

# NeXT System Administration Release Notes: PPP

This file contains installation instructions, configuration information, and release notes for PPP, which is new in OPENSTEP Release 4.0.

## Notes Specific to Release 4.0

This distribution is provided as part of OPENSTEP Release 4.0 as a courtesy. It is not supported by NeXT Software, Inc. Stephen Perkins authored most of the following text. Principal software developers for this distribution are Stephen Perkins, Phillipe Prindeville, and Pete French.

For more information, see the ReadMe and other text files in **/etc/ppp/Examples**. Also see the UNIX manual pages for chat(8), pppd(8), and pppstats(8).

For the most complete and up-to-date information, including FAQ (Frequently Asked Questions) files, check the following Web site:

<http://www.thoughtport.com:8080/PPP/>

Full source and binaries are also available through this site. If you don't currently have access to a Web viewer, you may get the files from **ftp.thoughtport.com (199.171.236.106)** in the directory **/pub/next/ppp/**.

Support for PPP users is provided through an electronic mailing list. To direct questions to this list, address your message to:

[nextppp@listproc.thoughtport.com](mailto:nextppp@listproc.thoughtport.com)

NeXT supplies PPP already installed. All you need to do is to configure it. If this doesn't work, see the "Troubleshooting" section later on.

## Initial Configuration

If this is the initial installation of PPP, there are several system administration steps that must be performed. These only need to be done once. These steps do not need to be reperformed for an upgrade.

- 1 Before PPP can successfully run, a module called the loadable kernel server (LKS) must be linked into the system. This is something that must be done each time the computer boots up.

So that you don't have to do this by hand each time the machine boots, you should modify a file called **/etc/kern\_loader.conf**. Because this is a system file, you must be root to perform the modification. Make a backup copy, then add the line:

```
/usr/lib/kern_loader/ppp/ppp_reloc
```

Check the output of **/usr/adm/messages** upon a successful reboot. There should be a section of output that says the PPP-2.2 LKS was successfully loaded.

- 2 You'll need to have **/etc/ppp/options** for pppd to work. You can link or copy the **options** file from **/etc/ppp/Examples** or use the **touch** command to create an empty **options** file (the only requirement is that the file exists under **/etc/ppp**). Notice that **/etc/ppp/Examples** includes sample **ip-up** and **ip-down** files you can copy and modify.
- 3 Once you start trying to make PPP connections, it is important to have access to the logging information that PPP generates. This will allow you to follow the progress of PPP and will aid in diagnosing problems. The user level process 'pppd' outputs logging information by using the standard UNIX syslog facility. Part of this facility allows you to select how much (i.e. what level of verbosity) and where (i.e. to which file) this information will be placed. While the following step is optional, it is highly recommended.

As root, make a backup copy of **/etc/syslog.conf**. You may wish to call it **/etc/syslog.conf.prePPP**. If you run into problems with the system logging error messages, you can replace **/etc/syslog.conf** with the original, reboot, and then you should be back to normal.

Now, as root, use **vi** or your favorite editor to edit the file **/etc/syslog.conf**. You need to add the line:

```
daemon.debug      /usr/adm/ppp2.2.log
```

It is imperative that you place a <tab> character between the level "daemon.debug" and the file name "/usr/adm/ppp2.2.log". Do not use spaces. If your editor converts tab characters to spaces, you need to use a different editor. Also beware of cutting and pasting between buffers. Sometimes a tab will be converted to spaces during that operation. Below is the actual contents of my **/etc/syslog.conf**:

```
daemon.debug      /usr/adm/ppp2.2.log
*.err;kern.debug;auth.notice /dev/console
kern.debug;daemon,auth.notice;*.err;mail.crit /usr/adm/messages
mark.debug,daemon.info /usr/adm/messages
lpr.debug        /usr/adm/lpd-errs
mail.info        /usr/spool/mqueue/syslog

*.alert;kern.err;daemon.err      operator
*.alert      root

*.emerg      *
```

Once you have modified **/etc/syslog.conf**, you then need to perform one more step. You need to actually create an empty logging file. This step is necessary because if syslog does not see the file, it will not create it. So, removing the file is a handy way to turn off the logging. To create an empty logging file, as root execute:

```
touch /usr/adm/ppp2.2.log
```

Upon a successful reboot, logging will be enabled for pppd (remember to specify the 'debug' option to pppd to get reasonable information sent to the logging file).

