

**making
the
software
work**

- Understanding your modem
- Hayes commands and S-Registers
- Installing Internet software
- AmiTCP step-by-step installation
- PPP step-by-step installation
- Dialscripts
- Testing your connection
- Your first file transfer
- Configuring email and Usenet news

2

Understanding your modem

Before you install any Internet applications or TCP/IP stacks you need to get the basics working. This means getting your Amiga talking to your modem and your modem happily making phone calls.

There should be a number of status LEDs on the front of your modem. These are handy when trying to get your Amiga and modem working properly. You can tell exactly what state the modem is in by knowing what combinations of LEDs mean what. Here are a few of the most important:

AA (Auto Answer) When the AA light is on your modem will attempt to answer any incoming calls. You can set how many rings the modem will wait before picking up the line.

SD (Send Data) The SD light is active when the modem is sending data down the phone line, and/or it is receiving this or other data from your Amiga.

RD (Receive Data) The SD light is active when the modem is receiving data from the phone line, and/or it is sending this or other data to your Amiga.

TR (Terminal Ready) The TR light should come on as soon as the modem is ready to accept data from a terminal (your Amiga, that is). Depending on how you have configured your modem, this LED may illuminate only when you start a terminal program which “opens” the Amiga serial port.

OH (Off Hook) The OH light comes on when the modem metaphorically takes the telephone handset off the hook—when the modem picks up the line in order to dial out. It should be on for the duration of a call. If the light suddenly goes out, the modem has hung up the line for some reason.

CD (Carrier Detect) The CD light comes on when a connection to another modem has been established and a “carrier signal” is available over which data can be sent and received.

Depending on the model of your modem there may be many more LEDs, or even an LED or LCD display showing the current status of the modem. There is usually some way of telling what speed the modem is set to, or whether it is in fax mode. During a call you should also be able to see what error correction and data compression (if any) protocols are in use by the two modems making the call. Refer to your manual for more details on these.

Making a call

To check if your modem is working correctly you should start a terminal program—NComm and Term are two popular shareware packages—and get your modem to make a telephone call.

When you have loaded your terminal program you should try typing in the command AT which tells the modem to stand to ATtention! If you are in luck the modem should respond with “OK”, if not skip down to the next section for some advice.

It’s always a good idea to reset the modem before you start dialling. To do this type ATZ. The modem should once again respond “OK”. Next you can try getting the modem to pick up the phone line—use ATH1. The OH light

should come on and you should be able to hear a dial tone. To hang up type ATH0 (that's a zero at the end). If you were able to hear a dial tone then all is well and you can try dialling a number. If not, check the phone line is plugged in correctly, or type ATM1 followed by ATL2 to make sure the modem speaker is turned on.

Note that you don't have to pick up the phone line with the ATH1 command before dialling. The modem will pick up and put down the phone line automatically when it attempts to dial. You should use the ATH commands only to check if a dial tone is present.

To dial a number you can use either ATDT or ATDP. The D stands for dial, the T and P for tone and pulse. Most new telephone exchanges now use tone dialling but if you are connected to an older exchange you may have to use pulse dialling. Try them both to see which works. Tone dialling is preferable.

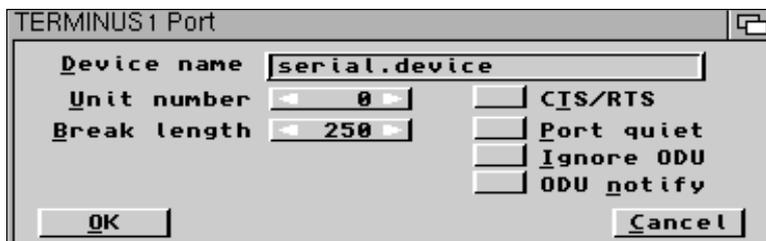
So try a number—ATDT123456 will dial 123456. Ideally you should try ringing your ISP or a local Bulletin Board System but any number which has a modem at the other end of the line is fine. Bear in mind that if you ring a number that doesn't have a computer and modem attached at the other end you'll hear a muffled "Hello" coming from your modem and the person who has answered your call will get an earful of modem tones! If you don't know any numbers to try out, call yourself to check that the modem dials and test for a busy signal. Or try calling the speaking clock. Really! At least you'll know the modem is making calls correctly.

What if it just doesn't work?

Okay, so you've switched on the modem, loaded your terminal program, but nothing is happening...

If you are typing on your Amiga and nothing is coming out in the terminal window, take a look at the modem TR, SD and RD lights. Is the TR light on? If not, the Amiga serial port is not talking to the modem. If the light is on, check if the SD and RD lights are flickering as you type. If they are, your commands are getting through to the modem and the modem is responding, so your terminal program probably just needs tweaking a little bit. If the TR light is on and SD and RD don't respond when you type, there is a more serious problem. Perhaps a hardware problem such as a faulty cable.

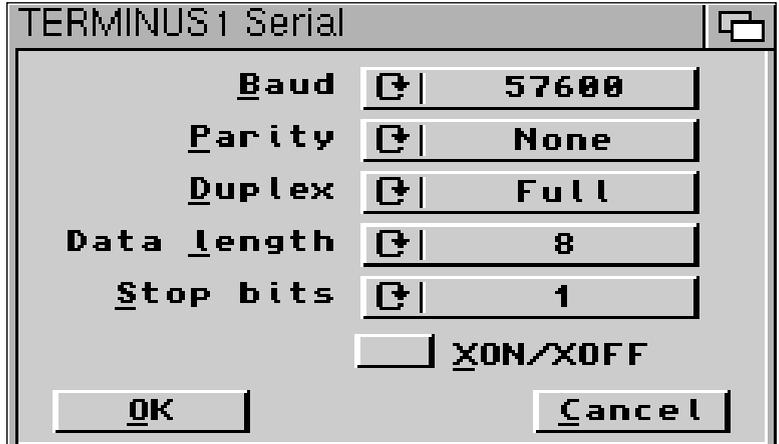
Modems are very fickle. Don't jump to any conclusions about them not working. They are just extremely easy to upset. You should check all the settings in your terminal program. The most important setting, and it is probably set correctly by default, is that your terminal program is talking to the Amiga "serial.device" unit 0 (zero).



After this, check the baud rate or speed at which your Amiga is talking to your modem. We discussed the best speeds to choose earlier in the book.

For testing purposes you can try 2400. Lastly , make sure your serial port settings are N-8-1.

N-8-1 is short for No parity, 8 data bits, 1 stop bit. Basically what this means is that for every eight data bits of data that the modem sends down the line, a single stop bit should be added to the end of the eight data bits, and no parity bit should be added when doing error checks. By convention a start bit is also always added, so for every eight bits of data you send a start bit is inserted before your byte and a stop bit is appended afterwards.



These extra bits are used for clocking information. The two modems don't know whether your eight bits are going to be ones or zeros, but if they can rely on start and stop bits always being present then over long periods of time they will never lose synchronisation with each other. If the two modems lost track of which bits were which and got confused about what was the start of a particular data byte, you would just get complete garbage coming out at both ends.

Hayes commands and S-Registers

The AT commands with which you controlled your modem in the last section are part of the Hayes Command Set. Hayes were one of the first companies to produce modems and the language they used for their modems became a de-facto standard in the modem industry. There are a number of basic Hayes commands which are common to almost all modems, but such is the complexity of today’s models that every vendor adds their own set of unique commands to control the enhanced features of their particular modem.

