



by Katja and Guido Socher  
<katja/at/linuxfocus.org  
guido/at/linuxfocus.org>

*About the authors:*

Katja is the German editor of LinuxFocus. She likes Tux, film & photography and the sea. Her homepage can be found here.

Guido is a long time Linux fan and he likes Linux because it is designed by honest and open people. This is one of the reasons why we call it open source. His homepage is at [linuxfocus.org/~guido](http://linuxfocus.org/~guido)

## (X)dialog: Talking shells



*Abstract:*

Xdialog and dialog are two classic utilities to enhance your shell scripts with a graphical user interface. You will need some shell programming knowledge to understand the article. To learn about the basics of shell programming you can read our article on Shell Programming.

---

## Introduction

The UNIX shell is a very productive environment in itself and works well without a graphical user interface.

In a few cases however it makes sense to have a graphical dialog with the user. An example would be the installation dialog of a program. You have a number of options for the features to install and you can select the target directory ....

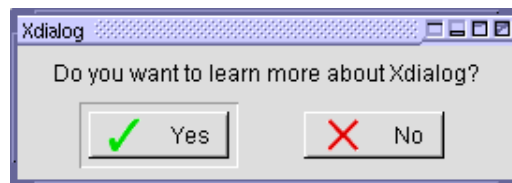
## Enter (X)dialog...

With dialog and Xdialog you can design a graphical application by writing just a short shell script. Dialog is a purely terminal based program and Xdialog is a X11 program. Here is an example:

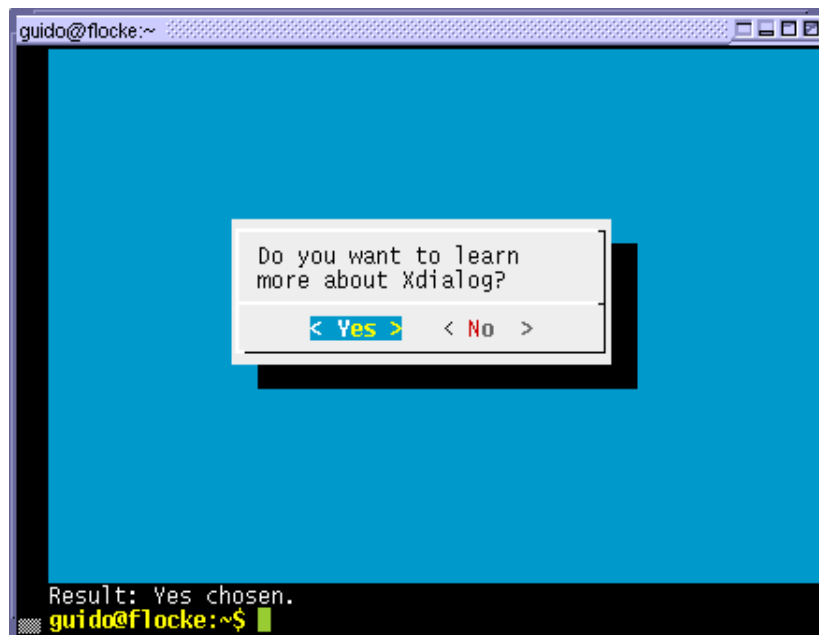
You can type (or copy/paste) the following lines into a shell window (xterm, konsole,...):

```
bash
Xdialog --yesno "Do you want to learn more about Xdialog?" 0 0;\
case $? in
0)
echo "Result: Yes chosen.";;
1)
echo "Result: No chosen.";;
255)
echo "ESC pressed.";;
esac
```

The box that appears will look like this:



If you use dialog instead of Xdialog (remove the X on the second line in the script shown above) then you get a curses based application that runs within the xterm and does not open a separate window. In some cases this is more appropriate for a shell script as it runs just within the terminal window. Especially if you want to run it remotely with a number of different hosts between your computer and the remote host where direct IP routing is not available. In this case dialog will work but you will not be able to start an X11 application like Xdialog.



