Outbound PPP Setup Guide
for
Solaris 2.x

By
Michael McKnight
mcknight@rbdc.rbdc.com

Version 1.7
Last Updated: January 23, 1996

After many long hours of fighting, here are the steps I took to finally get my PPP link working. There are no guarantees with this procedure list... just what finally worked for me.

This is a working document. I am by no means a writer, so I would be grateful to know if this document helped you, hurt you, and if you have any corrections, suggestions or additions. Please drop me a note to the address above and let me know!

This document is really written to try and explain how someone with a standalone workstation (ie. at home) can attach to the Internet using Solaris 2.3 for SPARC systems. It is pretty specific and doesn't really address the issues of those with machines in a more complex environment. It does not cover the setup of a PPP server or older versions of SunOS.

NOTE: This document was originally written for Solaris 2.3. I believe it holds true for newer versions of Solaris. I'm not going to keep modifying the bulk of the document... I'll start including a section at the end of the document indicating items that change with the various OS releases as I experience them and as they are reported to me.

Document Home Location

The home site for this document is:

ftp.rbdc.com

It is located in the directory:

/pub/solaris

There are two versions in this directory. One is in PostScript and the other one is in HP PCL:

sol2.x_isp_ppp.ps.gz	PostScript
sol2.x_isp_ppp.pcl.gz	HP PCL

The PostScript version is in a very generic version of the language and should print easily to any printer with PostScript. The file uses two fonts, Times and Courier.

The PCL version can be printed to nearly any HP LaserJet compatible printer. This file contains raw printer data and should simply copied directly to the printer. If you only get half of each page, your printer may not have enough memory.

Both versions have been compressed using the GNU Zip utility, gzip. You can obtain this at any GNU archive site.

Configure the Modem Port

Get the modem configured for dial-out (or dial-in and out) by using *Celeste's Modem Guide for Solaris* (available at ftp.batnet.com:/pub/wombats/stokely/ftp). The key here is to be able to use the modem with tip. If you can simply talk to the modem with tip, then you know you at least have the modem hardware set up. This is also how you can configure the modem by using AT commands.

Edit the /etc/hosts file

You will need to add a few entries to the /etc/hosts file. One entry will be used to assign a void IP address to your PPP interface. The other entries will be the various addresses of your service provider. My /etc/hosts file looks like this:

```
0.0.0.0            void                      # PPP interface, void IP assignment
123.456.789.1      router.rbdc.com           # Internet Provider's IP Router
123.456.789.2      rbdc.rbdc.com             # Internet Provider's UNIX Host
123.456.789.3      xyplex0.rbdc.com xyplex   # Internet Provider's Xyplex Terminal Server
```

In my case, I don't actually dial into my provider's UNIX machine. Instead, I dial into a terminal server that offers all the functionality that I need. With this kind of set-up, my provider can take down his UNIX machine for maintenance and I can still access the Internet by simply using the terminal server, the router and a backup nameserver (explained later).

Edit the /etc/asppp.cf file

This is the file that aspppd uses to configure the PPP interface for communications with your Internet provider. My /etc/asppp.cf file looks like this:

```
ifconfig ipdptp0 plumb void xyplex down netmask 255.255.255.0
path
        interface            ipdptp0
        peer_system_name     ppp-rbdc       #This is the name of the reference in /etc/uucp/Systems
        inactivity_timeout   1800           # 30 minutes to timeout (in seconds)
        debug_level          8
```

The ifconfig line breaks down like this:

ipdptp0	This is the name of the PPP network interface.
plumb	This tells the system that this is a TCP/IP interface. I think.
void	This is the IP address as in /etc/hosts assigned to the local side of the PPP

	interface. This value will be changed to the IP address received from the Internet provider. If you are assigned a fixed IP address, you will substitute it for the void entry in the /etc/asppp.cf file and set the hostname in the /etc/hosts file.
xyplex	This is the IP address of the terminal server as in the /etc/hosts file. This number is assigned as the remote side of the interface. You will see how these tie together.
down	Although the interface is defined, we aren't quite ready for it to receive data. By setting this to down, we keep data from going to this interface. This will also prevent aspppd from complaining about the void (0.0.0.0) IP address.
netmask	This tells the system how to interpret IP packets. It lets it know how to determine which numbers in the address are network numbers and which are node numbers. In my case, the netmask is 255.255.255.0 but it may differ for you. Check with your provider and adjust this as needed. **Steve Drach (steve.drach@eng.sun.com)** noted that this parameter is optional on single-point connections and is only required when configuring for a multi-point connection.