As well as the AT commands your modem has a number of Status or “S” registers which store the current configuration. The contents of these S-Registers can be controlled with an AT command. To check the current settings on your modem try AT&V (or check with your manual for the correct command). You should see something like:

```

TERMINUS1
at
OK
at&v
ACTIVE PROFILE:
B1 E1 L1 M1 N1 Q0 T V1 W0 X4 Y0 %C3 %E0 %G1 -K0 &C1 &D0 &K3 &Q5 &R1 &S0 &X0 &Y0
S00:000 S01:000 S02:043 S03:013 S04:010 S05:008 S06:004 S07:050 S08:004 S09:012
S10:014 S11:050 S12:050 S18:000 S23:190 S25:005 S26:001 S36:007 S37:000 S38:020
S40:104 S41:131 S44:131 S46:138 S48:007 S91:010 S95:000 S109:062 S110:014

STORED PROFILE 0:
B1 E1 L1 M1 N1 Q0 T V1 W0 X4 Y0 %C3 %E0 %G1 -K0 &C1 &D0 &K3 &Q5 &R1 &S0 &X0
S00:000 S02:043 S06:004 S07:050 S08:004 S09:012 S10:014 S11:050 S12:050 S18:000
S23:190 S25:005 S26:001 S36:007 S37:000 S40:104 S41:131 S46:138 S95:000

STORED PROFILE 1:
B0 E0 L2 M1 N0 Q0 T V0 W0 X0 Y0 %C2 %E0 %G0 -K3 &C1 &D2 &K0 &Q4 &R1 &S1 &X0
S00:043 S02:004 S06:050 S07:004 S08:012 S09:014 S10:095 S11:050 S12:000 S18:052
S23:000 S25:001 S26:073 S36:000 S37:020 S40:131 S41:138 S46:007 S95:062

TELEPHONE NUMBERS:
0=
1=
2=
3=
OK
TERMINUS1 V1102 LLC 2 FX 8N1 57600 00:00:02 16:02:22
    
```

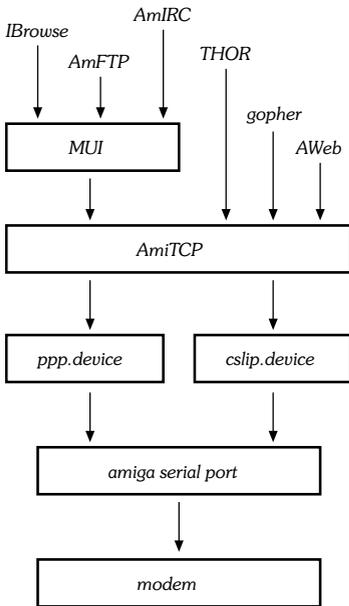
At the top of the screen you can see the active settings for this modem. Further down you can see that the modem has two memory banks where it can store alternative settings. It can even store frequently called phone numbers. Let’s go through some of the most important AT commands and S-Registers. The commands shown to the right have been tested only on Supra modems but most modern modems should use the same commands. Again, check with your manual for any small differences.

The bottom AT command in that table shows how to set the value in an S-Register. An important S-Register is S0, which determines whether or not the modem will answer incoming calls. Or, to be more precise, how many rings to wait before answering. So to make the modem answer incoming calls after two rings you would type AT&S0=2. To prevent the modem answering tell it to answer after zero rings—AT&S0=0.

In general, the factory defaults for your modem will probably be adequate for your needs. You may want to adjust one or two settings, but you shouldn’t have to spend time learning what each of the Hayes commands does. As long as you know how to reset the modem and make a call, you’re all set. If you do make any adjustments to the defaults, make sure to store your settings by issuing the AT&W command. Your configuration will then be stored in the modem’s NVRAM (permanent memory) and will take effect every time you turn on your modem from cold.

- AT&F Reset to factory defaults (on Supra modems use AT&F2)
- ATZ Reset to last saved settings
- AT&W Store current settings to non-volatile ram (NVRAM)
- AT&V Show current modem settings
- ATM0 Turn off speaker
- ATM1 Turn on speaker
- ATLx Speaker volume control (x=0-3)
- ATDx Dial phone number x
- ATA Answer incoming call
- ATH1 Pick up phone line
- ATH0 Hang up phone line
- AT&D0 Ignore DTR signal (modem ignores status of Amiga terminal)
- AT&D2 Follow DTR signal (hang up on exiting terminal program or reboot)
- AT&Sx=y Set S-Register x to value y

Internet software stack



Installing Internet software

Modem plugged in and working, now's the time to configure and test your Internet connection. We'll go through the different questions AmiTCP will ask you during installation and explain what they mean. Having successfully installed and tested the software, we'll look at how you can now transfer new software to your Amiga over the Internet. And we'll briefly mention some other utilities you might like to install which will make life easier when configuring email and news programs later on.

Installing AmiTCP

AmiTCP comes in two archives: the user binaries and documentation, and the software development kit (SDK). The documentation is quite extensive and a full description of the AmiTCP programming interface is given in the developer kit. AmiTCP can be found on Aminet in the `comm/tcp` directory. The user files are in "AmiTCPdemo_40.lha" and the developer files in "ATCP_sdk_40.lha".

The user archive has everything you need to install AmiTCP. A standard Commodore Installer script is provided and after several releases of the software it is now pretty good. It can be quite daunting though, especially if you're not certain what you want—default domains, gateways and nameservers may not mean too much to the new Internet user. You'll probably find yourself installing the package several times over as you figure out exactly what you're doing.

The installer creates all the necessary configuration files for AmiTCP and once you understand the structure of these files you won't need to go near the installer again.

Applications included with AmiTCP are the basic Internet utilities such as Telnet, FTP and Finger. Numerous other tools such as netstat, route, ping, and arp are included for configuring and troubleshooting your IP link. We'll take a look at how to use some of these later on in this chapter.

As well as these standard networking utilities from the Unix world there is an Amiga specific application called NetFS. This allows you to connect two Amigas and share filesystems across that network using AmiTCP.

Configuring AmiTCP can be tricky to get right first time. Most of the defaults given by the Commodore Installer will work just fine, but answer one crucial question incorrectly and you could be tearing your hair out in frustration for hours. Here are the most important things you need to have figured out before installing AmiTCP.

What protocol are you using?

Is it SLIP, CSLIP or PPP? Your ISP will tell you what choices you have. SLIP and CSLIP come bundled with AmiTCP. PPP is available separately and is often the first choice an ISP will give you. I recommend you install and use PPP with AmiTCP. PPP comes with its own built-in dialler so it will automatically call your ISP.

Your IP address

Every computer on the Internet has an IP address. You will either get a static IP address, which is a fixed number assigned to your Amiga for good, or your ISP will use dynamic addressing where a new IP address will be transmitted down the phone line each time you connect to the Internet.

Dynamic addressing is much more convenient for your ISP. It is also easily handled by PPP. If you are using dynamic addressing, enter 0.0.0.0 as your IP address.

Netmask

Check with your ISP but it is most likely that your netmask will be 255.255.255.0.

Broadcast address

Again, check with your ISP. If you are using static addressing and have an IP address in the form of 1.2.3.4, then your broadcast address will be 1.2.3.255.

Gateway

The gateway or default route is the address of the machine in your ISP's premises which routes packets to the Internet. When your Amiga has a packet destined for a remote site it needs to send it somewhere, so it sends it to the default route or gateway.

Nameserver

The nameserver is the machine that performs the translations from Internet addresses like *www.broadcom.ie* to the IP numeric equivalent of 192.107.110.20. If your nameserver is incorrectly set you will only be able to use IP addresses in numeric form and will not be able to use the friendlier domain name system (DNS).

Domain name

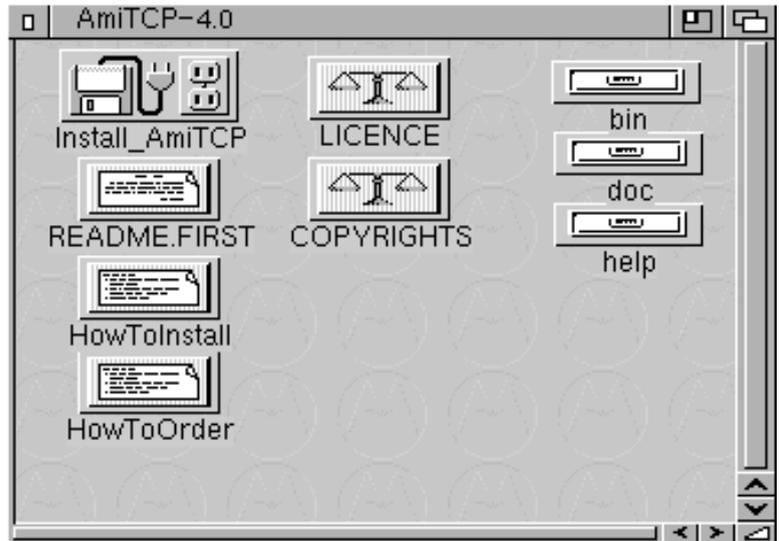
This is usually the same as that of your ISP, so it will be something like *isp.co.uk*. If perhaps you are starting your own company and have arranged with your ISP to get a static IP address and your own domain name, then it might be something like *mycompany.co.uk*.

When you have successfully configured AmiTCP you'll need a way of automating the login process to your ISP. If you are using PPP you can use the built-in PPP dialler program, for which you can write a "chatscript" that tells your Amiga how to dial your ISP's number and enter your username and password. If you don't have PPP then you can either login manually with a terminal program, quit the terminal program and then start AmiTCP, or you can get a separate dialler program whose sole purpose is to log you in to your ISP. (It lets AmiTCP take over from there.) Note that if you are using a terminal program to login to your ISP you should enter the command AT&D0 to your modem before you dial. This tells your modem not to hang up when you quit the terminal program before starting AmiTCP.

