



**DHA Suffa University**  
**CS 206 – Operating Systems – Lab**  
**Fall 2017**



**Lab 06 - Shell Scripting in GUI with Zenity**

**Twitter: @oslabatdsu**

**Objective(s):**

- Introduction to Zenity
- Common Basic Dialog Boxes
- Zenity in Shell Scripting
- File Manipulations using Zenity
- Editing a File using Zenity GUI Commands

**What is Zenity?**

Zenity is an open source and a cross-platform application which displays GTK+ Dialog Boxes in commandline and using shell scripts. It allows to ask and present information to/from shell in Graphical Boxes. The application lets you create Graphical dialog boxes in command-line and makes the interaction between user and shell very easy. There are other alternatives, but nothing compares to the simplicity of Zenity, especially when you don't need complex programming. Zenity, a tool you must have your hands on.

**Installation of Zenity in Linux:**

Zenity is by default installed or available in repository of most of the Standard Linux distribution of today. You can check if is installed onto your machine or not by executing following commands.

```
osboxes@osboxes ~  
File Edit View Search Terminal Help  
osboxes@osboxes ~ $ zenity -- version  
You must specify a dialog type. See 'zenity --help' for details  
osboxes@osboxes ~ $ zenity --version  
3.18.1.1  
osboxes@osboxes ~ $
```

```
osboxes@osboxes ~  
File Edit View Search Terminal Help  
osboxes@osboxes ~ $ whereis zenity  
zenity: /usr/bin/zenity /usr/share/zenity /usr/share/man/man1/zenity.1.gz  
osboxes@osboxes ~ $
```