The above was a pretty useless example of dialog/Xdialog but it shows how easy it is to program a

simple graphical dialog. There are more interesting dialog boxes. Boxes like calendar, menus, filemanager, progress bar, text input, message box, password dialog, ... You can run

```
dialog --help  
or  
Xdialog --help
```

to get a list of the available dialog boxes. Xdialog has a few more boxes than dialog.

## How it works

The dialog boxes are configured on the command line.

```
dialog --yesno "text string" <height> <width>
```

After typing dialog or Xdialog you have to give the name of the box you want followed by its specific parameters.

The yesno box takes 3 parameters. The <height> and <width> can be set to zero in which case the geometry of the box will be automatically adjusted to the size of the text. The result is returned as exit status to the script in the "\$?" variable. If there is more to return like e.g. the names of selected options then this is returned on standard error. Standard error is normally just printed to the screen but can be redirected with "2>".

A very simple but efficient solution.

## Real applications

Now a real world application where Xdialog/dialog really provides an advantage over conventional shell script programs: A menu where you can select different Internet Service Providers and dial out to connect to the Internet. This script requires the ppp-on/ppp-off scripts from the March 2001 article Using different ISPs for your Internet access. The script is called pppdialout and displays a different menu dependent on whether you are online or not.

```
#!/bin/sh  
#  
#DIALOG=Xdialog  
DIALOG=dialog  
#  
# name of your default isp:  
defaultisp=maxnet  
#  
error()  
{  
    echo "$1"  
    exit 2  
}  
help()  
{  
    cat <<HELP  
pppdialout -- select an ISP and dial out.
```

All available ISPs must have a config file in /etc/ppp/peers

pppdialout executes the ppp-on/ppp-off scripts as described  
in <http://linuxfocus.org/English/March2001/article192.shtml>

pppdialout, copyright gpl, <http://linuxfocus.org/English/November2002>

```
HELP
}
  exit 0
}

# parse command line:
while [ -n "$1" ]; do
case $1 in
  -h) help;shift 1;; # function help is called
  --) shift;break;; # end of options
  -*) echo "error: no such option $1. -h for help";exit 1;;
  *)  break;;
esac
done

tempfile=/tmp/pppdialout.$$
trap "rm -f $tempfile" 1 2 5 15

# check if we have a ppp network interface
if /sbin/ifconfig | grep '^ppp' > /dev/null; then
  # we are already online
  $DIALOG --title "go offline" --yesno "Click YES to \
    terminate the ppp connection" 0 0

  rval="$?"
  clear
  if [ "$rval" = "0" ]; then
    echo "running /etc/ppp/scripts/ppp-off ..."
    /etc/ppp/scripts/ppp-off
  fi
else
  # no ppp connection found, go online
  # get the names of all available ISP by listing /etc/ppp/peers
  for f in `ls /etc/ppp/peers`; do
    if [ -f "/etc/ppp/peers/$f" ]; then
      isplist="$isplist $f =="
    fi
  done
  [ -z "$isplist" ]&&error "No isp def found in /etc/ppp/peers"
  #
  $DIALOG --default-item "$defaultisp" --title "pppdialout" \
    --menu "Please select one of\
the following ISPs for dialout" 0 0 0 $isplist 2> $tempfile
  rval="$?" # return status, isp name will be in $tempfile
  clear
  if [ "$rval" = "0" ]; then
    isp=`cat $tempfile`
    echo "running /etc/ppp/scripts/ppp-on $isp..."
    /etc/ppp/scripts/ppp-on "$isp"
  else
    echo "Cancel..."
  fi
  rm -f $tempfile
fi
# end of pppdialout
```

**How the script works:**

At the beginning we define some functions, error and help, next we check for commandline arguments then a name for a temporary file is defined (/tmp/pppdialout.\$\$). \$\$ is the name of the current process and is a unique number for every computer. The trap statement is executed if the program terminates abnormally (like the user presses ctrl-C) and deletes the tempfile in our case. After that we check if we are already online or not (command: /sbin/ifconfig | grep '^ppp'). If we are already online then we open a yesno-box, the one you have already seen above, and ask the user if she wants to go offline. If we are not online then a menu box is opened. We get all the available ISPs by listing the files in /etc/ppp/peers (ls /etc/ppp/peers). The syntax of the menu box is:

```
dialog --menu "text" <height> <width> <menu height> <tag1> <description> ...
```