Lastly, to start up AmiTCP and get online type STARTNET. When you're all done, and have shut down your applications, you type STOPNET. Both commands are actually just script files in the amitcp:bin drawer and can be tailored to perform custom operations at startup or shutdown.

AmiTCP Step-by-Step Installation Guide

The TCP/IP stack is the most complicated piece of software required for Internet access. Unless you have a strong Unix or communications background you'll probably find the installation process somewhat bewildering, so let's go through the AmiTCP installation process step-by-step.



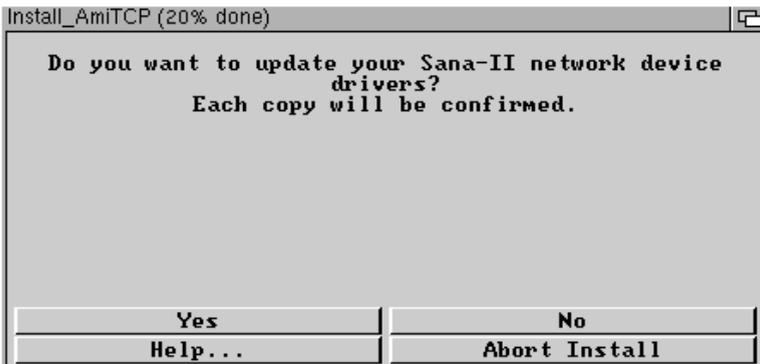
After you have unpacked the AmiTCP demo archive you will find an AmiTCP-4.0 drawer on your system.

AmiTCP comes with a standard Commodore Installer script which helps automate the installation process. Double-click the Install_AmiTCP icon when you're ready to install.



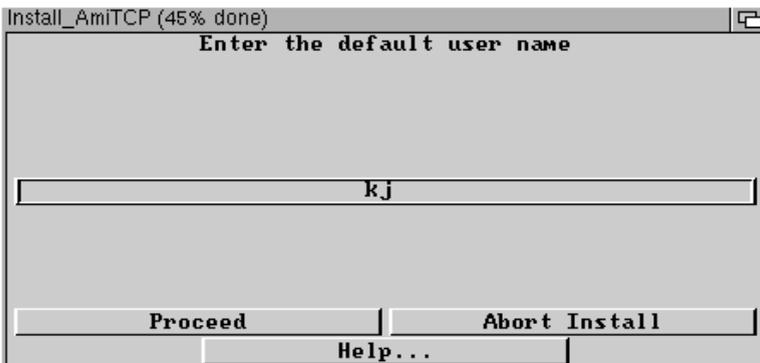
Your first choice, as with all Installer scripts, will be to set the installation mode to Intermediate or Expert. If you want to double-check everything in the installation process then you can choose Expert mode, otherwise you should leave the default Intermediate mode active and click Proceed.

Next you will be asked if you want to install for real or whether you just want to have a trial run by pretending to install. Again, leave the default option active and install for real. You will now have to choose where to install AmiTCP on your hard disk. Pick a directory on your system where you plan on keeping all your Internet related programs.

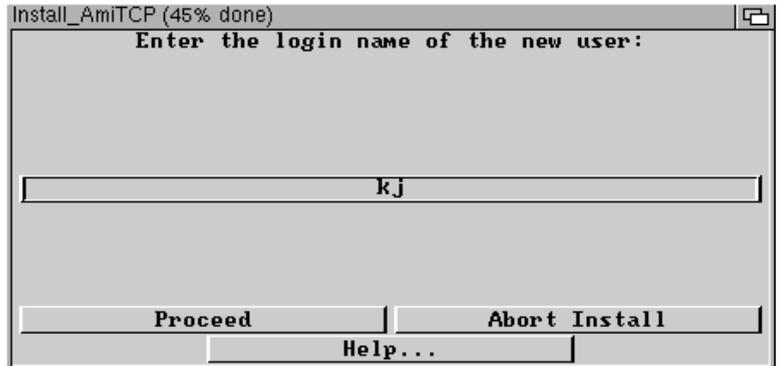


AmiTCP talks to network devices such as modems via Sana-II device drivers. You will be asked if you want to update any devices that you may already have installed on your system. Choose Yes and then check that each device AmiTCP attempts to install is newer than any existing version already installed on your system.

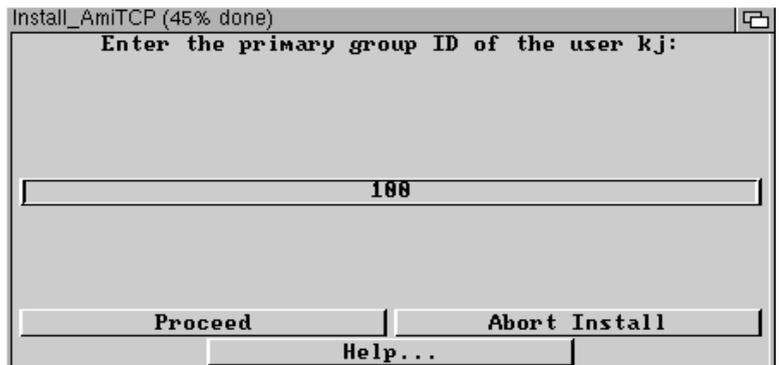
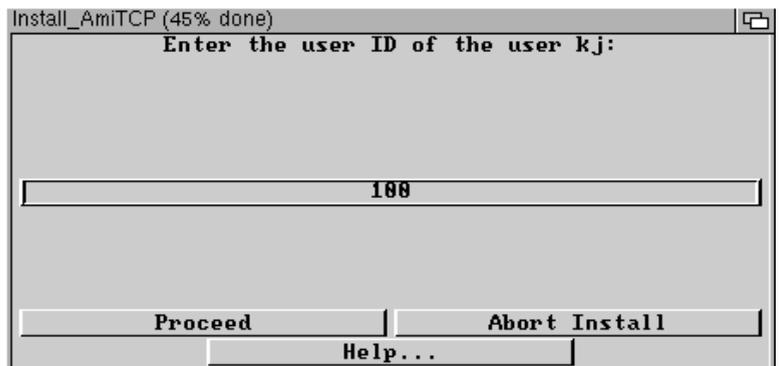
Napsaterm is a small terminal program that comes bundled with AmiTCP. You are given the option to install the fonts that come with Napsaterm. Unless you really have a good reason to do otherwise, you should go ahead and install the fonts.



If you have previously installed AmiTCP you will be asked if you want to use settings from that installation. But since this is likely to be your first time installing AmiTCP you can just ignore this and click No. AmiTCP will next ask for a default user name. This can be anything you like, but if you want to avoid confusion down the road it makes a lot of sense to use the same login name for AmiTCP that your ISP has given you on their system.

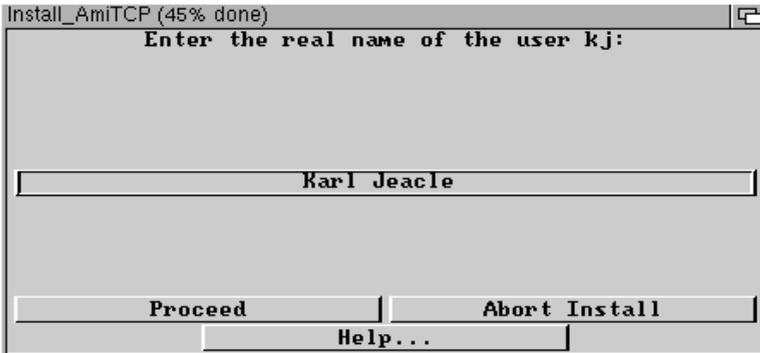


What you just entered was the default user which AmiTCP will login when it starts up. You now have to actually add that user to your system, so just enter your login name again.

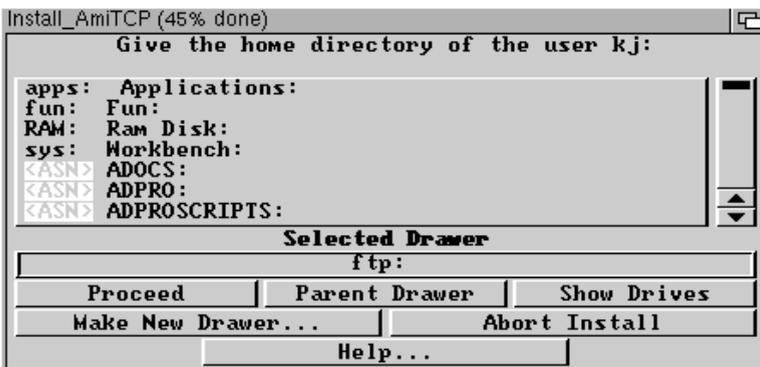


Next you have to enter a User ID (UID) and Group ID (GID). This has its roots in Unix and other multi-user systems. Each user on a networked computer has to have an ID number associated with him or her. Computers like numbers and so a UID is a lot more useful to it than an English word. The GID is supposed to represent the group you belong to. In the real world this might be a number representing departments such as sales, marketing, accounts, engineering and so on. In our simple Amiga world you can just choose the default UID and GID of 100.

