HOWTO.2

Configuration setup for a remote terminal session to a Linux Box through a phone or wireless connection

A: Phone connection

For more information on the configuration setup for the US Robotics Courier V.Everything modems for this application, please refer to the *HOWTO.1* notes (section 1) . The configuration outlined below has been tested for a US Robotics Courier V.Everything 33.6 modem used with RedHat Linux 5.2.

Setup procedure for remote server (dial-in):

Make sure that the `mgetty` binary exists in the `/sbin` directory.

```
File:           /etc/inittab
Line to add:    d1:2345:respawn:/sbin/mgetty -D /dev/ttyS#
```

where # is the number of the port which will be monitored by the mgetty process. This port should be dedicated to incoming calls and not be used to interface with any other devices (such as a UPS serial connection).

Note that the option `-D` is important as it forces `mgetty` to treat the modem as a DATA modem. No fax initialization is attempted.

Re-boot the machine and the `mgetty` sub-process will be started automatically by the `inittab` master process.

Set up procedure for the client (dial-out):

On a Linux box, all you need is a communication program such as `cu` to dial out to the remote server. The command below will allow you to communicate with the modem attached to your computer.:

```
cu -l/dev/ttyS#  -s19200
```

Then, issue the dial string:

```
atdtphonestring
```

Once a connection is established to the remote server, a login prompt should appear .

Note that a remote terminal session to a Linux box can also be started from any Windows or NT computer by using the Hyperterminal communication program.

Troubleshooting tips:

-if the connection is established but the login prompt from the remote machine is garbled (garbage characters are written to the screen), there is a good likelihood that at least one of the modems does not have hardware flow control on. For US Robotics Courier V.Everything phone modems, make sure dip switch 10 is *off* on both the local and remote modem.

-if you get a message such as “device busy” or “access denied” when starting the communication program, make sure that permissions are set properly on the device file you are using for the connection (i.e. `/dev/ttyS#`). From root, issue the command `chmod a+rw /dev/ttyS#`. For dial-out access, you should not use a port which is being monitored by a `getty` process checking for incoming calls. The `getty` program controls the port it is listening to, and periodically resets permissions on the device file associated with the port.

B. Wireless Connection

The configuration outlined below has been tested for a pair of Freewave DGR-115H radios used with RedHat Linux 5.2 running on both the local and remote computer.

Setup procedure for remote server:

Since the radio modems are basically transparent to the link and act as a null-modem (direct serial) connection, there is more than one way to communicate with the remote computer. A good compromise is to follow the setup procedure outlined in A. above for the remote server. The `mgetty` command should be of the form:

```
d1:2345:respawn:/sbin/mgetty –r /dev/ttyS# –s <speed>
```

where # is the number of the port which will be monitored by the `mgetty` process. This port should be dedicated to incoming calls and not be used to interface with any other devices (such as a UPS serial connection).

Note that the option `–r` is important as it informs `mgetty` that it is running on a direct line. There is no modem initialization whatsoever. The option `–s` determines the speed of the transmission. It should match that issued by the `cu` command on the client (see below).

Re-boot the machine and the `mgetty` sub-process will be started automatically by the `inittab` master process.

Setup procedure for client:

On a Linux box, all you need is a communication program such as `cu` to establish a direct serial connection to the remote server. However, since you are not using a phone modem, you do not need to issue a dial string to connect. Instead, just press <`enter`> once or twice after a successful serial port connect, and the remote computer will send the usual login prompt.

```
cu –l/dev/ttyS#  –s19200
```

Note that the speed should match the rate expected by the `mgetty` program running on the server.