If it's not installed, you can install it using Apt as shown below

***\$ sudo apt-get install zenity***

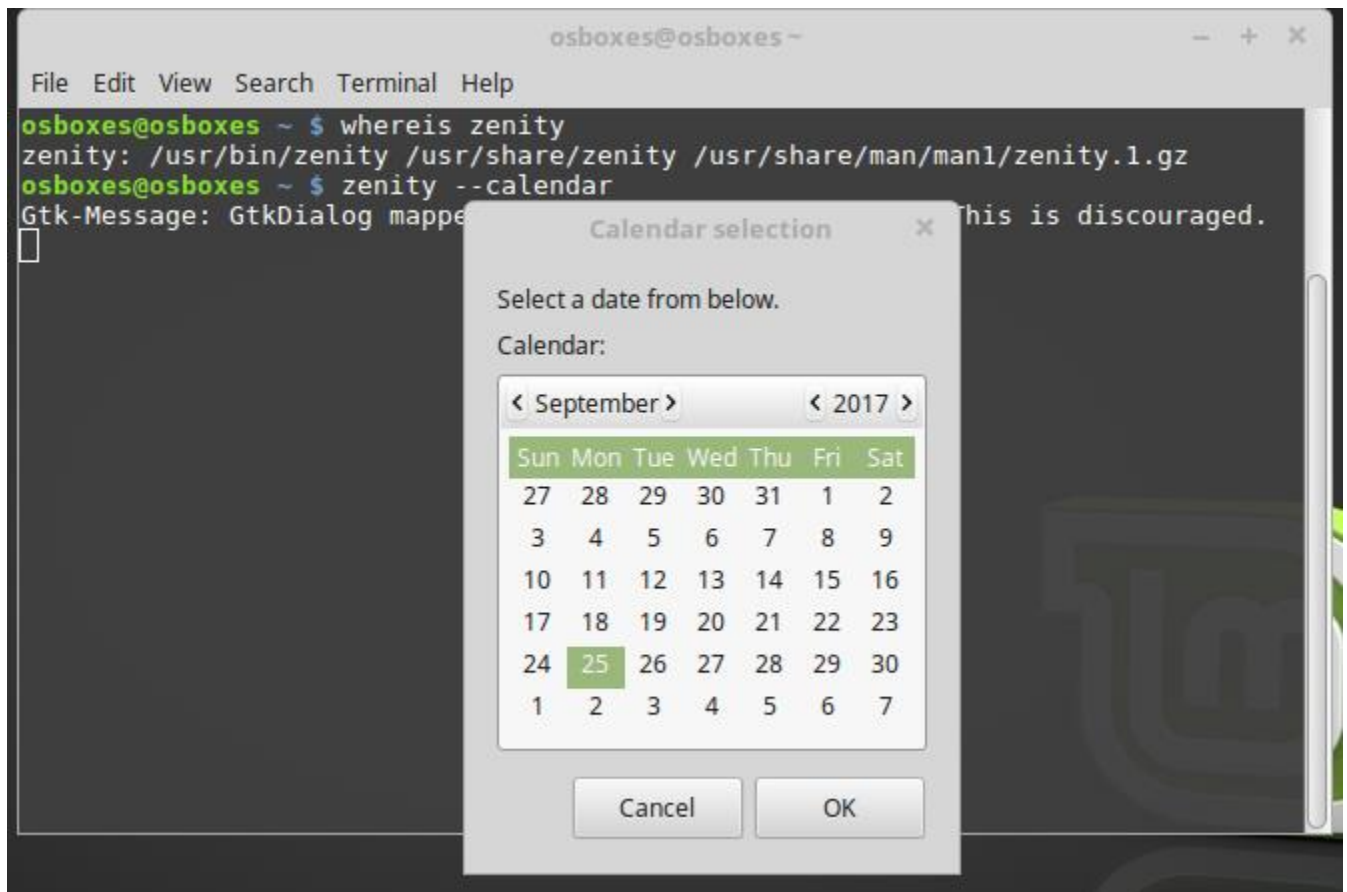
## Zenity Basic Dialog Boxes

### 1. Calendar Commands

Command: `--calendar`

What it does: Allows input and retrieval of a date.

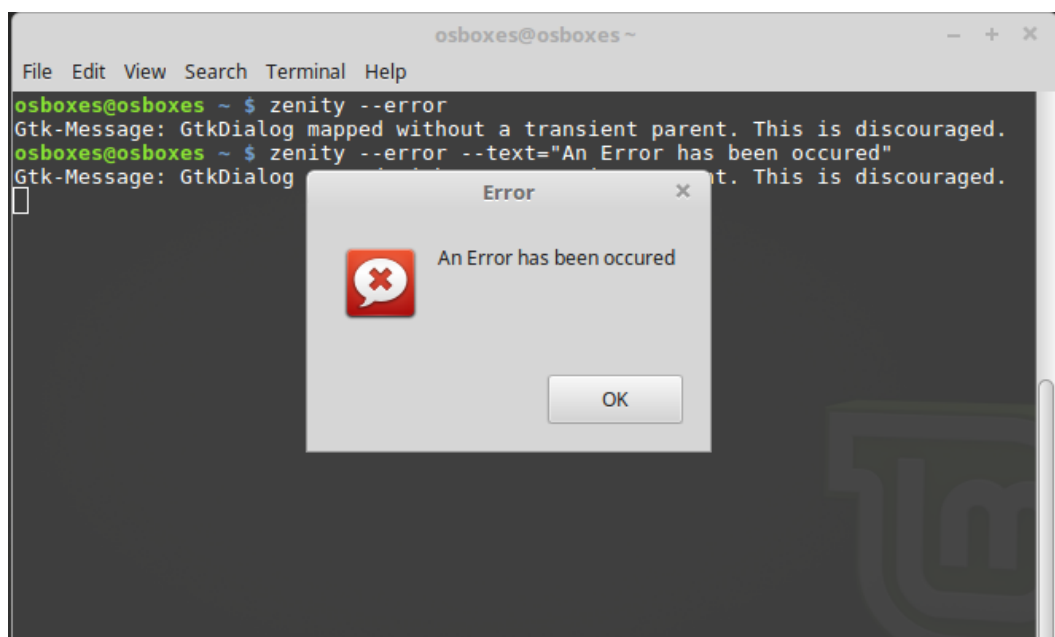
