in Red. This transmits port information to our server and takes your browser to a web page customized for your configuration.

2. We ask you a few questions about your Internet connection. We also ask how your client machines are connected to the machine running IPNetRouter (your gateway machine). Depending upon your entries, we will present you with you a few more easy to answer options.

3. Our server builds a custom IPNetRouter configuration file developed specifically for your network. This file is automatically downloaded to your machine.

4. You check and/or modify your active TCP/IP control panel settings.

5. You launch IPNetRouter from the custom configured file which has been downloaded to your machine.

That's it. This entire process is automated and typically takes about 5 minutes to complete.

**What is required:**

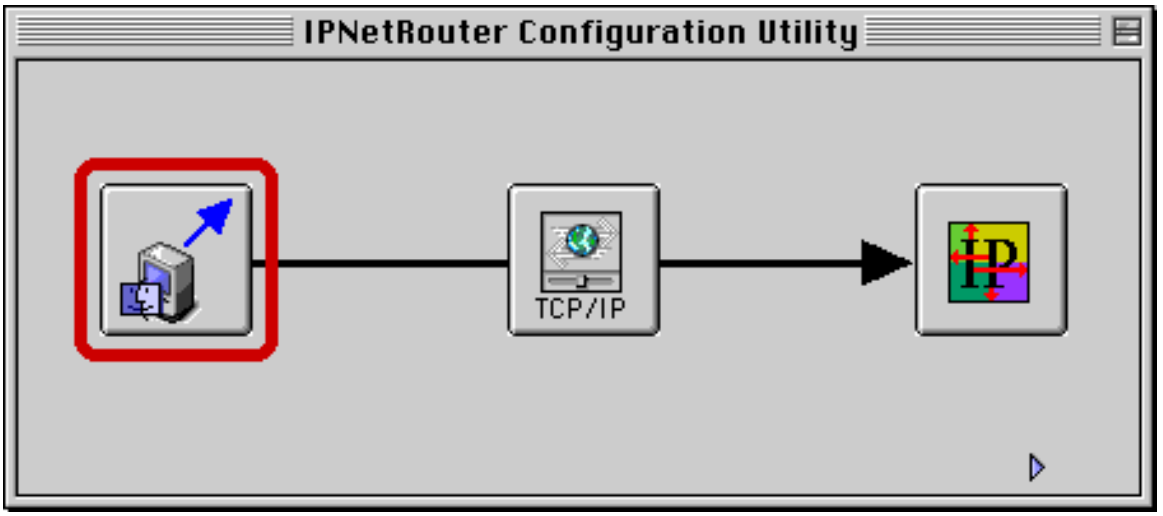
- IPNetRouter must be installed (but not running). This can be either the regular or Faceless Background (FBA) version*.
- An active Internet connection.
- A running browser (either Netscape Navigator/Communicator or Internet Explorer).

*IF you have installed the FBA version of IPNetRouter, launching the Configuration Application will automaticaly quit it. The FBA will restart after you click the IPNetRouter icon button in the Configuration Application. It will also load the "Router Config" settings file which is contained within in your Preferences Folder.

**What you need to do:**

Download and install the IPNetRouter Configuration Application on the machine on which you will be running IPNetRouter. We recommend that you install this application in the same folder as IPNetRouter (but you do not have to install it in the same folder).

Run the Configuration Application. You should see a screen similar to the following:




Click the button circled in red and follow the directions as presented in your browser. This should do it!

---

# Other features of Configuration Application:

**1. Modify your existing IPNetRouter configuration documents (advanced users).**

Launch the Configuration Application while holding down the "Option" key.

