

```

#!/bin/sh
#*****
#
# adsl-setup
#
# All-purpose slicing/dicing shell script to configure rp-pppoe.
#
# LIC: GPL
#
# Copyright (C) 2000 Roaring Penguin Software Inc.
#
# $Id: adsl-setup.in,v 1.5 2002/04/09 17:28:39 dfs Exp $
#*****

# From AUTOCONF
prefix=/usr
exec_prefix=/usr

# Paths to programs
IFCONFIG=/sbin/ifconfig
PPPD=@PPPD@
PPPOE=/usr/sbin/pppoe
ECHO=/bin/echo
LOGGER="/usr/bin/logger -t `basename $0`"

# Set to "C" locale so we can parse messages from commands
LANG=C
export LANG

CONFIG=/etc/ppp/pppoe.conf

# Protect created files
umask 077

copy() {
    cp $1 $2
    if [ "$?" != 0 ] ; then
        $ECHO "*** Error copying $1 to $2"
        $ECHO "*** Quitting."
        exit 1
    fi
}

$ECHO "Welcome to the Roaring Penguin ADSL client setup.  First, I will run"
$ECHO "some checks on your system to make sure the PPPoE client is installed"
$ECHO "properly..."
$ECHO ""

# Must be root
if [ "`/usr/bin/id -u`" != 0 ] ; then
    $ECHO "$0: Sorry, you must be root to run this script"
    exit 1
fi

# Prototype config file must exist
if [ ! -r "$CONFIG" ] ; then
    $ECHO "Oh, dear, I don't see the file '$CONFIG' anywhere.  Please"
    $ECHO "re-install the PPPoE client."
    exit 1
fi

```

```

# Must have pppd
if [ ! -x $PPPD ] ; then
    $ECHO "Oops, I can't execute the program '$PPPD'. You"
    $ECHO "must install the PPP software suite, version 2.3.10 or later."
    exit 1
fi

. $CONFIG

if [ "$DEMAND" = "" ] ; then
    DEMAND=no
fi

# pppoe must exist
if [ ! -x $PPPOE ] ; then
    $ECHO "Oh, dear, I can't execute the program '$PPPOE'. Please"
    $ECHO "re-install the rp-pppoe client."
    exit 1
fi

$ECHO "Looks good! Now, please enter some information:"

while [ true ] ; do
    $ECHO ""
    $ECHO "USER NAME"
    $ECHO ""
    $ECHO -n ">>> Enter your PPPoE user name (default $USER): "
    read U

    if [ "$U" = "" ] ; then
        U="$USER"
    fi

    # Under Linux, "fix" the default interface if eth1 is not available
    if test `uname -s` = "Linux" ; then
        $IFCONFIG $ETH > /dev/null 2>&1 || ETH=eth0
    fi

    $ECHO ""
    $ECHO "INTERFACE"
    $ECHO ""
    $ECHO ">>> Enter the Ethernet interface connected to the ADSL modem"
    $ECHO "For Solaris, this is likely to be something like /dev/hme0."
    $ECHO "For Linux, it will be ethn, where 'n' is a number."
    $ECHO -n "(default $ETH): "
    read E

    if [ "$E" = "" ] ; then
        E="$ETH"
    fi

    $ECHO ""
    $ECHO "Do you want the link to come up on demand, or stay up continuously?"
    $ECHO "If you want it to come up on demand, enter the idle time in seconds"
    $ECHO "after which the link should be dropped. If you want the link to"
    $ECHO "stay up permanently, enter 'no' (two letters, lower-case.)"
    $ECHO "NOTE: Demand-activated links do not interact well with dynamic IP"
    $ECHO "addresses. You may have some problems with demand-activated links."
    $ECHO -n ">>> Enter the demand value (default $DEMAND): "
    read D