- 4 If you have a standalone machine, it is necessary for you to make sure your system is configured to correctly support a network. The following steps will help in correctly configuring your machine.

Please note, if you do not have a standalone machine (i.e. you are on a local network), you need to speak with your system administrator to determine the proper configuration for your machine. The following applies only to standalone machines.

It is suggested that you reboot your machine before you start these procedures. This is to verify that you successfully completed the previous steps and that your machine does indeed reboot. If it doesn't, you should replace **/etc/kern\_loader.conf** with the backup copy and reboot. Then, determine why your modifications failed. After successfully rebooting, you are ready to move on to these steps.

As root, make a backup copy of **/etc/hostconfig**. You might name it **/etc/hostconfig.prePPP**. If, after the following steps, you run into problems booting your machine, you may replace **/etc/hostconfig** with the backup and you should be able to successfully reboot.

Start up the **HostManager.app** (under **/NextAdmin**). After successfully entering the root password, you should get the HostManager Menu. Select the "Local..." menu option to get the main window. Modify it as follows:

For "NetInfo Binding", select "use local domain only" and make sure the box for "Readable only by local net" is checked.

For "Hostname", fill in your host name as appropriate. Do not add the domain name. For example, use "sidney" rather than "sidney.cps.msu.edu".

For "NIS Domain Name", select "None"

For "Internet Address", use the provided IP address (i.e. don't change what is there). If you don't have a value there, the NeXT default of 192.42.172.1 is a good choice. See **IP Address Selection** below for comments on the IP address selection.

For "Broadcast Address", select "Default"

For "Time Standard", select "Ignore Network Time"

For "Netmask", select "Default"

For "Router", select "None"

When this is complete, click the "Set" button. You will then have the option to reboot or to skip the reboot. Choose the appropriate action. It is suggested that you reboot immediately and verify that your system does indeed reboot (and behave as normal). If it doesn't, replace **/etc/hostconfig** with the backup copy and then reboot. Once you are back up, determine the problem with your configuration.

Please note that this description is generic and will work in most cases. There are lots of variations that people can and will need to make. Your network administrator will be able to provide you with the correct details for your particular site.

Congratulations! You have successfully installed PPP and are now ready to start up a connection. See the section "Initial Testing" for steps to verify that PPP works on your system.

## IP Address Selection

It is important to understand the fundamentals of IP addresses. Each interface attached to your computer needs its own IP address. If you have one Ethernet card, then your machine will need one IP address. If you have two Ethernet cards, then you will need two IP addresses. The PPP interface appears to the system as another interface similar to an Ethernet interface. It will also need its own IP address. However, the address we use above determines the IP address for your machine's Ethernet card. It is not the IP address associated with the PPP interface. If your PPP provider assigned you a static IP address, do not put this static address in the form above. Instead, you will need to supply this address to pppd as a command-line argument when starting up PPP. pppd will use this address to negotiate the PPP link. If your provider did not supply you with a static IP address, then don't worry. PPP will automatically negotiate the temporary use of an address each time it starts up. See the pppd man page for more information on providing IP addresses.

## Initial Testing

One of the most difficult portions of getting PPP links up and running involves writing the script that automatically dials your modem, connects to the peer, and starts the remote PPP process. Once you are connected to the peer, each PPP process will start communications and things become much easier.

Before you dive into script writing, there is a simpler solution that will allow you to test the PPP portion. Once this works, getting the dial scripts to work is a matter of sheer determination!

The mechanism is this: use a communications package (tip or kermit are good choices) to manually dial the modem and log into the remote server. There, manually start up the pppd process (the remote process, once started, will probably print some garbage on the screen: you can ignore this). Once this is done, you can exit the communications process (to free up the device it is using). Then, start your local pppd on the same device. The pppd processes will then start communicating. The premise is that you manually perform the operations that you would like your dial script to automatically perform. Once you know PPP works, you can spend time on the dial scripts.

**Note:** you must exit from your communications program before you start your local pppd. For example, if you are in kermit and you want to start the peer pppd process by hand, you must completely exit kermit before you start pppd. If you don't, the connection will not form. Further, if you find that when you exit, your modem immediately hangs up, you need to instruct the modem to ignore DTR. There is an AT command that will do this (AT&D on Supra), but you will need to check the manual for your modem to determine the correct command.