The lines below the ifconfig line are used to show the specifics of how the interface will be configured for PPP. The lines are explained here:

interface	This tells which interface this path section is for. Since you could have more than one PPP interface (in the example of a server), you could specify each one of them separately here. In this case, we use interface 0 (zero), hence the name ipdptp0.
peer_system_name	This is the name the system uses to lookup dial and connect information from the /etc/uucp files. This is the name of the system you are going to dial. This is not the host name of the system... it is the name used in the /etc/uucp/Systems file.
inactivity_timeout	This number is in seconds. If there is no activity on the interface within this timeout period, the connection will be dropped. This one is set to 30 minutes (1800 seconds), but your Internet provider may have a different limit. You can set this value to 0 (zero) if you don't need a timeout.
debug_level	This tells aspppd how much information about the interface to log. At level 8, it will log almost every packet and instruction sent to the interface. I used this for debugging and disabled it as soon as I had things working ok. **Be very careful not to forget to disable this... it**

can grow very large, very fast and may fill up /var. This debugging data is saved in the file /var/adm/log/asppp.log. The debug_level meanings are as follows:

0 = Errors only
1 = Minimal information
4 = Some UUCP chat-script information
5 = All UUCP chat-script information
7 = Maximum UUCP information
8 = PPP message traces
9 = Raw IP packets

Whenever you make changes to this file, you will need to stop and re-start aspppd.

Edit the UUCP Files

Solaris uses the UUCP files to control the modem when aspppd needs it. So, now we have to update our UUCP files.

/etc/uucp/Dialers

Add a line in /etc/uucp/Dialers to control your modem. If your modem type is already listed in the file, copy the configuration to something unique so if you have to change it later, you wont mess up the original setting.

For me, I took the hayes entry and copied it to supra (since I have a Supra 28.8 v.34 modem) and then I modified it. For example:

```
hayes   =,-, "" \dA\pTE1V1X1Q0S2=255S12=255\r\c OK\r \EATDT\T\r\c CONNECT

supra   =,-, "" ATDT\T\r\c CONNECT
```

Notice that my entry doesn't have anything special in it at all. I tried to keep my configuration as basic as possible. If your modem requires special settings for optimization or compatibility with the modem you will be calling, this is where you would put them.

/etc/uucp/Devices

Here we need to set up the device type and relate it to the modem. I took these entries right out of the *AnswerBook* and they worked fine for me. Simply add these lines to the end of the /etc/uucp/Devices file:

```
ACUEC          cua/a - 9600      supra
ACUEC          cua/a - 19200     supra
ACUEC          cua/a - 38400     supra
```

Notice that I referenced the modem configuration name that I had set up in /etc/uucp/Dialers and that I am using serial port cua/a. If you are using another port, simply put it in as required in place of the cua/a part.

/etc/uucp/Systems

This file does all the magic when it comes to making the connection. This file contains a line for a remote host and the line contains information such as the phone number to dial, username and password, and any other login stuff needed. You will need to be root to modify this file. Since this file contains password information, I'd keep it that way.