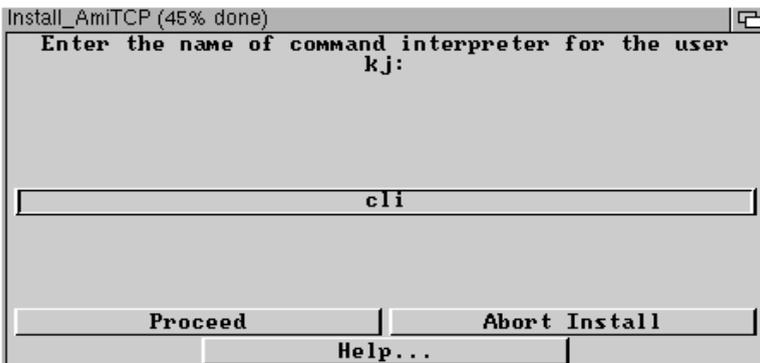
AmiTCP supports UIDs and GIDs so that it can allow you to have multiple user set-ups on your system. You could for example allow friends on the Internet to remotely login and access your Amiga.



After login names, UIDs and GIDs you have to enter your real name.



You will need to choose a home directory on your system. This should be a drawer somewhere on your hard disk where you might keep local configuration files for Amiga Internet programs—if you ever connect to your Amiga remotely this will be the default directory you will find yourself in.

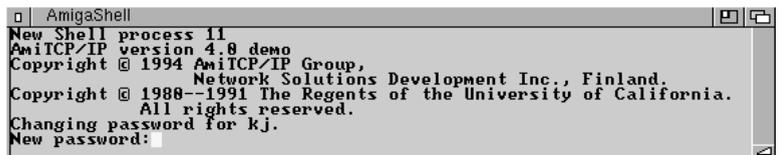


If you configure your Amiga to allow yourself to login across the Internet from another computer, you have to have set a command interpreter which will be run for you. The standard Amiga shell can be invoked with the CLI command, so you should use this as the AmiTCP user command interpreter.



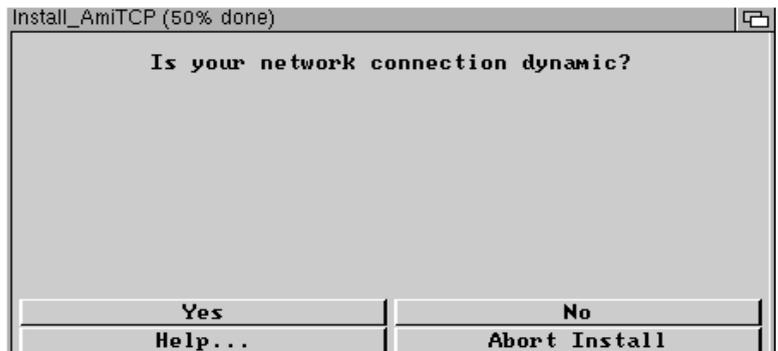
When you have entered all the relevant details for your user account set-up, the installation script will ask you to confirm that everything has been entered correctly. Click Add if all is well.

At this point AmiTCP will create the user account. To do this it needs a password, so a new shell window will appear with the AmiTCP login program running, prompting you for a password.



Try to use numbers, symbols or upper case characters in your password. You should make it as difficult to guess as possible. Don't make it too obscure, however, as you may have trouble remembering it yourself!

Don't imagine that the password is not important because you are the only person who uses your Amiga. When you are connected to the Internet, if you allow incoming Telnet or FTP sessions your machine could be vulnerable to attack while you are online. Don't be paranoid about this, it's highly unlikely that your machine's security will be compromised, especially when you are a dialup user with a dynamically assigned IP address (more on this later), but just be careful, and pick a sensible password.



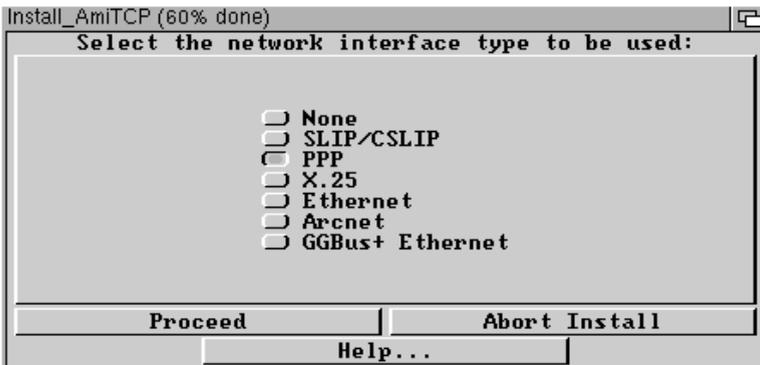
Having fully configured all the login user details, AmiTCP will attempt to configure any network interfaces you have attached. The first thing it needs to know is whether your network connection is static or dynamic.

A static connection means that your Amiga will have a single IP address uniquely assigned to it. These days, for a number of reasons, it's quite unlikely that you will be using a static IP address unless you are connecting to a machine at work or have a particularly friendly ISP. In most cases you will have a dynamic network connection. This means that each time you call your ISP it will allocate you an IP address for your Amiga that will be yours for the duration of that call only. When you hang-up, the IP address will be reused for someone else logging on.

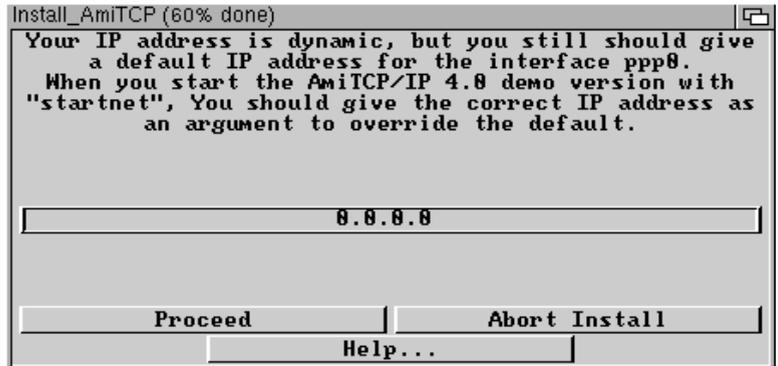
So unless your ISP has given you a specific IP address to use you should click Yes—your network connection is dynamic.



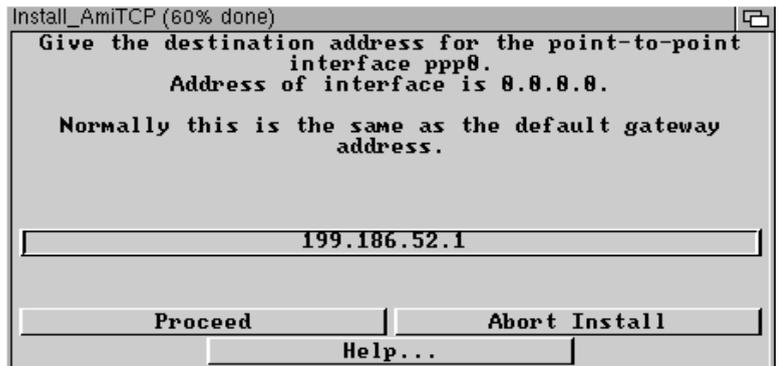
Next you'll have to enter your domain name. This will be something like *isp.co.uk* or if you are connecting to a university account, *blah.ac.uk*. If you know what your email address will be, your domain name is probably everything after the @ in your email address. For example, if your email address is *jblogs@isp.co.uk* then your domain name is *isp.co.uk*.



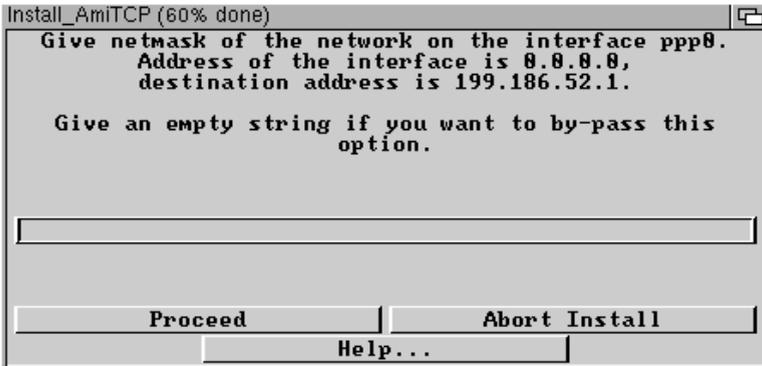
TCP/IP can run over a variety of different network types. It is possible to install Ethernet or Arcnet cards in a number of Amigas and network them together locally. However, if you are simply using a modem to connect to the Internet then you need a piece of software under AmiTCP to talk to your modem. This is usually SLIP, CSLIP or PPP. The most popular these days is PPP, using the aforementioned dynamic addressing. Click on whichever protocol your ISP has told you to use.