If you are planning on using kermit, Stephane I. Matis <petergun@vectrex.login.qc.ca>, has supplied this excellent definition that you may place inside your **.kermrc** file to help with PPP testing:

```
set term byte 8
# define pppd
define pppd -
```

```
!pppd < \v(line) > \v(line) defaultroute
```

To use this, add the above to your `~/.kermrc` file. Then start kermit. After you have started the remote PPP server by hand, return back to your local kermit prompt and execute the command 'do pppd'. In this particular circumstance, you will not need to exit from kermit.

Before you start initial testing, you may want to read the pppd man page. This will allow you to familiarize yourself with the some of the options available to you for starting your local pppd.

Of particular interest for most people is the 'defaultroute' option to pppd. If you have a standalone machine, then all your foreign traffic must go to the peer. Adding the 'defaultroute' option to pppd instructs pppd to set your system up in such a manner.

### Determining if the link is actually up

There are several ways to determine if the link has actually been established.

- 1 You may look at the pppd log file (typically `/usr/adm/ppp2.2.log`). If you see lines that look similar to:

```
Jan 11 23:13:38 sidney2b pppd[2141]: local IP address 35.9.12.55  
Jan 11 23:13:38 sidney2b pppd[2141]: remote IP address 35.9.10.13
```

You are probably up.

- 2 You may check the status of the PPP interface. Using the command:

```
/usr/etc/ifconfig ppp0
```

You should see that the interface is UP and that there are valid IP addresses assigned to it (0.0.0.0 is not valid). Here is an example of what you might see:

```
ppp0: flags=51<UP,POINTOPOINT,RUNNING>  
inet 35.9.12.104 --> 35.9.10.14 netmask ff000000
```

- 3 You may check the routing. When the connection comes up, you should get at least one route to the new interface. If you specified 'defaultroute' to pppd, you should also see a default route. The command for checking routes is 'netstat -rn'. Here is an example of what you might see:

```
Routing tables  
Destination Gateway Flags Refs Use Interface  
127.0.0.1 127.0.0.1 UH 2 558 lo0  
35.9.12.104 127.0.0.1 UH 2 558 lo0  
35.9.10.14 35.9.12.104 UH 0 0 ppp0  
default 35.9.10.14 UG 2 25 ppp0  
192.42.172 192.42.172.1 U 6 1939 en0
```

In the above case, the peer is 35.9.10.14 and my local machine has been assigned 35.9.12.104. All foreign traffic goes through the default route to the peer.

If you don't have an Ethernet card installed on your system, you will not have an 'en0' interface.

The routing issue is important. Discussion of this issue is outside the scope of these instructions, but I thought it might be beneficial to list a few other important tools that may help you out. The man pages can give more details:

```
/usr/etc/ping - send packets to an IP address or hostname  
traceroute - Show the route to a particular machine
```

The IP address that you use can be negotiated automatically in PPP. Unlike SLIP, you

do not have to specify an IP address when the link is brought up. If no address is specified as an argument to pppd, then PPP will negotiate the address with the peer. This is the preferred mechanism of operation. Probably the only time you should specify an IP address as an argument to pppd is if you are assigned your own IP address by your system administrator. Otherwise, sit back and let PPP do the work for you.

## A Typical PPP Session

A typical PPP session begins when you log into your system. From a Terminal window, you will run your dial script by typing its name at the prompt. If you use the Workspace manager, you can double click on the scripts icon. This will start the chat process that will dial the modem and log into the remote system. It will then turn control over to pppd. If your script is successful (as described in the "Determining if the link is actually up" section), you will be all set. All your apps, OmniWeb, FTP, telnet, etc, should work.

Once you are through using the connection, you can close down the PPP link by executing the pppdown script. Again, this can be done by typing the name of the pppdown script in a terminal window, or double clicking on the appropriate icon in the Workspace manager. At this point, PPP will terminate the phone connection and pppd will die off.

You may start and kill the PPP session as many times as you like while you are logged on. However, if you do not kill the PPP session, it will not die once you log out. Thus your telephone will remain off the hook and your computer will remain connected to the net until you log back in and shut down the connection. Turning off the computer will obviously close the connection.

## Name Resolution

Once you have a PPP connection up, you may notice that your machine will not be able to resolve the names of machines to their IP addresses.

You can check this by trying to telnet to a machine outside your local domain. If "telnet <IP address of machine>" is successful, but "telnet <machinename>" is not, then your name resolution is not configured correctly. The fix is to edit the file **/etc/resolv.conf**. This file contains two important items.

The first is your domain name. This is the name that is automatically tacked on to a computer name if you don't specify the complete name. For example if my domain is 'cps.msu.edu' and I enter 'telnet sidney', the computer will try 'telnet sidney.cps.msu.edu' (although, it will not print this name on the command line).