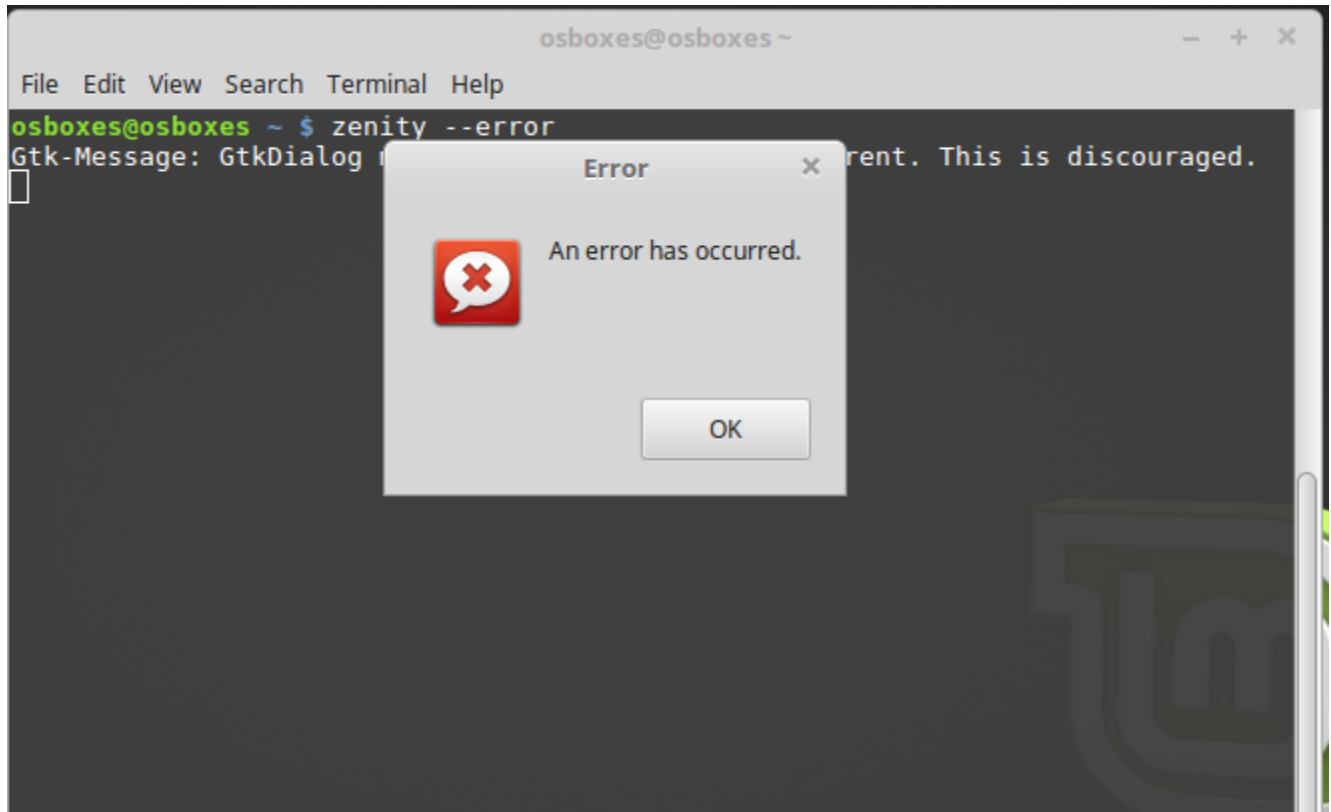
Command	Description
<code>--text</code>	displays text above the calendar
<code>--day</code>	specify a day to show on the calendar
<code>--month</code>	specify a month to show on the calendar
<code>--year</code>	specify a year to show on the calendar
<code>--date-format</code>	the format the date is returned in