I added this entry to the end of my /etc/uucp/Systems file:

```
ppp-rbdc Any ACUEC 38400 *70,5551212 "" P_ZERO name> michael\n word> mypass\n PPP> 2\n
```

The parts are defined as this:

ppp-rbdc	This is the name used in /etc/asppp.cf as the peer_system_name.
Any	This means the system can be called at any time. If you have the need to limit call times, this is one place you could do it. Check the *AnswerBook* for more details.
ACUEC	The device type it needs, as entered in /etc/uucp/Devices.
38400	The speed to set the port to (ie. baud rate).
*70,5551212	The phone number to dial. Notice the *70, is there to disable call-waiting.
""	Beat's me. **Karl S Hagen (greyhelm@engg.ksu.edu)** suggested I put it there.
P_ZERO	This sets the modem to 8-bits, NO parity, and 1 stop-bit (8N1).
name>	My provider sends the prompt username> when its time for me to log in. This is what I tell Solaris to watch for. When it sees this, it replies with the next item.
michael\n	From the item above, when Solaris receives the name> prompt, it replies with michael\n. ***The \n tells Solaris to send a carrage-return after it sends the name.***
word>	My provider then asks me for a password> and here I tell Solaris to simply look

	for the word> part of that. When received, it will send the next item in reply.
mypass\n	From the item above, when Solaris receives the word> prompt, it replies with mypass\n.
PPP>	The next thing my provider asks is whether I want an interactive session or a PPP session. The prompt ends in PPP>, so I tell Solaris to look for this.
2\n	From the item above. Since PPP is option number 2, Solaris sends a 2\n in reply to the question.

Testing the UUCP configuration

Now that the UUCP stuff is configured, we can use the cu program to test it. At a shell prompt, simply type cu <name to call>. For example:

```
cu ppp-rbdc
```

This should pick up the modem and dial the number. I was never able to get cu to take the name and password parts and continue, but it at least did pick up the phone and dial out. That's all we need to test here anyway. The cause of this was pointed out to me by **Eoin Campbell (ecampbel@iol.ie)**. Evidently, cu only reads the /etc/uucp/Dialers file. If you put the whole login sequence in the /etc/uucp/Dialers file, cu should work completely (more than we really need for PPP, but nice to know).

Once connected, type ~. and hit RETURN to exit the cu program. This will hang up the phone as well.

Routing Configuration

In my case, I have a small network set up at home. Since my IP structure at home doesn't comply with the Internet's requirements and governing bodies, I don't want my internal network traffic to pass outside... and I don't want Internet traffic to make it to my internal network. These routing concerns may differ for your site.

Edit the /etc/gateways file

This file may not exist on your system. If it doesn't, create it as root.

I didn't want routing information (RIP) broadcasting across my PPP link. To prevent this, I added the following line to my /etc/gateways file:

```
norip     ipdptp0
```

Edit the /etc/defaultrouter file

This file may not exist on your system. If it doesn't, create it as root.

Since the Internet provider will most likely be your Internet router, you will want to specify to your system to make the provider the default router. Note that this is only in my case. If you already have a default router, you may just want to configure a route to the provider manually. I added the following line to my `/etc/defaultrouter` file:

```
router.rbdc.com
```

After editing these files, you may need to reset your routing daemon. Use `ps` to find the `in.routed` programs process ID (PID) and then kill it:

```
kill -9 in.routed_PID
```

Then flush the routing tables:

```
route -f
```

Then restart the routing daemon:

```
/usr/sbin/in.routed
```

Configure DNS Client

In order to make full use of the Internet, you will want to configure your workstation as a DNS client. This will allow you to reach hosts which you have no local knowledge of (ie not in `/etc/hosts`).

Edit /etc/resolv.conf

This file may not exist on your system. If it doesn't, create it as `root`.

In this file, you simply define a domain and tell it who your nameserver is. I was unable to get this thing to work with the name of the nameserver, I had to give it an IP address. My `/etc/resolv.conf` file looks like this:

```
DOMAIN com.
nameserver          123.456.789.2     ;rbdc.rbdc.com
```

```
; If the name server above fails, try the one(s) listed below
nameserver                  120.23.1.1          ;another.nameserver.somewhere
```

Notice the period (.) after the word com... don't forget it. In my case, the nameserver IP address is the same as my Internet providers IP address. This may not be the same for you.

The second nameserver is used in the event my primary nameserver (rbdc.rbdc.com) fails. You can have as many nameservers as you want. They will be polled in the order they are listed until one returns data.