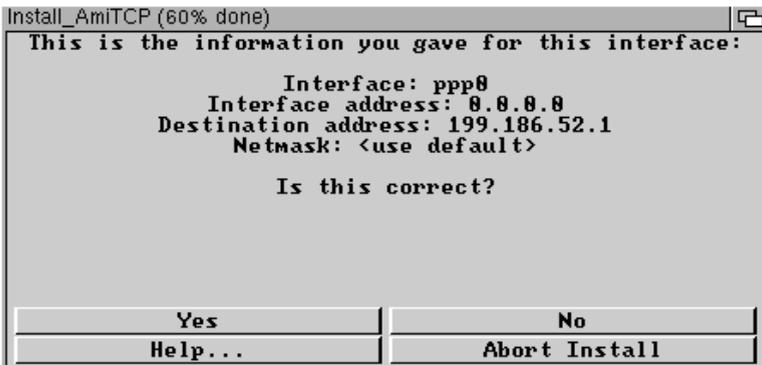
In the likely event that you choose PPP as your network interface, AmiTCP will ask you to confirm that this is indeed that case before going on to ask you your IP address. When using dynamic addressing you can enter the address 0.0.0.0 for the ppp0 interface.



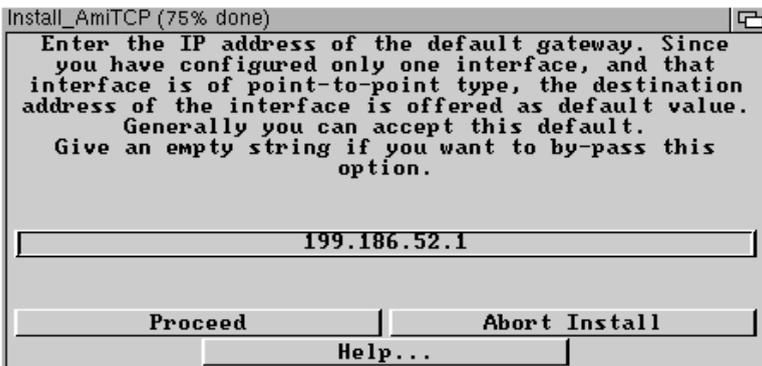
The computer at the other end of the modem will give you your IP address so you can get away without knowing your own address, but in order to communicate with the remote machine it is essential that you have its IP address. Only your ISP can tell you this. As AmiTCP notes, this address is usually the same address as your ISP's default gateway, so use this address if it's the only remote IP address that your ISP has given you. If you are really stuck here and your ISP has told you nothing, try using 0.0.0.0 again—it's possible that the PPP protocol will figure out the relevant IP addresses when it starts up.



With every IP address you must supply a netmask. It's usually quite easy for AmiTCP to figure out this address, so you can leave this option blank. However, if for some reason you have to enter a netmask then check with your ISP, though 255.255.255.0 is a good bet.



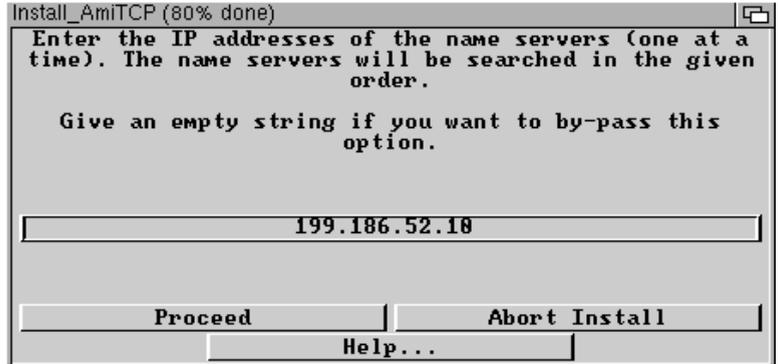
AmiTCP will now present a summary of your answers for this interface. Click Yes if you have entered everything correctly.



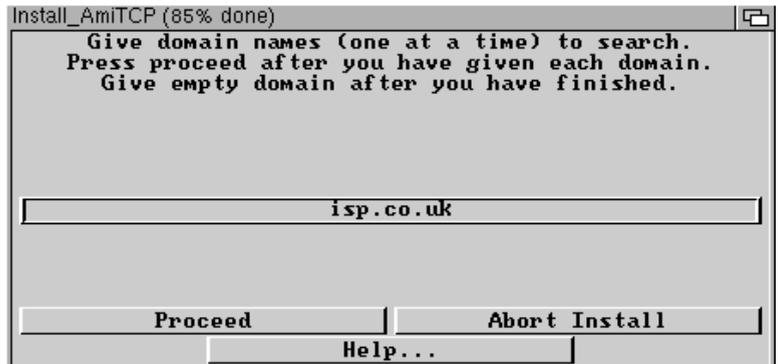
The default gateway is the machine to which your Amiga will forward all Internet destined packets. Because you have a point-to-point link to your ISP, all of your IP packets must go to the machine at the other end of your modem link, so usually this remote machine is the default gateway.

However, it is possible that the default gateway will be a machine somewhere else in your ISP's premises. In this case your ISP will provide you with the two separate IP addresses you require: the remote PPP address, and the default gateway address.

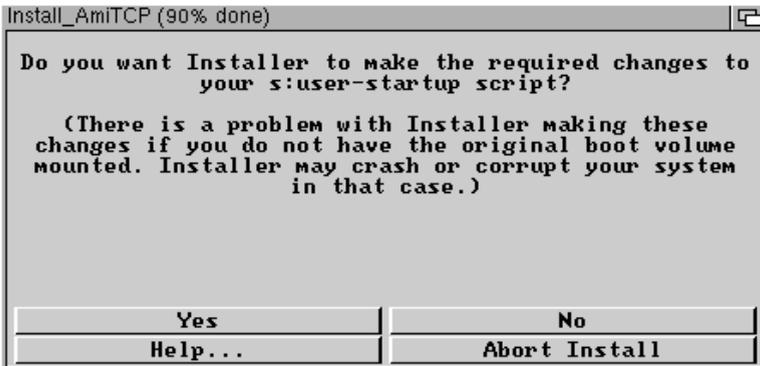
If they have only given you one address you can assume that the machine at the other end of the line is also your default gateway.



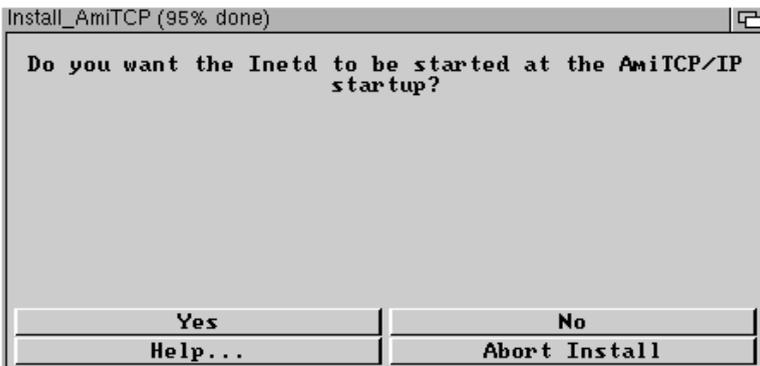
The one other IP address that your ISP should have provided you with is that of its name server or DNS machine. Name servers translate normal English-like hostnames into their numeric IP equivalents. AmiTCP can still work without a DNS server but you won't be able to make a lot of sense of anything without a DNS entry in your AmiTCP configuration files.