The second thing is a list of name servers. These should be local to your ppp provider. Your network administrator will be able to provide you with the appropriate addresses. There are default name servers to use in case the local provider is not responding. An example file might look like:

```
domain cps.msu.edu
#
# Insert local name servers here
#
#
# MSU name servers
#
```

```
nameserver 35.8.2.41
nameserver 35.8.2.41
nameserver 128.247.160.56

#
# Other servers -- hopefully something will work if needed
#shadooby.cc.umich.edu
nameserver 35.1.1.91
#ns.nasa.gov
nameserver 128.102.16.10
```

You will need to reboot your computer for the new nameservers to take effect.

## Making startup and shutdown scripts

By this time, I'm assuming that PPP has been successfully installed. However, there are a few more steps that you must perform so that using PPP (now and with future upgrades) will be convenient and easy. These steps are optional, but they are highly recommended.

Make scripts 'pppup' and 'pppdown' that bring up and shutdown PPP connections. There are a number of example scripts that you can copy and modify. Reading the man page for 'chat' will help you understand these scripts. You should note that once these scripts are made, you probably won't need to change them for future upgrades to PPP. (Note that OPENSTEP Release 4.0 includes such scripts.)

One note that you should be very careful about. These files are shell scripts. This means that the contents are executed in a shell just as if you had typed them in by hand. You must make sure that any characters that are treated specially by the shell (such as < > | \ ) are inside quotation marks (""). Otherwise, they will be interpreted by the shell in a manner that is probably to your dislike. For example, if your peer sends you a prompt like MSUnet> you must add it to the chat portion of your script like "MSUnet>".

In order to ensure that these scripts are not removed or modified when new versions of PPP are installed, you should copy all important scripts into a new directory. I suggest **/etc/ppp/scripts**. This directory will not be modified during installation. Further, as you did earlier for the important binaries, you may want to add important scripts to a directory that is in the default UNIX Search Path. The mechanism for doing this is as follows.

Suppose you have a script 'pppup' (found in **/etc/ppp/scripts**) that you want to be available on your command line. You could execute the commands:

```
/bin/mkdirs -o root -g wheel -m 755 /usr/local/bin
```

```
ln -s /etc/ppp/scripts/pppup /usr/local/bin
```

Once you log out and log back in, this script will be available. If you want to be able to call this script from a non-root account, you need to modify the permission on the file to make it `suid root`.

**Note:** this can be a potential security hazard. See your system administrator for more details.

As mentioned above, of particular interest for most people is the 'defaultroute' option to pppd. If you have a standalone machine, then all your foreign traffic must go to the peer. Remember to add the 'defaultroute' option to pppd in your startup scripts if necessary (or place it in the **/etc/ppp/options** file).

## Troubleshooting

The best source for up to date information and solutions to common problems would be the Web site:

<http://www.thoughtport.com:8080/PPP/>

Unfortunately this is a catch-22 if you don't have net access because you can't get PPP running.

The best thing to do is to use a different machine to check the FAQs on this site. If you still have problems, please send mail to the mailing list (as described below in the **Updates** section). To help with some of the basic problems, a few FAQs are included below.

Here are some common problems that people have:

### 1 pppd bombs out with an error similar to:

```
Jan 26 14:46:25 localhost pppd[256]: Connected...
Jan 26 14:46:26 localhost pppd[256]: ioctl(PPPIOCGUNIT): Inappropriate ioctl for device
Jan 26 14:46:26 localhost pppd[256]: Exit.
```

The most common error is that you are using the inappropriate device for communication to your modem. You should be using **/dev/cu~~fa~~** or **/dev/cu~~fb~~**.

### 2 PPP works fine, but when the link is up, NetInfo sleeps when you try to print or send mail.

You'll see this problem if you don't have a route from your "hostname" to localhost. In essence, your host doesn't know how to "get back to itself". There is a fix! Do this in your **rc.local**:

```
route add your_host_name localhost 0
```

with the appropriate substitution of `your_host_name`. Without it, NetInfo will sleep when trying to connect to the local domain. You will need to reboot or perform this command manually (as root) for it to take effect. (Many thanks to: Scott Bender, Pete French, and Terence Parr for help on this one.)

### 3 People who are trying to set up a NeXT as a PPP server that they dial into often complain that they can make a connection, but the remote machine can only ping the server. No other packets work.