Edit /etc/nsswitch.conf

You will want to tell your system when to use DNS and when not to. By editing the file /etc/nsswitch.conf, you can specify when the system will look to DNS for an answer. My /etc/nsswitch.conf file looks like this:

```
#
# /etc/nsswitch.files:
#
# An example file that could be copied over to /etc/nsswitch.conf; it
# does not use any naming service.
#
# "hosts:" and "services:" in this file are used only if the /etc/netconfig
# file contains "switch.so" as a nametoaddr library for "inet" transports.

passwd:         files
group:          files
hosts:          files    dns
networks:       files
protocols:      files
rpc:            files
ethers:         files
netmasks:       files
bootparams:     files
publickey:      files
# At present there isn't a 'files' backend for netgroup;  the system will
#   figure it out pretty quickly, and won't use netgroups at all.
netgroup:       files
automount:      files
aliases:        files
services:       files
sendmailvars:   files
```

Notice the line that has dns in it. This tells the system to first look for host names in the respective /etc file, and if the data isn't found there, ask DNS for name resolution.

Once you have completed the changes to these files, you will need to restart inetd. To do this, use ps to find the PID of inetd and then type the following:

```
kill -HUP inetd_PID
```

Install the PPP Patch

I had no success whatsoever until I installed patch 101425-04. It was like the PPP daemon wasn't paying attention to the system and therefore wasn't even attempting to dial out. When I installed this patch, the first attempt afterwards dialed out.

You can check to see if you have this patch already installed by typing the following:

```
showrev -p | grep 101425-04
```

I was unable to find this patch on the net. I had to get it from my vendor. You may have to do the same. I'd put it on the net, but I'm not sure how Sun determines which patches are ok to put on the net and which ones are not allowed.

Once you obtain the patch, install it as you would any other patch. Remember, this patch is only required for Solaris 2.3.

Start the PPP Daemon

Normally this is started when you boot the system. But, should you not want to reboot or ever need to stop/start the PPP daemon, use this. You need to type the following as root:

```
/etc/init.d/asppp start
```

Use this to stop it:

```
/etc/init.d/asppp stop
```

Establish a PPP Connection

This was the trickiest part of the whole thing. What I ran into was the inability of Sun's PPP to dynamically accept an IP address assignment from the Internet provider. The first version of this document described a method to get around that. I was mistaken ;)

It wasn't until I received a very enlightening E-Mail reply from **Peter Blok (pblok@inter.NL.net)** explaining to me that if you use an original IP assignment of 0.0.0.0 for the PPP interface, aspppd will accept a replacement for that value from the remote Internet provider. Peter also provided me with two scripts he had written. These scripts allowed a pretty painless method of establishing a connection.

I gave Peter's method a try, and sure enough, the interface received a new IP address. This was wonderful; however, I still had some problems.

It seems that when the ifconfig command is issued, routes are automatically added to the routing tables. These routes either confuse or conflict with the routes I actually need in order for the connection to function correctly. I had to modify Peter's script a little for my own use. My version clears out some routes that I found to be blocking my communications... even when the PPP connection was running correctly.

Below, I have included Peter's scripts and my script. You can try both of them and see which one works best for you. You may have to modify them to satisfy your particular configuration.

This still doesn't fix Sun's PPP problems. The problem is that when you issue a PPP request, the system is to establish that connection and configure naming, routing, etc. without further action by the user. Since there are problems with Sun's PPP, these scripts must be executed **BEFORE** you try any outbound requests. If you enter an outbound request, the system will dial and connect for you, but if you have routing problems it wont work. My problems came when the routing issues were preventing my packets from making it out. If you find a solution, please send it my way!

Attach/Detach Scripts

Here are the attach scripts mentioned above.