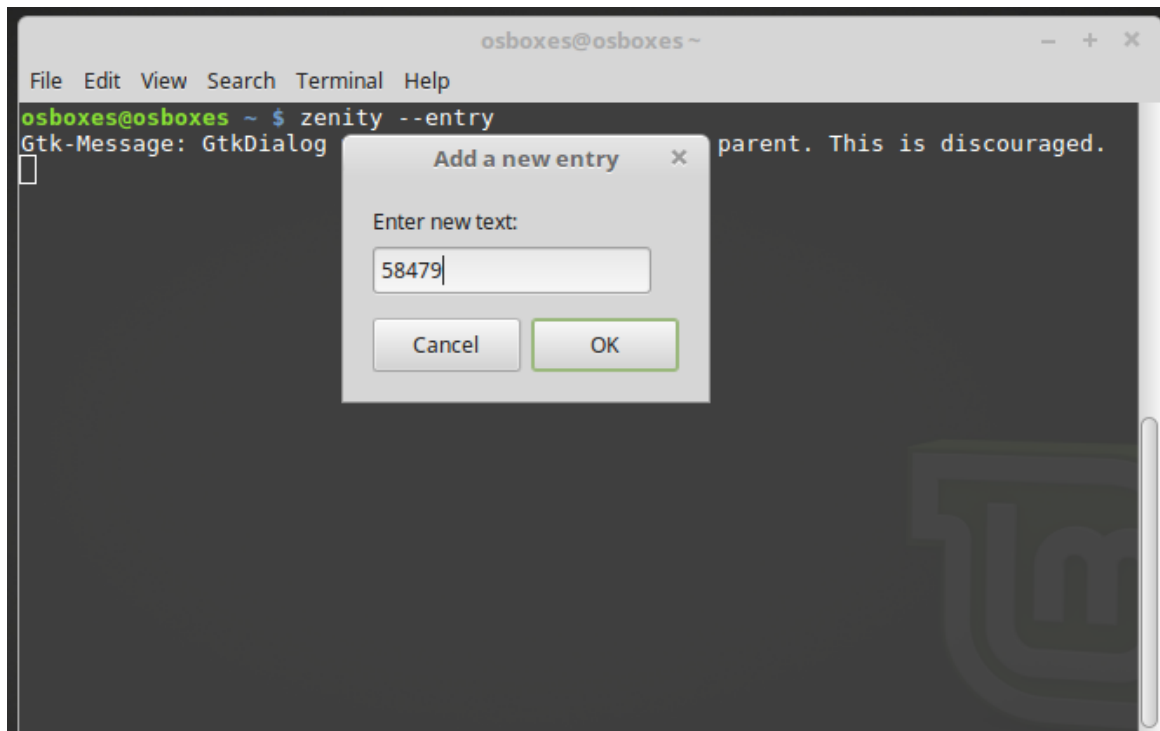
## 2. An error Dialog Box

Command: `--text`

What it does: the text to be displayed on the dialog



### 3. A General text Entry Dialog Box

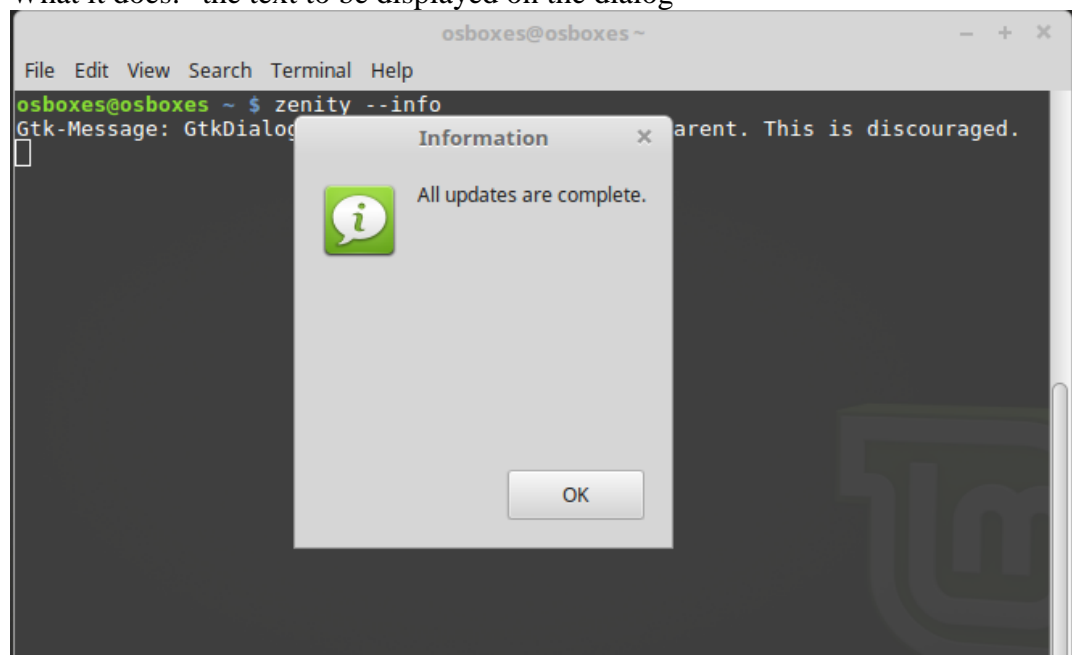


Command	Description