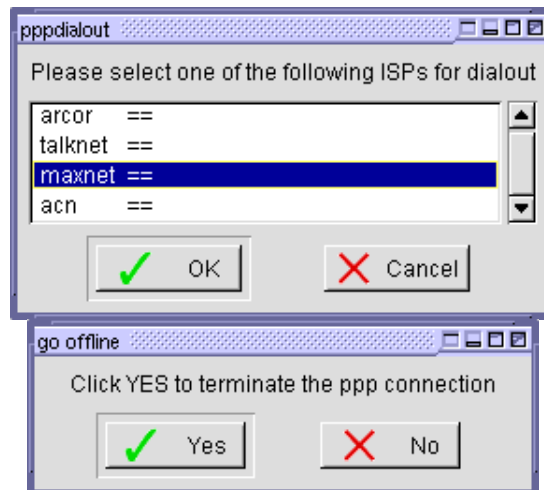
The <height>, <width> and <menu height> are again set to zero (automatic size, see above) and then the program expects pairs of strings (<tag1> <description>). We don't really have a description so we set it to something meaningless (== in this case). The data in the variable isplist will look like this:

```
isp1 == isp2 == isp3 ==
```

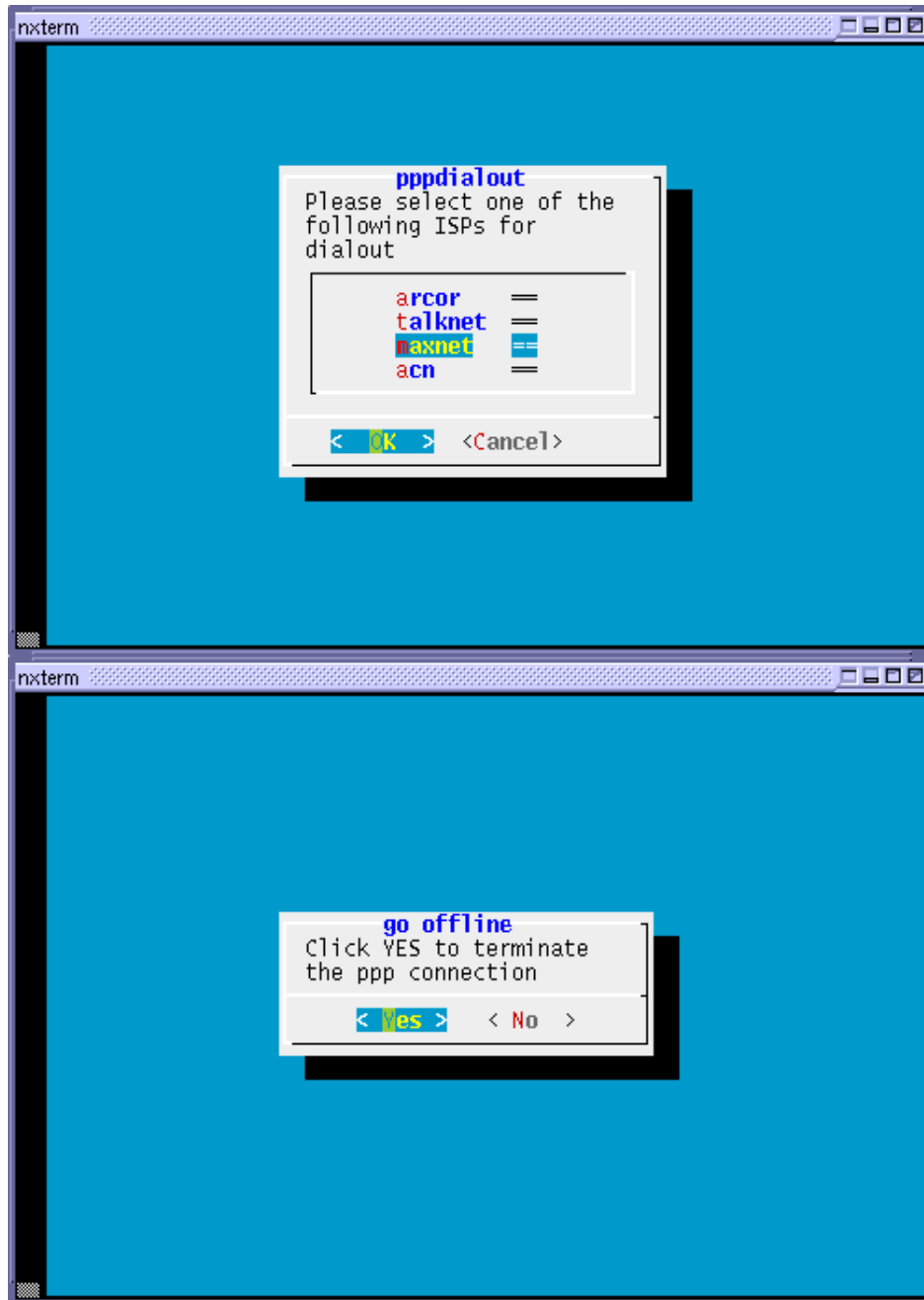
The result of the user's choice is printed by (X)dialog on standard error. The shell command "2> \$tmpfile" writes it to our tmpfile. The menu box offers also the possibility to press cancel. Therefore we have to check \$? (exit status) to find out which button is pressed.