Peter’s original attach script:

```
#!/bin/sh
#
# attach
#
detach $1
ifc_cmd=‘egrep "^ifconfig.*$1" /etc/asppp.cf‘
ifc=‘echo $ifc_cmd | awk ’{print $2}’‘
ifconfig $ifc up
route add $1 $1 0 >/dev/null
ping $1
exit 0
```

My modified attach script:

```
#!/bin/sh
#
#attach
#
ifc_cmd=‘egrep "^ifconfig.*$1" /etc/asppp.cf‘
ifc=‘echo $ifc_cmd | awk ’{print $2}’‘
ifconfig $ifc up
ping $1 > /dev/null

echo "Waiting for connection to complete..."
sleep 50
new_ifc=‘ifconfig $ifc‘
new_ip=‘echo $new_ifc | awk ’{print $6}’‘

echo "IP Address received for $ifc from $1 is $new_ip"
route delete xyplex0.rbdc.com       $new_ip
route delete rbdc.rbdc.com          $new_ip
route delete 123.456.789.0          $new_ip
route delete default                $new_ip

ping $1
exit 0
```

In my version, I made a few minor changes and then deleted a bunch of routes the system sets up for me. I also added a wait (ie. sleep 50) so my modem would have time to make a connection and the PPP interface could be configured before I deleted those routes. Note that the route shown as 123.456.789.0 is the network number of my provider and will be different for you. The scripts are pretty self explanatory.

Make sure you run /etc/init.d/asppp start before trying these scripts. The syntax for running the scripts is as follows:

```
attach xyplex
```

Where xyplex is the name of the service provider as listed in the /etc/hosts file. These scripts must be run as root.

Initial Connection

To make the initial connection, su to root and enter the following:

```
attach xyplex
```

You should hear the modem dial out and make a connection. Once the connection is made, you will see the modem lights flash a little while the login series is run and the PPP interface IP number is assigned. If you are using my version of the script, you will notice a delay and then you will see the output of the `route delete` commands. If all works well, you will finally get a reply such as `xyplex is alive`.

If you don't get the `xyplex is alive` reply, you most likely have a routing problem. If the `route delete` lines gave you a response such as:

```
delete net 123.456.789.0: gateway 123.456.789.105: No such file or directory
```

then you may want to `detach` (see script below) and try `attach` again. Sometimes that will clear up the problem. If after trying an `attach` again it still doesn't work, then you may have other routing problems.

Once you are able to get a `ping` to return, check the routing with the `netstat -r` command. This command should produce output similar to the following:

```
Routing Table:
  Destination           Gateway           Flags   Ref    Use      Interface

localhost               localhost         UH      0      349346   lo0
xyplex0.rbdc.com        rbdc1.rbdc.com    UH      6      0        ipdptp0
123.456.789.0           rbdc1.rbdc.com    U       4      1
rdbc.rdbc.com           rbdc1.rbdc.com    UH      6      1
1.0.0.0                 sun               U       2      83       le0
default                 rbdc1.rbdc.com    U       4      0
```

Note how the rbdc entries are expanded to rbdc1.rbdc.com. The rbdc1 shows that I hit his first modem this time. The important thing here is that rbdc1.rbdc.com points to the default route. rdbc1.rbdc.com is the name assigned by my Internet provider for my PPP interface (ipdptp0).

NOTE: It is important to realize that any command that needs information from a host outside of your own /etc/hosts table will ask aspppd to make a connection. So, a user who enters, for example, `ping sun.com` will cause the modem to dial out (if it isn't currently connected). This will make a connection via PPP, but will be useless because of the routing issues described above. Also, commands like netstat and mail may cause very slow responses because they will first try to establish a connection and then they will ask DNS to provide name resolution, but without correct routing you wont even be able to make it to your DNS nameserver. I don't know how to get around these problems yet. If someone out there does, please send it my way.

You may not have any of these routing problems if your workstation is not on a network.