Assuming you have already created an IPNetRouter configuration document (and this document resides in either the same folder as the configuration application or in your Preferences folder), you can easily transmit this document to our server for additional modification by clicking the upload document button:

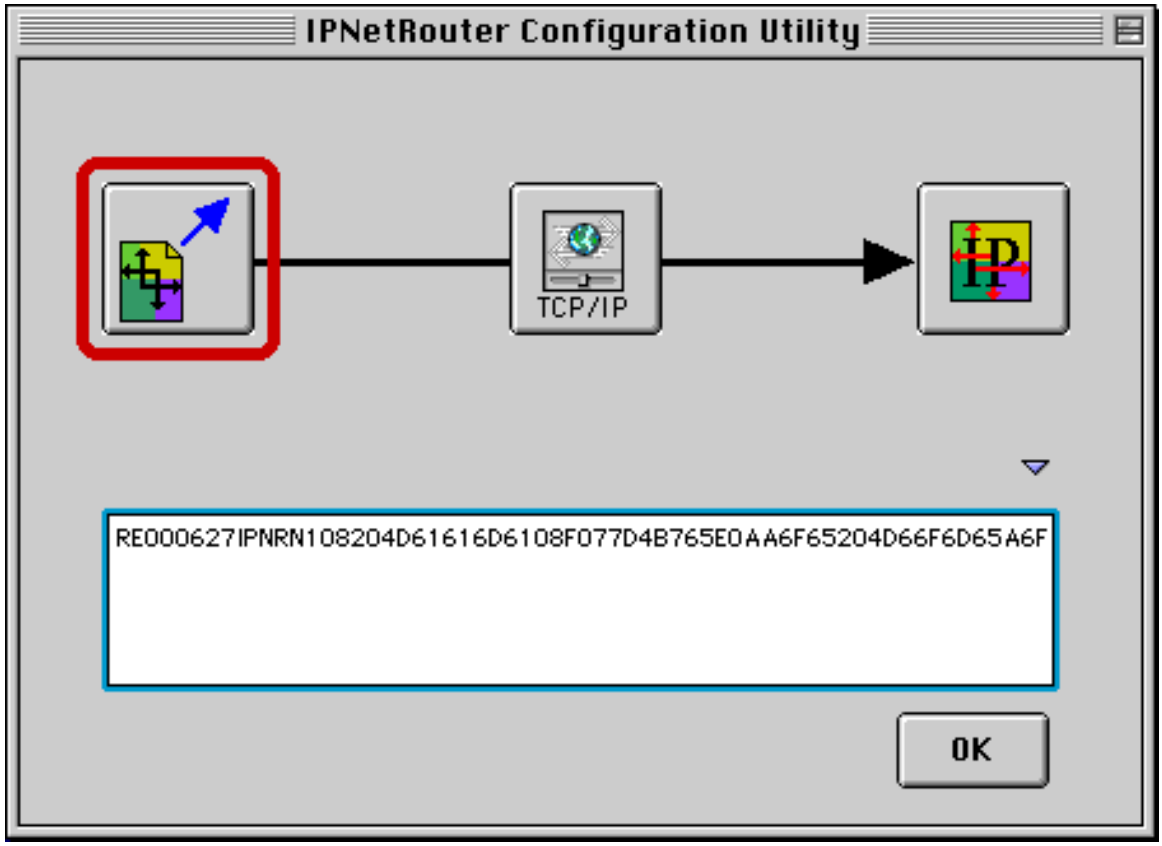
Just click on this button to upload your existing IPNetRouter configuration document. If you have several such documents, you will first be presented with the standard Open File dialog box to choose the file you wish to upload and modify.

Once the file is uploaded, you will be presented with a page which allows you to manually add other interfaces, functions, port maps, and filters. You can then automatically download the modified file to your machine.

**2. Enter your IPNetRouter Registration.**

First you need to purchase a registration code from our online registration server. Once this is done, you can register either the normal IPNetRouter application or the Faceless Background version through the Configuration application.

Just expose the edit box using the small arrow button. Copy and paste your IPNetRouter hash key into this edit box. Click the OK button. That will do it.




(hash key shown above is only for illustration purposes and will not actually register the application)

---

Note: The IPNetRouter Configuration Utility requires incoming TCP connections on Port 4670. You may have to inform your Network Administrator of this fact if you are attempting to use this utility from behind another firewall or NAT router. For most users, this will not be an issue.

---