```

```

if [ "$D" = "" ] ; then
    D=$DEMAND
fi

$ECHO ""
$ECHO "DNS"
$ECHO ""
$ECHO "Please enter the IP address of your ISP's primary DNS server."
$ECHO "If your ISP claims that 'the server will provide DNS addresses',"
$ECHO "enter 'server' (all lower-case) here."
$ECHO "If you just press enter, I will assume you know what you are"
$ECHO "doing and not modify your DNS setup."
$ECHO -n ">>> Enter the DNS information here: "

read DNS1

if [ "$DNS1" != "" ] ; then
    if [ "$DNS1" != "server" ] ; then
        $ECHO "Please enter the IP address of your ISP's secondary DNS server."
        $ECHO "If you just press enter, I will assume there is only one DNS server."
        $ECHO -n ">>> Enter the secondary DNS server address here: "
        read DNS2
    fi
fi

while [ true ] ; do
    $ECHO ""
    $ECHO "PASSWORD"
    $ECHO ""
    stty -echo
    $ECHO -n ">>> Please enter your PPPoE password: "
    read PWD1
    $ECHO ""
    $ECHO -n ">>> Please re-enter your PPPoE password: "
    read PWD2
    $ECHO ""
    stty echo
    if [ "$PWD1" = "$PWD2" ] ; then
        break
    fi

    $ECHO -n ">>> Sorry, the passwords do not match. Try again? (y/n)"
    read ANS
    case "$ANS" in
        N|No|NO|Non|n|no|non)
            $ECHO "OK, quitting. Bye."
            exit 1
        esac
done

# Firewalling
$ECHO ""
$ECHO "FIREWALLING"
$ECHO ""
if test `uname -s` != "Linux" ; then
    $ECHO "Sorry, firewalling is only supported under Linux. Consult"
    $ECHO "your operating system manuals for details on setting up"
    $ECHO "packet filters for your system."
    FIREWALL=NONE

```

```

else
    $ECHO "Please choose the firewall rules to use. Note that these rules are"
    $ECHO "very basic. You are strongly encouraged to use a more sophisticated"
    $ECHO "firewall setup; however, these will provide basic security. If you"
    $ECHO "are running any servers on your machine, you must choose 'NONE' and"
    $ECHO "set up firewalling yourself. Otherwise, the firewall rules will deny"
    $ECHO "access to all standard servers like Web, e-mail, ftp, etc. If you"
    $ECHO "are using SSH, the rules will block outgoing SSH connections which"
    $ECHO "allocate a privileged source port."
    $ECHO ""
    while [ true ] ; do
        $ECHO "The firewall choices are:"
        $ECHO "0 - NONE: This script will not set any firewall rules. You are responsible"
        $ECHO "        for ensuring the security of your machine. You are STRONGLY"
        $ECHO "        recommended to use some kind of firewall rules."
        $ECHO "1 - STANDALONE: Appropriate for a basic stand-alone web-surfing workstation"
        $ECHO "2 - MASQUERADE: Appropriate for a machine acting as an Internet gateway"
        $ECHO "        for a LAN"
        $ECHO -n ">>> Choose a type of firewall (0-2): "
        read a
        if [ "$a" = 0 -o "$a" = 1 -o "$a" = 2 ] ; then
            break
        fi
        $ECHO "Please enter a number from 0 to 2"
    done

    case "$a" in
        0)
            FIREWALL=NONE
            ;;
        1)
            FIREWALL=STANDALONE
            ;;
        2)
            FIREWALL=MASQUERADE
            ;;
    esac
fi

$ECHO ""
$ECHO "** Summary of what you entered **"
$ECHO ""
$ECHO "Ethernet Interface: $E"
$ECHO "User name:          $U"
if [ "$D" = "no" ] ; then
    $ECHO "Activate-on-demand: No"
else
    $ECHO "Activate-on-demand: Yes; idle timeout = $D seconds"
fi

if [ "$DNS1" != "" ] ; then
    if [ "$DNS1" = "server" ] ; then
        $ECHO "DNS addresses:      Supplied by ISP's server"
    else
        $ECHO "Primary DNS:        $DNS1"
        if [ "$DNS2" != "" ] ; then
            $ECHO "Secondary DNS:     $DNS2"
        fi
    fi
fi
else