Disconnecting

When you are done with your PPP connection, you can manually disconnect it with Peter’s detach script or you can just let it timeout. Peter’s detach script basically down’s the interface, resets aspppd and then removes some routing. The script looks like this:

```
#!/bin/sh
#
# detach
#
ifc_cmd=‘egrep "^ifconfig.*$1" /etc/asppp.cf‘
ifc=‘echo $ifc_cmd | awk ’{print $2}’‘
if [  -n "‘ifconfig  -a | grep $ifc‘" ]
then
        ifconfig $ifc down          # stop provider... down the PPP interface
        id=‘ps  -e | grep aspppd | awk ’{print $1}’‘
        if test  -n "$id"
        then
                kill  -1 $id        # stop connection... hang up the modem
        fi
        ifconfig $ifc unplumb       # get rid of all routing for the PPP interface
fi
eval $ifc_cmd
exit 0
```

To run this script, you simply type the following as root:

```
detach xyplex
```

Where xyplex is your providers name as listed in /etc/hosts. This will cause the modem to hang up and the system will then wait for an IP request needing the PPP connection.

File Summary

Here is a list of the files we have been playing with and what they do.

/etc/hosts	This is the file where names and IP addresses are assigned to various special hosts.
/etc/asppp.cf	This is the configuration file for the PPP interface.
/etc/uucp/Dialers	This is the file that explains how to dial the modem.
/etc/uucp/Devices	This is the file that maps the defined modem dialer to a UUCP device.
/etc/uucp/Systems	This is the file that contains the information needed to make a connection. Information such as phone number, login name and password, and who to call when a system name is specified.
/etc/gateways	This is where we want to make sure routing information isn't broadcast over the PPP interface.
/etc/defaultrouter	This file contains a single line naming the default IP router to use.
/etc/resolv.conf	This is where we specify Domain and nameserver information for DNS name resolution.
/etc/nsswitch.conf	This file contains a list of services and where to look to get information on those services. This is where we tell the system to ask DNS for names if we can't find them locally.

Troubleshooting Hints

This section of this document will grow as people send me problems and solutions that they overcame and how they did so (assuming people contribute to this).

- Use the `snoop` program to watch the IP traffic on your interface. This will help you see what is going on at the interface level. For example... I was able to have my Internet provider `ping` me when I was unable to `ping` him. By using `snoop`, I was able to watch his inbound pings and was able to determine that it was a routing problem preventing my `ping` replies from getting out. Use `snoop` like this to watch the PPP interface:

  ```
  snoop -d ipdptp0
  ```

- When you set `debug_level` to 8 in the `/etc/asppp.cf` file, you will get a whole lot of information about the connection. This file will show you the data received from the provider. It will show you the login conversation and packet exchange that occurs in the initial connection and any traffic from then on. Just remember to disable the debugging in the `/etc/asppp.cf` file when you finally get it working.

- If you can't get anywhere, or the modem hangs up almost right away, check the `/var/adm/log/asppp.log` file and make sure the name and password conversations were successful. You can use the command `ifconfig ipdptp0` to check and see if the connection was made correctly. You can tell this by the IP addresses listed. For example, if I issue the command `ifconfig ipdptp0` I get the following output:

  ```
  ipdptp0: flags=8d1<UP,POINTOPOINT,RUNNING,NOARP,MULTICAST> mtu 1500
           inet 199.171.83.102    > 199.171.83.3 netmask ffffff00
  ```

  The line that reads `inet` shows the IP address of my PPP interface pointing to the IP address of my Internet provider. If the first IP address is `0.0.0.0` then the connection failed to assign a new IP address to the PPP interface. This will have to be cleared up before anything else will work. The first place to look to fix this would be the login portions of the `/etc/uucp/Systems` file.

- Make sure the netmask is correct. If the netmask is wrong, IP packets will be interpreted differently on each end of the connection and it will look like data is not passing.

- Search the *AnswerBook* for PPP and read as much as you can so you will understand the processes that occur in a PPP environment.

- If you get a route entry that looks like the following, delete the entry that looks correct. This will actually cause the one with the "- -" in it to take on the correct information and will clean up the routing problem:

```
host            gateway             interface
1.2.3.4         - -                 ipdptp0
1.2.3.4         mygateway           ipdptp0
```

In this case, you would want to issue the following command:

```
route delete 1.2.3.4 mygateway
```

- If you have trouble getting an interface with a fixed IP address to work, try setting the `ifconfig` line in `/etc/asppp.cf` to `up` rather than `down`.