--text	text to be displayed above the Entry widget
--entry-text	default text to be shown within the entry

### 4. An Information Dialog

Command: --text

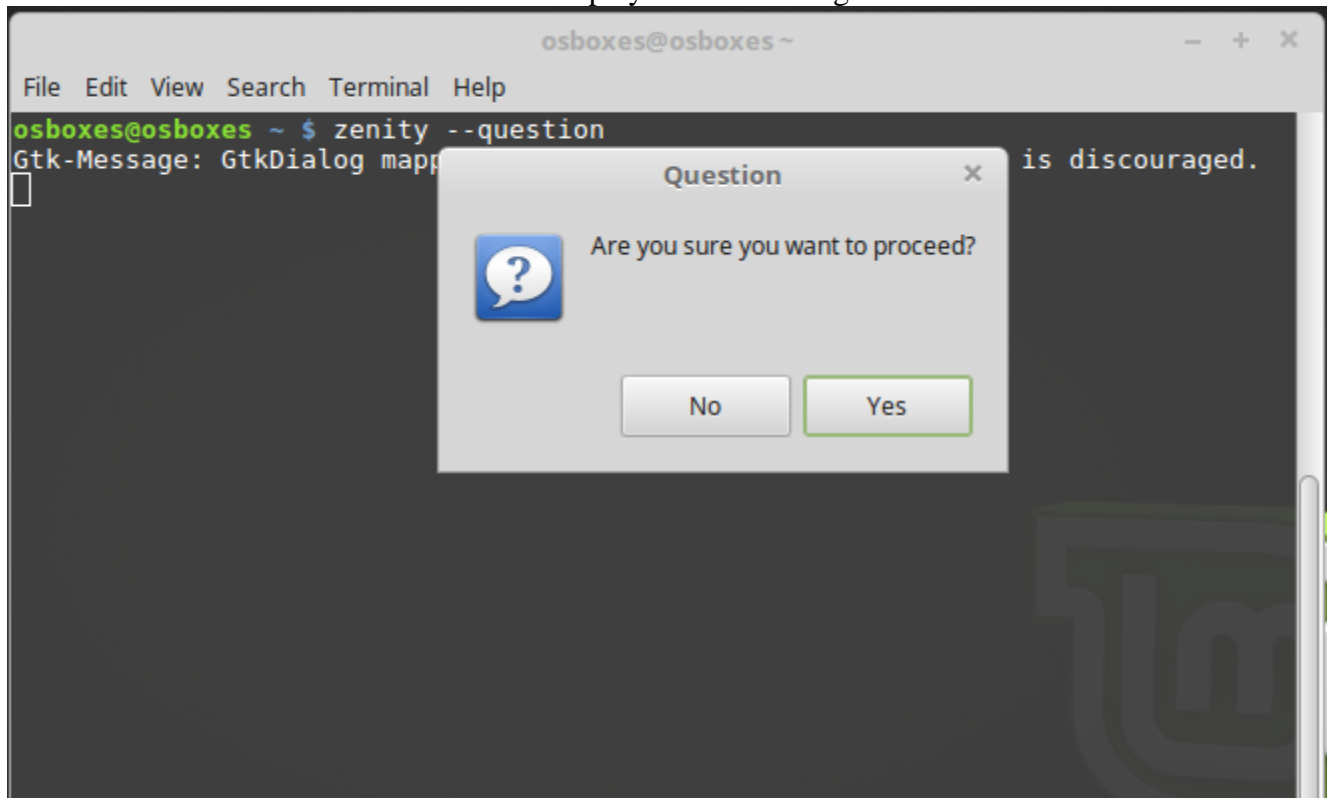
What it does: the text to be displayed on the dialog