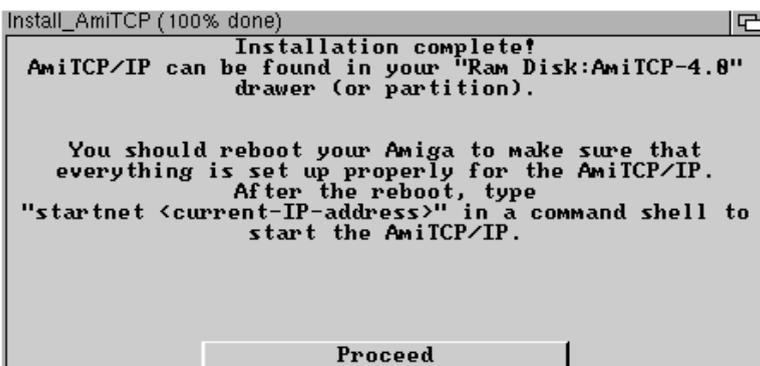
You will be offered the chance to enter more than one DNS server address. If you have more than one then by all means enter it here. If you are connected to multiple domains it is possible that you might have to enter more than one DNS server, and you will also have to enter each of the domain names that these DNS servers correspond to. This is not likely to be the case with the average home Internet connection so you can simply enter your single domain name and then leave the next one blank and click Proceed.



AmiTCP needs some AmigaDOS Assign commands and path additions to be made before it can run so the installation program asks you if you want it to modify your startup scripts to do this automatically. Click Yes.



The Inetd program is a daemon program which runs in the background waiting for incoming connections to your machine. If you want to set up FTP servers or Web servers or other advanced Internet services on your Amiga you may as well have the Inetd program started every time you launch AmiTCP. However, as mentioned earlier, this is where the security risk can lie. I would recommend choosing No here. You can add Inetd to your startup scripts at a later date if you like.



No more questions! AmiTCP is now installed on your system. As the requester says, you should reboot your Amiga for all the changes to take effect.

PPP Step-by-Step Installation Guide

AmiTCP requires a Sana-II driver to be installed underneath it to talk to a network device. Your network hardware is your modem and your Sana-II driver can be something called SLIP, CSLIP or PPP.

SLIP means Serial Line Internet Protocol. It was one of the first ways of getting IP to run over a serial line, the same kind of line your modem uses. CSLIP (Compressed SLIP) came next and improved on the speed of SLIP by compressing some of the IP packet headers being sent up and down the line. Both SLIP and CSLIP are commonly used today.

The most recent protocol to be used to get IP running over serial lines is the Point-to-Point Protocol. PPP is smarter than SLIP and CSLIP in a number of ways, but really all you need to worry about is that PPP should be installed along with AmiTCP in order to get your Amiga talking to your modem and hence your ISP.

Holger Kruse has written a shareware version of PPP for the Amiga. It is available on Aminet as “comm/net/PPP1_45.lha”. When you unpack the archive you will find the following files...

```

Output Window
Lha Evaluation V1.38 - Copyright (c) 1991,92 Stefan Boberg.
All rights reserved. Not for commercial use.

Listing of archive 'PPP1_45.lha':
-----
Original  Packed Ratio   Date      Time      Name
-----
11416    6971 38.9%  12-Feb-96 11:48:56 PPP/bin/PPPInfo
9684    6847 37.5%  12-Feb-96 11:48:58 PPP/bin/PPPLog
13248   8414 36.4%  12-Feb-96 11:41:84 PPP/bin/PPPLQR
18892   6658 38.8%  12-Feb-96 11:48:58 PPP/bin/PPSstats
654     349 46.6%  15-Jan-96 08:49:88 PPP/COPYRIGHT
41216  22799 44.6%  09-Mar-96 09:32:36 PPP/devs/ppp.device.000.eval
39324  22325 43.2%  09-Mar-96 09:32:38 PPP/devs/ppp.device.020.eval
36      36  8.8%   15-Jan-96 08:45:18 PPP/env/SANA2/ppp0.config
925     651 29.6%  09-Mar-96 09:35:28 PPP/FILES
97443  31658 67.5%  09-Mar-96 09:32:18 PPP/PPP.guide
1885    738 59.5%  15-Jan-96 08:48:38 PPP/REGISTRATION
-----
226635  106638 52.9%  24-Mar-96 02:06:00 11 files

Operation successful.

```

The files in the bin directory are just used for getting information on the PPP link while it is running. Don't worry about these. The important files are in devs and env. There are two copies of the “ppp.device” file which AmiTCP will try to open when it starts up—one for all Amigas, one for Amigas with at least a 68020 processor. You need only one of these files. The “ppp0.config” file is where your PPP settings will be stored. Lastly, the “PPP.guide” file is a comprehensive online AmigaGuide help file which explains in detail how to install PPP. Let's go through the steps involved in installing PPP once the AmiTCP demo is on your system.

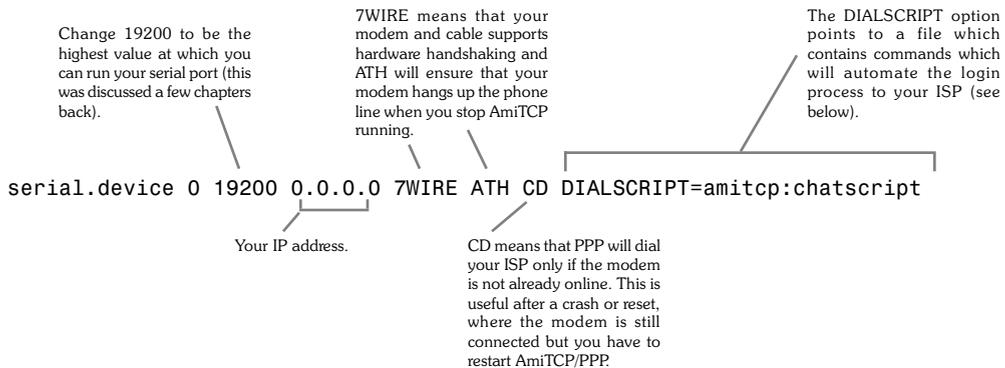
PPP Installation for AmiTCP 4.0 demo

First, make sure that you followed all the instructions given in the previous section when installing AmiTCP—that is, you should have chosen PPP as your network interface and entered 0.0.0.0 as your IP address if using a dynamic connection.

To install PPP itself you must choose one of the files “ppp.device.000.eval” or “ppp.device.020.eval” and copy it to your devs:networks directory, renaming it to just “ppp.device”. If you have an A1200 or A4000 you can use the 020 version.

If you want to use the PPP status programs copy the files in the PPP bin directory into the AmiTCP:bin directory. This is where all the AmiTCP binaries are held so it's a pretty good choice for where to keep the PPP files.

Next create a configuration file for PPP. It should be a text file containing a single line. The Configuration section of "PPPguide" goes through all the options possible, but a good start is...



Save this single line file to both "envarc:sana2/ppp0.config" and "env:sana2/ppp0.config".

Now you must modify the AmiTCP startnet and stopnet scripts to use PPP.

For AmiTCP:bin/startnet you should:

- Delete the first few lines that start with `.key`, `.bra`, `.ket`, `.def`.
- Add to the beginning of the file the line:
`online devs:networks/ppp.device 0`
- Change the line starting with `AmiTCP:bin/ifconfig ppp0` to:
`AmiTCP:bin/ifconfig ppp0 $ppp0iplocal $ppp0ipremote`
- Change the line starting with `AmiTCP:bin/route add` to:
`AmiTCP:bin/route add $ppp0iplocal localhost`
- After that line, add the line:
`AmiTCP:bin/route add default $ppp0ipremote`

For AmiTCP:bin/stopnet you should:

- Add the following line to the end of the file:
`offline ppp.device 0`

And lastly...

- You must insert this line somewhere in the "AmiTCP:db/interfaces" file:
`ppp0 DEV=DEVS:networks/ppp.device UNIT=0 IPTYPE=33 NOARP P2P`

Dialscripts

If you want PPP to automatically make a phone call to your ISP every time you run the AmiTCP startnet command, make sure you have the `DIALSCRIPT=amitcp:chatscript` option in your `ppp0.config` file as shown in the previous section. PPP has a built-in dialler which can be configured to automatically make a phone call, type in your username, your password, wait for your ISP to respond and then start PPP itself. To use this feature you should carefully read the Dialing section in the PPP manual, but to get you going let's create a simple chatscript.

Use your favourite text editor again to create the file `"amitcp:chatscript"`. The file should look something like this:

```
ECHO ON
TIMEOUT 3000
SEND "ATZ"
WAIT "OK"
SEND "ATDT123456"
WAIT "CONNECT"
WAIT "ogin"
SEND "kj"
WAIT "assword"
SEND "secret"
WAIT "PPP"
```

This chatscript first tells PPP to echo what's happening to a console window, and then to allow 60 seconds (one second = 50 ticks) for a connection to be made. After these first two lines there is a sequence of `SEND` and `WAIT` commands. Initially we send `ATZ` to our modem, which should reset it. We wait for an `OK` back from the modem.

Next we dial our ISP's phone number, 123456 in this example. We wait for the phone call to be made and for the modems to handshake. If a connection is established between our modem and the ISP modem, our modem will return `CONNECT`.