- Make sure that the device `/dev/ipdptp` exists. The system will complain when you try to start `aspppd` if it isn't there. If you are missing this device entry, you are probably missing the PPP packages provided by Sun on your Solaris CD-ROM. You can check for the packages by issuing following command:

```
pkginfo | grep ppp
```

You should get output similar to this:

```
system      SUNWapppr     PPP/IP Asynchronous PPP daemon configuration files
system      SUNWapppu     PPP/IP Asynchronous PPP daemon and PPP login service
system      SUNWpppk      PPP/IP and IPdialup Device Drivers
```

If you don't have these, you will need to install the packages. **When completed, you must perform a reconfiguration boot.** For example:

```
touch /reconfigure
reboot
```

or

```
halt
boot -r   (once at the prom prompt)
```

Solaris 2.4 Notes

When I upgraded to Solaris 2.4, I ran in to a couple of problems. Mainly, my attach script no longer worked like I wanted it to. Also, upon boot-up, the system complained that the default gateway couldn't be reached and wouldn't start in.routed.

First, the attach script. Mainly, it looks like 2.4 doesn't create a whole bunch of useless routing information like 2.3 did. I just removed most of the route delete commands and added an extra ping to a destination outside of my local providers network (ie. sun.com), just to make sure routing and DNS were also responding as they should. As usual, feel free to customize the scripts as you need.

My new attach script looks like this:

```
#!/bin/sh
#
# attach 2.4
#
ifc_cmd=`egrep "^ifconfig.*$1" /etc/asppp.cf`
ifc=`echo $ifc_cmd | awk '{print $2}'`
ifconfig $ifc up
route add default void 0 >/dev/null
ping $1 > /dev/null                             # trigger the interface to dial

echo "Waiting for connection to complete..."
sleep 30                                        # give the modem time to negotiate
new_ifc=`ifconfig $ifc`
new_ip=`echo $new_ifc | awk '{print $6}'`

echo "IP Address received for interface $ifc from $1 is $new_ip"
route delete default            $new_ip

ping $1                                         # can I reach my provider?
ping sun.com                                    # can I reach someone far away?

exit 0
```

To fix the problem with in.routed not starting, I just added /usr/sbin/in.routed to the end of my /sbin/rc3 file. There are more graceful methods, but this one is quick and simple!

Now I just use this new attach and the old detach scripts to make my connections. Someone told me that Solaris 2.4 had the PPP fixed and that it would correct the routing entries as it re-established a connection... I haven't had any luck with it. Maybe 2.5 will fix it?

Solaris 2.5 Notes

I have not upgraded to 2.5 yet, but I have received some feedback for the use of dynamic PPP with Solaris 2.5. Below is a copy of what I have received.

Tim Bandy (bandy@cs.umn.edu) sent this:

I did get dynamic routing working, although it has a problem with resetting after the link goes down (i.e., have a ppp connection, let it timeout. The old default route will still exist in the routing table for the following connections unless manually removed. Of course, this is only with default_route enabled in /etc/asppp.cf).

So, here is my asppp.cf file

```
        ifconfig ipdptp0 plumb void umn mtu 256 up

        path
                interface ipdptp0
                inactivity_timeout 900          # ~15 minutes
                peer_system_name umn ppp
                default_route
                lcp_mru 256
        #       negotiate_address on
                debug_level 5
```

I haven't looked too deeply into what the negotiate_address line will do if left uncommented. Anyway, to bring up a ppp link, all I do now is this:

1: Make an entry in the routing table that will use the ppp link. Note that this should not be the default route. I use "umn," which is a dns server inside my provider's domain. I also have a "void" entry in my hosts file, which is my ppp interface. Ex.

```
                        route add umn void 0
```