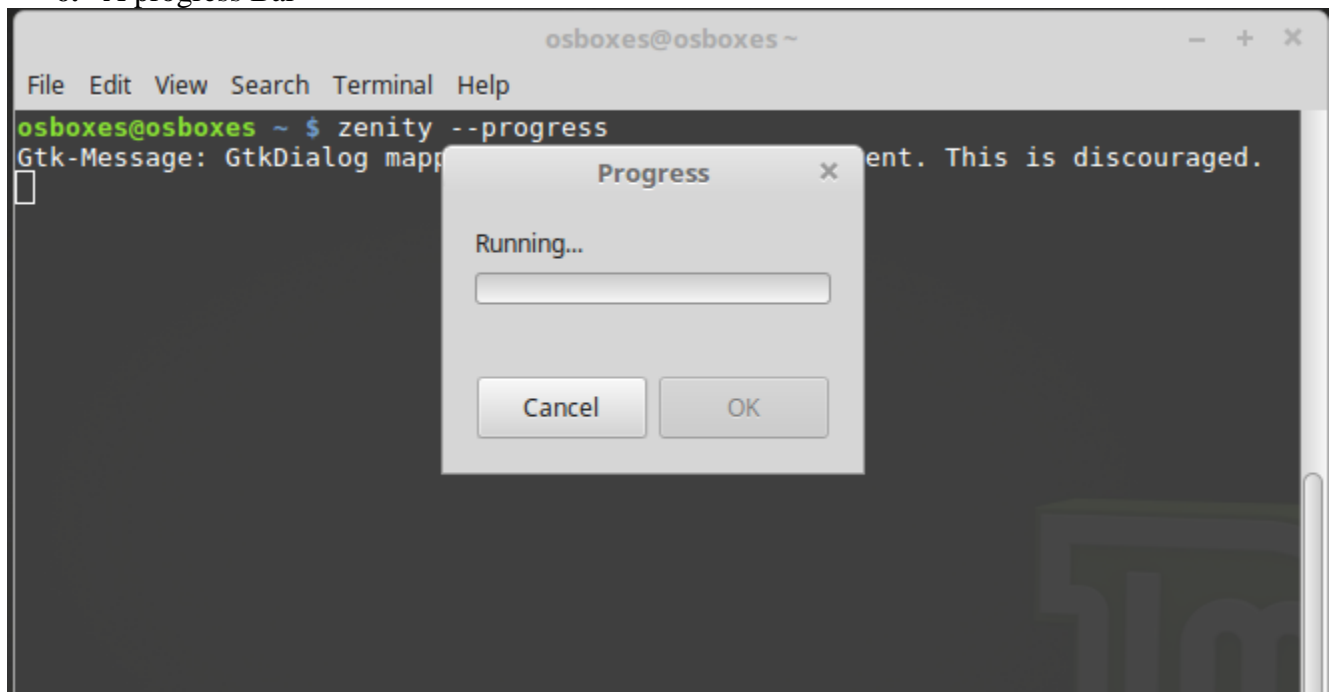
5. A question Dialog box

Command: `--text`

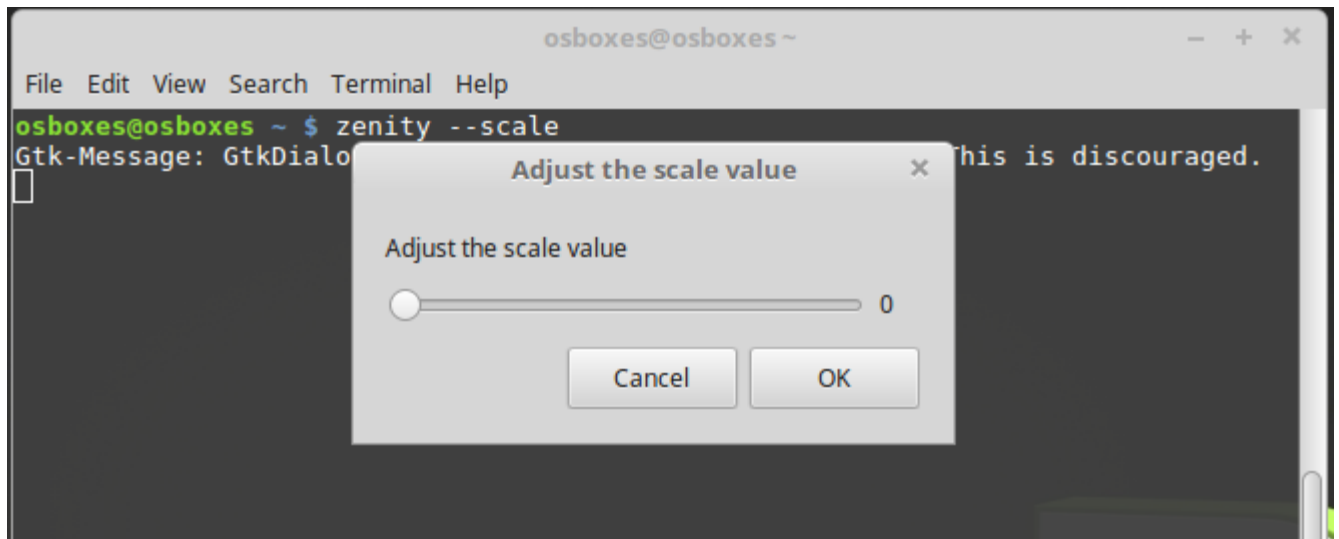
What it does: the text to be displayed on the dialog