```

```

        $ECHO "DNS:                Do not adjust"
    fi
    $ECHO "Firewalling:            $FIREWALL"
    $ECHO ""
    while [ true ] ; do
        $ECHO -n '>>> Accept these settings and adjust configuration files (y/n)? '
        read ANS
        case "ANS" in
            Y|y|yes|Yes|oui|Oui)
                ANS=y
                ;;
            N|n|no|No|non|Non)
                ANS=n
                ;;
        esac
        if [ "$ANS" = "y" -o "$ANS" = "n" ] ; then
            break
        fi
    done
    if [ "$ANS" = "y" ] ; then
        break
    fi
done

# Adjust configuration files.  First to $CONFIG

$ECHO "Adjusting $CONFIG"

copy $CONFIG $CONFIG-bak
if [ "$DNS1" = "server" ] ; then
    DNSTYPE=SERVER
    DNS1=""
    PEERDNS=yes
else
    PEERDNS=no
    if [ "$DNS1" = "" ] ; then
        DNSTYPE=NOCHANGE
    else
        DNSTYPE=SPECIFY
    fi
fi

# Where is pppd likely to put its pid?
if [ -d /var/run ] ; then
    VARRUN=/var/run
else
    VARRUN=/etc/ppp
fi

# Some #$(*& ISP's use a slash in the user name...
sed -e "s/^USER=.*&USER='$U'&" \
    -e "s/^ETH=.*&ETH='$E'&" \
    -e "s/^PIDFILE=.*&PIDFILE=\"$VARRUN/\$CF_BASE-adsl.pid\"&" \
    -e "s/^FIREWALL=.*&FIREWALL=$FIREWALL/" \
    -e "s/^DEMAND=.*&DEMAND=$D/" \
    -e "s/^DNSTYPE=.*&DNSTYPE=$DNSTYPE/" \
    -e "s/^DNS1=.*&DNS1=$DNS1/" \
    -e "s/^DNS2=.*&DNS2=$DNS2/" \
    -e "s/^PEERDNS=.*&PEERDNS=$PEERDNS/" \
    < $CONFIG-bak > $CONFIG

```

```

if [ $? != 0 ] ; then
    $ECHO "*** Error modifying $CONFIG"
    $ECHO "*** Quitting"
    exit 1
fi

if [ "$DNS1" != "" ] ; then
    if [ "$DNS1" != "server" ] ; then
        $ECHO "Adjusting /etc/resolv.conf"
        if [ -r /etc/resolv.conf ] ; then
            grep -s "MADE-BY-RP-PPPOE" /etc/resolv.conf > /dev/null 2>&1
            if [ "$?" != 0 ] ; then
                $ECHO " (But first backing it up to /etc/resolv.conf-bak)"
                copy /etc/resolv.conf /etc/resolv.conf-bak
            fi
        fi
        $ECHO "# MADE-BY-RP-PPPOE" > /etc/resolv.conf
        $ECHO "nameserver $DNS1" >> /etc/resolv.conf
        if [ "$DNS2" != "" ] ; then
            $ECHO "nameserver $DNS2" >> /etc/resolv.conf
        fi
    fi
fi

$ECHO "Adjusting /etc/ppp/pap-secrets and /etc/ppp/chap-secrets"
if [ -r /etc/ppp/pap-secrets ] ; then
    $ECHO " (But first backing it up to /etc/ppp/pap-secrets-bak)"
    copy /etc/ppp/pap-secrets /etc/ppp/pap-secrets-bak
else
    cp /dev/null /etc/ppp/pap-secrets-bak
fi
if [ -r /etc/ppp/chap-secrets ] ; then
    $ECHO " (But first backing it up to /etc/ppp/chap-secrets-bak)"
    copy /etc/ppp/chap-secrets /etc/ppp/chap-secrets-bak
else
    cp /dev/null /etc/ppp/chap-secrets-bak
fi

egrep -v "^$U|^\"$U\"" /etc/ppp/pap-secrets-bak > /etc/ppp/pap-secrets
$ECHO "\"$U\" * \\"$PWD1\"" >> /etc/ppp/pap-secrets
egrep -v "^$U|^\"$U\"" /etc/ppp/chap-secrets-bak > /etc/ppp/chap-secrets
$ECHO "\"$U\" * \\"$PWD1\"" >> /etc/ppp/chap-secrets

$ECHO ""
$ECHO ""
$ECHO ""
$ECHO "Congratulations, it should be all set up!"
$ECHO ""
$ECHO "Type 'adsl-start' to bring up your ADSL link and 'adsl-stop' to bring"
$ECHO "it down. Type 'adsl-status' to see the link status."
exit 0

```

This is a demo version of txt2pdf v.10.1
 Developed by SANFACE Software <http://www.sanface.com/>
 Available at <http://www.sanface.com/txt2pdf.html>