Check out the 'proxyarp' option to pppd. Servers (connected to their LAN) must proxyarp for the remote address (i.e. the address of the machine dialing in). Further, read the file **/etc/ppp/Examples/README.NeXT** for more information on proxyarp.

### 4 Your pppd chat dials the modem but you cannot get a negotiation to start. The **/usr/adm/ppp2.2.log** log shows something similar to:

```
Mar 13 12:02:41 crystal pppd[243]: sent [LCP ConfReq id=0x1 <pcomp> <accomp>]
Mar 13 12:02:44 crystal pppd[243]: sent [LCP ConfReq id=0x1 <pcomp> <accomp>]
Mar 13 12:02:47 crystal pppd[243]: sent [LCP ConfReq id=0x1 <pcomp> <accomp>]
Mar 13 12:02:51 crystal pppd[243]: sent [LCP ConfReq id=0x1 <pcomp> <accomp>]
Mar 13 12:02:54 crystal pppd[243]: sent [LCP ConfReq id=0x1 <pcomp> <accomp>]
Mar 13 12:02:57 crystal pppd[243]: sent [LCP ConfReq id=0x1 <pcomp> <accomp>]
Mar 13 12:03:00 crystal pppd[243]: sent [LCP ConfReq id=0x1 <pcomp> <accomp>]
Mar 13 12:03:03 crystal pppd[243]: sent [LCP ConfReq id=0x1 <pcomp> <accomp>]
Mar 13 12:03:06 crystal pppd[243]: sent [LCP ConfReq id=0x1 <pcomp> <accomp>]
Mar 13 12:03:09 crystal pppd[243]: sent [LCP ConfReq id=0x1 <pcomp> <accomp>]
Mar 13 12:03:12 crystal pppd[243]: LCP: timeout sending Config-Requests
Mar 13 12:03:12 crystal pppd[243]: Connection terminated.
```



Mar 13 12:03:12 crystal pppd[243]: Serial link is not 8-bit clean:  
Mar 13 12:03:12 crystal pppd[243]: All received characters had bit 7 set to 0

This is a common problem. It is typically the result of a failure to properly start the remote PPP process. Make sure you add the '-v' option to chat (in your dial script) and then check the output in **/usr/adm/ppp2.2.log** to see why chat failed to start the remote PPP process.

- 5 Your LCP sessions get into an infinite loop while trying to negotiate the magic number. Alternately, every LCP echo you send is accompanied with a message saying that "our own echo reply was received". I'm not sure why some systems have this trouble, but the solution is to turn off magic number negotiation at your end. Add the '-mn' flag to pppd.
- 6 NXHosting applications over PPP fails. I don't know why this happens. Rest assured, that I and others do have it working. The most obvious things to check are that you have your system set up as a public window server. Also, make sure that you reset the nmsvr in **/etc/ppp/ip-up** (see the example **ip-up** file). If it still doesn't work, you might want to add your peer to your **/etc/hosts.equiv** file. I don't know if that has anything to do with it or not, but I don't know why some people can't NXHost and others can.

## Updates

You may want to join the mailing list for PPP. This will keep you informed of new releases and will provide an arena for discussing problems with the NeXT specific PPP port. To add yourself to the list (or for any other administrative requests), send an email message to:

listproc@listproc.thoughtport.com

with no subject and message body consisting of:

subscribe nextppp Your\_Own\_Name

To send mail to all the participants on the list, address your messages to:

nextppp@listproc.thoughtport.com

If you want announcements only, there is a second "announcements only" list. To subscribe to this, you may use the alternate body:

subscribe nextppp-announce Your\_Own\_Name

You don't need to subscribe to both. All announcements are forwarded to the regular list.

Any time that you have a bug report to send, please include your hardware type and the LKS version number in all reports. This number may be found in the file **/usr/adm/messages** (once the LKS has been installed). Also, for most questions, it is best to append a copy of the **/usr/adm/ppp2.2.log** file.

## Security

Security issues are not dealt with in this document. Note that the pppd file is installed suid root. This is a potential security hazard.

## **Modem Configuration**

It is important to use hardware flow control if you use a high speed modem. On my supra V.32bis modem, the command to use Hardware Flow Control is 'AT&K3'.

Also, you will probably want to set your modem so that when the DTR is dropped, the modem will disconnect. On my modem the command is 'AT&D2'. This setting disables auto-answer so if you want to allow dialins, you must read your modem manual to determine the correct setting.