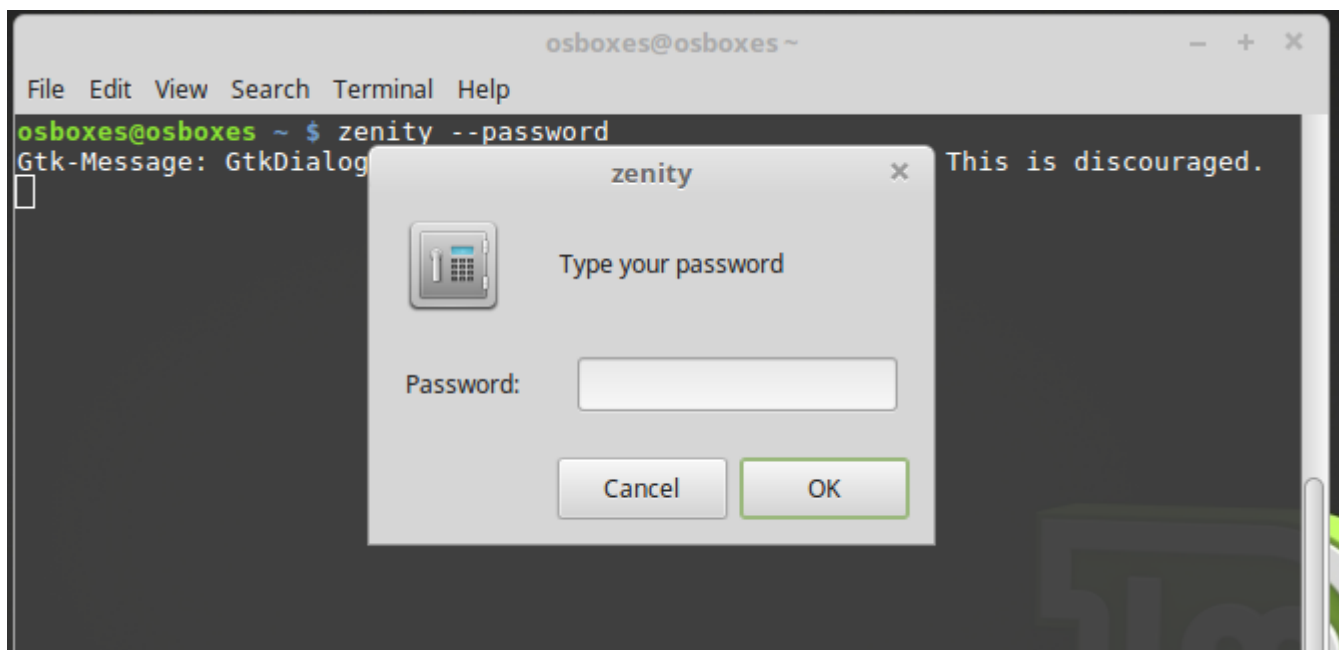
6. A progress Bar



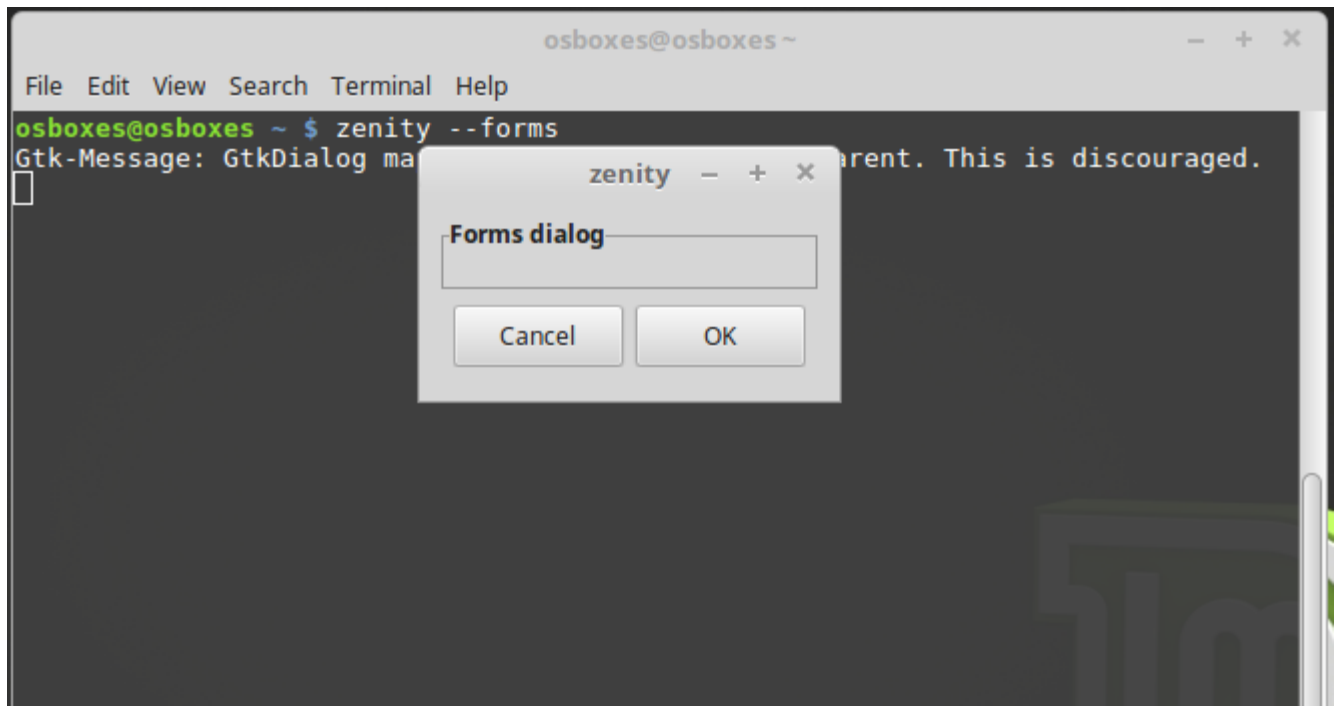
## 7. Scale Dialog



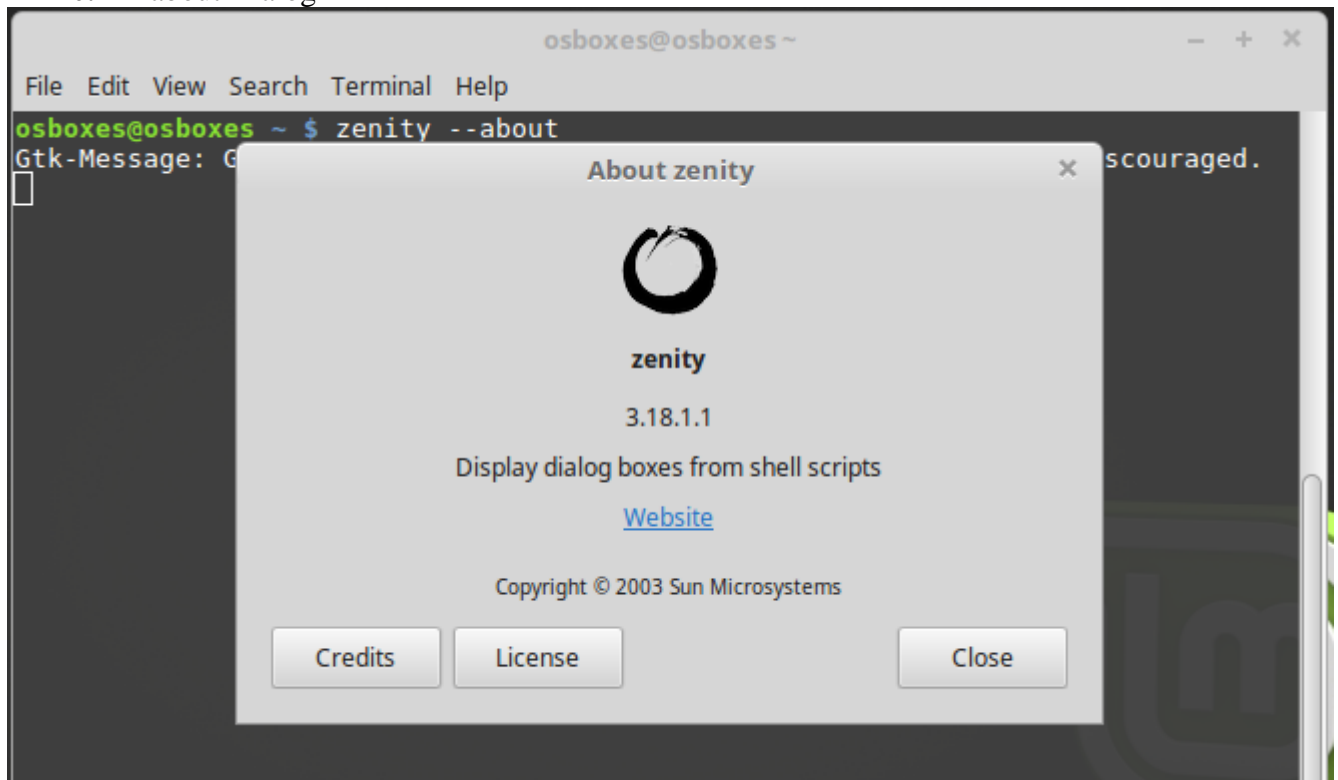
## 8. A Password Dialog



## 9. A Form Dialog box



## 10. An about Dialog



### **Example 01:**

Adding Friend's Information in a CSV File

```
zenity --forms --title="Add Friend" \
      --text="Enter information about your friend." \
      --separator="," \
      --add-entry="First Name" \
      --add-entry="Family Name" \
      --add-entry="Email" \
      --add-calendar="Birthday" >> addrecord.csv

case $? in
  0)    echo "Friend added.>";;
  1)    echo "No friend added.>";;
  -1)   echo "An unexpected error has occurred.>";;
esac
```

### **Example 02:**

Opening a selected file in gedit

```
FILE=`zenity --file-selection --title="Select a File"`
case $? in
  0)    gksudo -u root "sh -c 'gedit $FILE'";;
  1)    echo "No file selected.>";;
  -1)   echo "An unexpected error has occurred.>";;
esac
```

**Note:** If gedit is not installed, it can be installed using the following command:

*sudo apt-get install gedit*



### Example 03:

Overwriting a file's contents

```
zOpenPath="$(zenity --file-selection)"
if [ "$zOpenPath" != "" ]
then
zData=$(cat "$zOpenPath")
zNewData=$(echo -n "$zData" | zenity --text-info --editable --width
650 --height 400)
zSavePath=$(echo -n "$(zenity --file-selection --filename="$zOpenPath"