This is where things start to vary greatly from ISP to ISP. In our example we are instantly presented with `login:` and `password:` prompts to which we enter our username `kj` and password `secret` respectively. Lastly we wait for our ISP to say something like `Okay, PPP is ready...` and our last line will catch the PPP in this message and tell AmiTCP to startup.

Your ISP will do something very similar to this, but it will probably have a slightly different login process. For example it might ask you for `Username:` rather than `login:`. Or after you enter your password it might require you to send a command such as `cs1ip` or `ppp` to start whatever type of connection you require.

If your ISP has not told you exactly what the login process is you can use a basic terminal program (NComm or Term for example) to dial your ISP directly and login manually. By noting everything that your ISP sends down the line, and the order in which you must answer login questions, you should have all the information necessary to create your own custom chatscript.

Testing your connection

When you have finished installing AmiTCP and have rebooted your system the first thing you should do is test that your installation is working properly. When you type in `startnet`, if all the software is installed your modem should dial your ISP and login. Assuming no obvious error messages appear, you can test if everything is working properly by using ping.

The ping program is a simple tool that bounces Internet packets off a specified machine. So to test if your Amiga has TCP/IP installed correctly you should type `ping localhost`. (The machine you are using, your Amiga, is the localhost.) If everything is working you should see something like this:

```
Ram Disk:>ping localhost
PING localhost (127.0.0.1): 56 data bytes
64 bytes from 127.0.0.1: icmp_seq=0 ttl=255 time=2 ms
64 bytes from 127.0.0.1: icmp_seq=1 ttl=255 time=2 ms
64 bytes from 127.0.0.1: icmp_seq=2 ttl=255 time=2 ms
64 bytes from 127.0.0.1: icmp_seq=3 ttl=255 time=2 ms
64 bytes from 127.0.0.1: icmp_seq=4 ttl=255 time=2 ms
64 bytes from 127.0.0.1: icmp_seq=5 ttl=255 time=2 ms
64 bytes from 127.0.0.1: icmp_seq=6 ttl=255 time=2 ms
64 bytes from 127.0.0.1: icmp_seq=7 ttl=255 time=2 ms
- localhost ping statistics -
8 packets transmitted, 8 packets received, 0% packet loss
round-trip min/avg/max = 2/2/2 ms
Ram Disk:>
```

In this and all subsequent online examples in this section, **bold type** is used to denote commands you must type at the prompt.

The next thing you should try is contacting a machine at the other end of the phone line—at your ISP’s premises in other words. The gateway machine you entered during installation is a good choice. When you ping this address (you can use the numeric format: `ping 1.2.3.4` for example) you should see output similar to the first ping. This time, however, the number of milliseconds on the right-hand side of the output (`time=`) should be much higher. This is because each packet is now being sent down the phone line to your ISP and is then coming back again.

If pinging both your own machine and your ISP’s gateway works, it is very likely that everything is working perfectly. A couple of other places you might like to ping are *ftp.eu.net* and *ftp.uu.net*. These are the addresses of large sites in Europe and the United States—if you can ping these, not only are you connected to your ISP but you have a full Internet connection to the world!

Another useful tool supplied with AmiTCP is traceroute. This traces the path an Internet packet will take from your Amiga to a specified destination machine. So if you type in `traceroute ftp.uu.net` you should see something like:

```
Ram Disk:>traceroute ftp.uu.net
traceroute to ftp.uu.net (192.48.96.9), 30 hops max, 38 byte packets
 1 portmaster.broadcom.ie (192.107.110.252) 152 ms 140 ms 194 ms
 2 * boing.broadcom.ie (192.107.110.250) 205 ms 137 ms
 3 Dublin9.Ieunet.ie (193.120.8.17) 252 ms 262 ms 208 ms
 4 Dublin.IE.EU.net (192.111.39.2) 226 ms 398 ms 148 ms
 5 Dublin11.IE.EU.net (193.120.242.34) 365 ms 325 ms 170 ms
 6 Amsterdam2.NL.EU.net (134.222.16.1) 307 ms 287 ms 298 ms
 7 Amsterdam5.NL.EU.net (134.222.85.5) 240 ms 193 ms 179 ms
 8 Vienna2.VA.US.EU.net (134.222.228.18) 349 ms 343 ms 357 ms
 9 Vienna1.VA.Alter.Net (192.41.177.249) 286 ms 289 ms 309 ms
10 Hssi1/0.GW2.FFX1.Alter.Net (137.39.100.70) 514 ms 283 ms 298 ms
11 UUNET2-GW.UU.NET (137.39.12.34) 394 ms 293 ms 348 ms
12 ftp.uu.net (192.48.96.9) 327 ms 288 ms 370 ms
Ram Disk:>
```

The above is the route from my Amiga at home to *ftp.uu.net*. The Internet has such a large and complex network topology that your route to the same

machine will probably look completely different, so don't worry if it's not identical to mine. In general, the fewer hosts that appear in the traceroute output the better as it means that IP packets don't have to travel as far from your Amiga to their destination—so it's faster, in other words.

There is a certain amount of magic about the ping and traceroute utilities. Although they are incredibly simple and are quite old tools, you can't help but watch the lights on your modem flash as you do a ping command to a machine on the other side of the world and watch open-mouthed as an IP packet travels so far and back again in well under a second.

Your first file transfer

The amount of Amiga related software readily available on the Internet is one of the main reasons many people get connected. Let's go through the steps involved in transferring a file from an Amiga FTP site to your Amiga.

The first thing you need to know is the address of the FTP archive. The Aminet collection can be found at the sites shown below.

LOCATION	NAME	IP ADDRESS	PATH	FILES	
USA (MO)	ftp.wustl.edu	128.252.135.4	pub/aminet/	ALL	USA
(WI)	ftp.netnet.net	198.70.64.3	pub/aminet/	ALL	
Australia	ftp.livewire.com.au	203.16.26.3	pub/aminet/	ALL	
Scandinavia	ftp.luth.se	130.240.18.2	pub/aminet/	ALL	
Switzerland	ftp.eunet.ch	146.228.10.16	pub/aminet/	ALL	
Switzerland	ftp.math.ethz.ch	129.132.104.6	pub/aminet/	1000	
Germany	kelly.uni-paderborn.de	131.234.128.206	pub/aminet/	ALL	
Germany	ftp.uni-paderborn.de	131.234.2.42	pub/aminet/	ALL	
Germany	ftp.uni-erlangen.de	131.188.3.2	pub/aminet/	7000	
Germany	ftp.cs.tu-berlin.de	130.149.17.7	pub/aminet/	6000	
Germany	ftp.uni-trier.de	136.199.8.81	pub/aminet/	4500	
Germany	ftp.uni-stuttgart.de	129.69.18.15	cd aminet	4000	
Germany	ftp.tu-chemnitz.de	192.108.33.193	pub/aminet/	4000	
Germany	ftp.fh-augsburg.de	141.82.16.242	pub/aminet/	3000	
Germany	ftp.uni-oldenburg.de	134.106.40.9	pub/aminet/	2500	
Germany	ftp.uni-bremen.de	134.102.228.2	pub/aminet/	2000	
Germany	ftp.uni-kl.de	131.246.9.95	pub/aminet/	1500	
Germany	ftp.uni-siegen.de	141.99.128.1	pub/aminet/	1500	
Germany	ftp.rz.uni-wuerzburg.de	132.187.1.2	pub/aminet/	700	
France	ftp.cnam.fr	163.173.128.15	pub/aminet/	8000	
Portugal	ftp.ua.pt	193.136.80.6	pub/aminet/	2000	
UK	ftp.doc.ic.ac.uk	155.198.1.40	pub/aminet/	ALL	
UK	micros.hensa.ac.uk	148.88.8.84	pub/aminet/	8500	

The ncftp application is supplied with AmiTCP. It is a reasonably smart text-based FTP client. We'll use it to transfer the file "uucode.lha" from an Aminet site—let's use *src.doc.ic.ac.uk*. Open a Shell prompt on the Amiga and type `ncftp src.doc.ic.ac.uk`. The following will appear on your screen:

```
Ram Disk:>ncftp src.doc.ic.ac.uk
Guest login ok, send your complete e-mail address as password.
The Archive - SunSITE Northern Europe
=====
SunSITE Northern Europe is located at the Department of Computing,
Imperial College, London and is running on a SPARCserver 1000 (with
8 CPUs and 61 GB of RAID5 disk space) kindly donated by Sun Microsystems.
Local time is Fri Feb 16 20:07:32 1996, you are user number 348 (max 450)
Please read the README files for more information (e.g. what a .gz file
is, extensions to ftp, etc). Note that if ftp hangs or dies, try
putting a hyphen at the start of your password. Another useful feature
is the ls -sf:package command which does a quick scan of The
Archive looking for something called 'package'. Finally please note
that *ALL* transfers are logged and any misuse will be acted upon.
DISCLAIMER: Neither Imperial College nor Sun Microsystems are liable
for any use, storage or transmission of any files stored on this
archive.
Please email suggestions and questions to wizards@doc.ic.ac.uk
NOTE::
::::: PLEASE use hostname sunsite.doc.ic.ac.uk to access here.
::::: If you cannot then use the IP address: 193.63.255.1
Please read the file README
it was last modified on Wed Dec 13 23:00:20 1995 - 65 days ago
Guest login ok, access restrictions apply.
Logged into phoenix.doc.ic.ac.uk.
1.8.3 (August 27, 1994)
phoenix.doc.ic.ac.uk:/
ncftp>
```

Okay, you're now logged on to the FTP server. By default ncftp uses anonymous login which means that you will login to the server as an anonymous user without a specific user account and password. Most FTP sites use this system, with the login name being `ftp` or `anonymous` and the password being your email address—*jblogs@isp.co.uk* for example.