Ok, enough theory. Here is how it looks like

... as nice GTK GUI with Xdialog:



... with the curses based dialog in the terminal:



## More applications

We have one more application for you. It is called mktgz and uses the checklist box of Xdialog. The plain terminal based dialog does not have a checklist therefore it works only with Xdialog. You can use mktgz to build tar.gz packages.

mktgz yourpackage .

This displays all files of the current working directory (".") and you can select which ones to include into the yourpackage.tar.gz package. You can download it here (mktgz.txt) We will not go through the code line by line as you should already know enough to read the script.

Xdialog and dialog come with a directory called "samples" where you can find more examples (Redhat 7.3 stores them under /usr/share/doc/Xdialog-2.0.5/samples). Be careful however as some of them really do something and are not pure demo applications.

## Conclusion

Xdialog and dialog offer a number of different dialog boxes. Not all of them are always appropriate for all types of shell scripts. The shell itself is also a very "powerful" environment. Completing a path with the tab-key can be a lot faster than searching the different directories in a GUI and clicking on them. Especially the possibility to pipeline and combine commands makes it a very powerful tool. Something like:

```
grep -i "somestring" file.txt | sort | uniq | wc -l
```

(for those not so experienced with UNIX shells: This counts all distinct lines in file.txt which contain the string "somestring") Such pipeline constructs are possible because all the commands are controlled by command line arguments. In other words: they don't stop and ask the user how she wants to continue.

There are however applications where graphical dialogs are really useful. Xdialog and dialog are very easy to use but of course not as powerful as a real graphical application. They fill the gap between a pure ASCII shell script and a full graphical application.

## Where to get Xdialog and dialog?

The CDs of your linux distribution are the first place to look for dialog and Xdialog. It could be that they are already installed on your computer (ask your computer: e.g. rpm -qil Xdialog, dpkg -L Xdialog). The homepage of Xdialog is:

<http://www.chez.com/godefroy/>

and dialog is at

<http://hightek.org/dialog/>

You can also download dialog/Xdialog from there.

## References

- Xdialog: <http://www.chez.com/godefroy/>
- dialog: <http://hightek.org/dialog/>
- Xdialog documentation: <http://www.chez.com/godefroy/doc/index.html>
- Other LinuxFocus articles:
  - Using different ISPs for your Internet access
  - Shell Programming
- pppdialout script
- mktgz script

---

Webpages maintained by the LinuxFocus Editor  
team

© Katja and Guido Socher  
"some rights reserved" see [linuxfocus.org/license/](http://linuxfocus.org/license/)  
<http://www.LinuxFocus.org>

Translation information:

en --> -- : Katja and Guido Socher <[katja@linuxfocus.org](mailto:katja@linuxfocus.org)  
[guido@linuxfocus.org](mailto:guido@linuxfocus.org)>