--save --confirm-overwrite)")
echo -n "$zNewData" > "$zSavePath"
fi
```

### Lab Assignment 05

**Write Shell Scripts to perform the following tasks**

1. Create a dialog box that takes filename from user then create .txt file on desktop of given filename.
2. Prompt User for Input username and check whether the user exists in the system or not.
3. Print the file selection dialog and open the file in editable mode.

### Submission Instructions:

1. Number your .sh files as question number e.g. Q1.sh, Q2.sh, etc. (Q is in upper case)
2. For every .sh file, there should be a word file containing at least three snapshots on different stages of editing the code in vi.
3. Convert your snapshot files into pdf.
4. Create a new **folder named** cs152abc where abc is your 3 digit roll #. e.g. cs152111.
5. Copy all the .sh and pdf files into this folder.
6. Right-click on the folder you created and create a **zip file** by selecting the option
  - “Send to” and selecting “Compressed (zipped) folder” [for windows].
  - “Create Archive” and change option to “.zip” instead of “.tar.gz” and click on “Create”. [for linux]Now make sure a zip file named cs152abc.zip is created e.g. cs152111.zip.
7. Upload the assignment solution on LMS under the assignment named Lab 05 Assignment – XX, where XX is your section name.
8. Due Date for assignment submission is **Oct 07, 2017**.