The next thing you should do is tell the FTP program that you want to transfer binary files as well as text files. You can do this by typing either `bin` or `type image`. After this you can change directory on the FTP server to where the Aminet collection of files are stored:

```
ncftp>bin
phoenix.doc.ic.ac.uk:/
ncftp>cd pub/aminet
Please read the file README
it was last modified on Sat Feb 10 12:28:00 1996 - 6 days ago
phoenix.doc.ic.ac.uk:/pub/aminet
ncftp>
```

Next let's see what directories are on offer by simply typing `dir`:

```
ncftp>dir
total 2727
drwxr-xr-x 21 2989 ext 1024 Feb 16 14:44 .
drwxr-xr-x 8 root root 512 Jan 20 16:30 ..
-r--r--r-- 1 2989 ext 113 Dec 6 18:15 .names
-rw-r--r-- 1 2989 ext 4383 Feb 12 06:18 CHARTS
drwxr-xr-x 1 2989 ext 1891675 Feb 16 14:43 INDEX
-rw-r--r-- 1 2989 ext 766383 Feb 16 14:44 INDEX.Z
drwxr-xr-x 1 2989 ext 1037 Feb 13 00:04 MOTD
-rw-r--r-- 1 2989 ext 19158 Feb 10 12:28 README
-rw-r--r-- 1 2989 ext 10302 Feb 16 14:37 RECENT
-rw-r--r-- 1 2989 ext 6039 Feb 16 14:37 RECENT.Z
-rw-r--r-- 1 2989 ext 7405 Feb 16 15:05 TREE
drwxr-xr-x 8 2989 ext 512 Feb 10 02:00 biz
drwxr-xr-x 21 2989 ext 512 Jan 21 14:00 comm
drwxr-xr-x 21 2989 ext 512 Jan 15 06:56 demo
drwxr-xr-x 17 2989 ext 512 Dec 13 08:13 dev
drwxr-xr-x 9 2989 ext 512 Dec 13 08:14 disk
drwxr-xr-x 10 2989 ext 512 Jan 27 02:00 docs
drwxr-xr-x 3 2989 ext 512 Dec 13 08:18 fish
drwxr-xr-x 14 2989 ext 512 Dec 13 08:14 game
drwxr-xr-x 13 2989 ext 512 Jan 21 14:09 gfx
drwxr-xr-x 5 2989 ext 512 Dec 13 08:18 hard
drwxr-xr-x 9 2989 ext 512 Oct 27 05:15 info
drwxr-xr-x 11 2989 ext 512 Feb 10 14:00 misc
drwxr-xr-x 48 2989 ext 1024 Jan 30 14:00 mods
drwxr-xr-x 6 2989 ext 512 Dec 13 08:18 mus
drwx-wx-wx 2 2989 ext 1024 Feb 16 17:00 new
drwxr-xr-x 23 2989 ext 512 Dec 13 08:22 pix
drwxr-xr-x 2 2989 ext 39424 Feb 16 15:05 recent
drwxr-xr-x 10 2989 ext 512 Dec 13 08:17 text
drwxr-xr-x 25 2989 ext 512 Dec 13 08:16 util
phoenix.doc.ic.ac.uk:/pub/aminet
ncftp>
```

We know from the previous section that “`uucode.lha`” is in the `util/arc` directory, so let's change to that directory:

```
ncftp>cd util/arc
util/arc - Archivers
phoenix.doc.ic.ac.uk:/pub/aminet/util/arc
ncftp>
```

We could get another directory listing here, but that would be a very long list of files and we already know the exact name of the file we want, so we'll just initiate the transfer:

```
ncftp>get uucode.lha
uucode.lha: .....
uucode.lha: 15894 bytes received in 32.95 seconds, 482.40 Bytes/sec.
phoenix.doc.ic.ac.uk:/pub/aminet/util/arc
ncftp>
```

Okay! It's arrived! We can close down the connection to the file server now by typing the `bye` command:

```
ncftp>bye
Ram Disk:>
```

The “`uucode.lha`” file is now on the Amiga. Let's test the archive with the Lha program (which we will have put in our C directory):

```
Ram Disk:>lha t uucode.lha
LhA Evaluation V1.38 - Copyright (c) 1991,92 Stefan Boberg.
All rights reserved. Not for commercial use.
Testing integrity of archive 'uucode.lha':
  Testing: ( 15272/ 15272) uuencode
  Testing: ( 12448/ 12448) uuencode
  Testing: ( 3018/ 3018) uuencode.man
3 files tested, all files OK.
Operation successful.
Ram Disk:>
```

Everything is fine. To install the program we simply move to a free directory on our hard disk and extract the package:

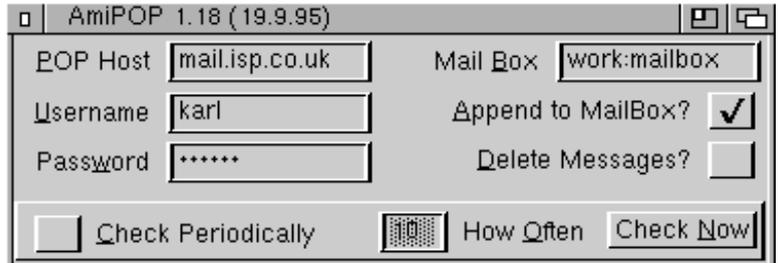
```
Ram Disk:>mkdir work:uucode
Ram Disk:>cd work:uucode
Work:uucode>lha x ram:uucode.lha
LhA Evaluation V1.38 - Copyright (c) 1991,92 Stefan Boberg.
All rights reserved. Not for commercial use.
Extracting from archive 'ram:uucode.lha':
  Extracting: ( 15272/ 15272) uuencode
  Extracting: ( 12448/ 12448) uuencode
  Extracting: ( 3018/ 3018) uuencode.man
3 files extracted, all files OK.
Operation successful.
Work:uucode>delete ram:uucode.lha
Work:uucode>
```

And that's it. The uuencode and uuencode programs are now installed in our Work:uucode directory.

Configuring email and Usenet news

Depending on how your Internet provider supplies you with email and Usenet news, two packages that you might need are “AmiPOP118.lha” (comm/net) and “IU-14-amitcp.lha” (comm/tcp).

A popular way for ISPs to provide email is to use something called the Post Office Protocol (POP). POP is just a way of connecting to an ISP machine to check if it has received any new mail sent to your email address. If your ISP uses POP they will give you the name of a machine in their building called the POP server. You can then use a program like AmiPOP to pick up your mail from this POP server.



When you use AmiPOP to connect to the POP server it will transfer any new messages for you from the POP server to your Amiga hard disk. The requester illustrated above shows how simple AmiPOP is to configure. All you need is the name of your ISP’s POP machine, along with your POP username and password. You can then specify a mailbox file anywhere on your Amiga hard disk. This is where your email will be stored on your Amiga.

The InetUtils set of programs are a little more complicated. In general, POP is used for receiving email, while a system called the Simple Mail Transfer Protocol (SMTP) is used for sending email. InetUtils provides a program called SMTPpost which can be used to connect to an ISP SMTP server machine and which will then take the email message from your Amiga and deliver it over the Internet.

InetUtils also comes with programs to transfer Usenet news articles from a Network News Transfer Protocol (NNTP) machine. The NNTP machine in your ISP is simply the machine that stores all the articles in the Usenet discussion groups.

Most Amiga email programs require the AmiPOP program for collecting new mail messages and SMTPpost to send outgoing mail. This lets the programmers of the email applications spend all their time on the usability of the email program itself rather than on how to send and receive messages.