2: Bring up the interface.

```
                        ping umn
```

After approximately 30 seconds, the ppp interface will have an ip address, and the routing table will have an entry using the ppp interface as the default route.

```
        netstat  irn

        Name    Mtu     Net/Dest       Address        Ipkts  Ierrs Opkts  Oerrs Collis Queue
        lo0     8232    127.0.0.0      127.0.0.1      8      0     8      0     0      0
        le0     1500    192.0.0.0      192.0.0.3      0      0     218    0     0      0
        ipdptp0 1500    160.94.23.1    160.94.23.4    33     0     70     0     0      0

        Routing Table:
          Destination           Gateway           Flags  Ref   Use   Interface

        127.0.0.1            127.0.0.1            UH       0      0  lo0
        128.101.101.101          --               UH       7      2
        128.101.101.101      160.94.23.4          UH       7      0  ipdptp0
        160.94.23.1          160.94.23.4          UH       7      3  ipdptp0
        192.0.0.0            192.0.0.3            U        3     36  le0
        224.0.0.0            192.0.0.3            U        3      0  le0
        default              160.94.23.1          UG       0      3
```

Gareth Meyrick (gareth@pangloss.ucsf.edu) sent this:

I upgraded my home machine to solaris 2.5 a few weeks ago.. and have just gotten the ppp to work with dynamic ip addressing (it worked with static addressing unless, of course, someone else was using the address I chose.. not too cool).

I tried using 0.0.0.0 as the local host ip address, but could never get it to work. The modem would make the connection, and the address given by the remote host was used (at least according to ifconfig). However, there was some sort of strange routing problem the packets seemed to be getting out, but there was never a response (perhaps the remote host was responding to the _different_ ip address it suggested in negotiation).

Anyway, there's a ppp configuration directive that seems to solve this problem:

"negotiate_address on." The docs say this will work with any address except 0.0.0.0 (probably 'cause it causes an error message) soo..

From /etc/inet/hosts:

```
       1.1.1.1         void
```

From /etc/asppp.cf:

```
       ifconfig ipdptp0 plumb void calon up

       path
              inactivity_timeout  600     # Approx. 10 minutes
              interface ipdptp0
              peer_system_name calon      # The name we log in with (also in
              negotiate_address on
```

There's another directive for setting up routing information: "default_route", which adds this connection to the routing table & then removes it when the connection is history but it's kind of useless, because connections to hosts other than calon fail with a "no route to host" error.. so i've setup calon as the default router.

anyway, things seem to be working so far...

I hope these two letters help those of you with Solaris 2.5. When I get around to upgrading (ie. when I hear that it is stable), I will put together a step-by-step version of this and update this document.

Have Fun

Once you are this far, you should be able to go anywhere you would want to. You can ftp, telnet, rlogin, use archie, gopher, Netscape or Mosaic all from any shell prompt.

Again, please send me a note telling me if this document was any use to you. I'd also like to know if you see any errors or if you have any suggestions.

I plan to include a troubleshooting section if people will send me descriptions of their problems and how they solved them.

I'm also willing to help if I can, so just send me a questions and I'll try to answer them as well.

My address is: **mcknight@rbdc.rbdc.com**

I look forward to hearing from you.

Special Thanks To:

Peter Blok (pblok@inter.NL.net)	For his explanation and scripts showing how to use the dynamic IP addressing for the PPP interface.
Karl S Hagen (greyhelm@engg.ksu.edu)	For his help on getting the /etc/uucp/Systems file to work with my modem.
Eoin Campbell (ecampbel@iol.ie)	For pointing out typo's in the scripts and UUCP files, and for explaining how the cu command works.
Steve Drach (steve.drach@eng.sun.com)	For explaining the use of the netmask in a PPP environment and for providing the list of debug_level parameter meanings.
Ronald F. Guilmette (rfg@rahul.net)	For suggesting some different wording throughout the document and tips on how to configure for a fixed IP address.
Carlos Crosetti (carlos@metaxis.turbo.net)	For clearing up what the "Any" entry in the /etc/uucp/Systems file was for.
Bruce Riddle (briddle@postoffice.ptd.net)	For the problem description and resolution when the system is missing the /dev/ipdptp device.
Tim Bandy (bandy@cs.umn.edu) Gareth Meyrick (gareth@pangloss.ucsf.edu)	For contributing information on the setup of PPP for Solaris 2.5 way before I even got around to upgrading!