



Standard Dialogs

Content

1 Standard dialogs.....	3
The message window, msgbox.....	3
The dialog window, confirmDialog.....	4
The dialog box, inputbox.....	6
2. Dialogues for files and directories selection	7
Function OpenFileDialog.....	7
SaveFileDialog Function.....	8
Function openDirectoryDialog.....	9

1 Standard dialogs

The dialogues allows the interaction between the user using their own scripts, either to notify something or to ask any questions.

The message window, msgbox

This window allows gvSIG users the display of a window with a text, in which the user can't perform any action on the window beyond accept.

The window function is invoked using msgbox found in a commons dialog module, with the following syntax:

```
msgbox(mensaje [, title, messageType])
```

- message, string: The text to display in the dialog
- title, string (optional): Title of the window. There is no title by default.
- messageType, int [0-3] (optional): Determines what kind of message is shown, select an icon to be displayed in the window. It is used to emphasize the type of message being displayed to the user. The constants are:

```
FORBIDEN  
IDEA  
WARNING  
QUESTION
```

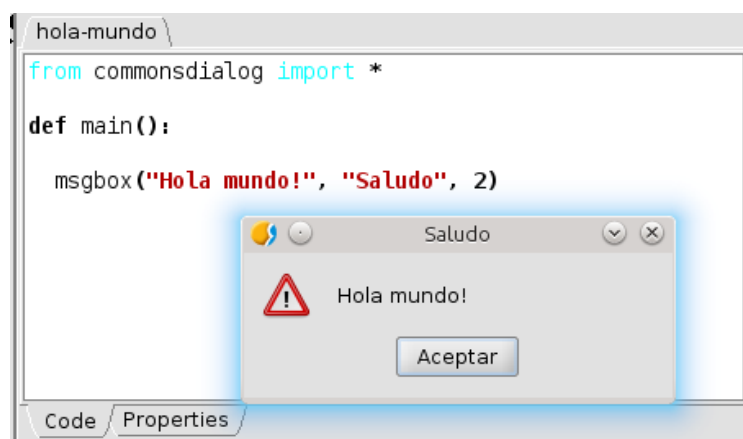
- FORBIDEN: Icon prohibited
- IDEA: Icon bulb
- WARNING: Icon attention
- QUESTION: Icon question

Within our *Hello World* script we can modify it and thus it can display instead of a text notification in the edition script area, it will display a prompt to the gvSIG user. The script code is the following (Example 1)

```
import commonsdialog  
  
def main():  
    commonsdialog.msgbox("Hola mundo!", "Saludo",  
        commonsdialog.WARNING)
```

Once entered, we'll keep with the *File / Save*, and we'll be ready to run it with the option *File / Run*.

Observe that it is present a message box with the text *Hello World!*, as a *attention symbol icon* and the title of the window is *Greetings*. In addition, the user can only press the *OK* button to continue.



msgbox Hola mundo

The dialog window, `confirmDialog`

This window allows us to show the user a message and decide among several options, not just accept.

It is invoked by `confirmDialog` function which is found in the `commonsdialog` module, with the following syntax:

```
confirmDialog(message, [title, optionType, messageType])
```

- `message`, string: The text to display in the dialog
- `title`, string (optional): Title of the window. There is no title by default.
- `optionType`, int [0-2]: Determines what buttons are available for the user to press them.

The supported constants are:

```
YES_NO  
YES_NO_CANCEL  
ACCEPT_CANCEL
```

- `YES_NO`: Yes / No buttons.
- `YES_NO_CANCEL`: buttons yes / no / cancel.
- `ACCEPT_CANCEL`: buttons to accept / cancel.
- `messageType`, int [0-3] (optional): Determines what kind of message is there showing, select an icon to be displayed in the window. It is used to emphasize the type of message being displayed to the user. The available constants are:

```
FORBIDDEN  
IDEA  
WARNING  
QUESTION
```

- FORBIDEN: Icon prohibited.
- IDEA: Light Bulb Icon
- WARNING: Icon attention.
- QUESTION: Question icon

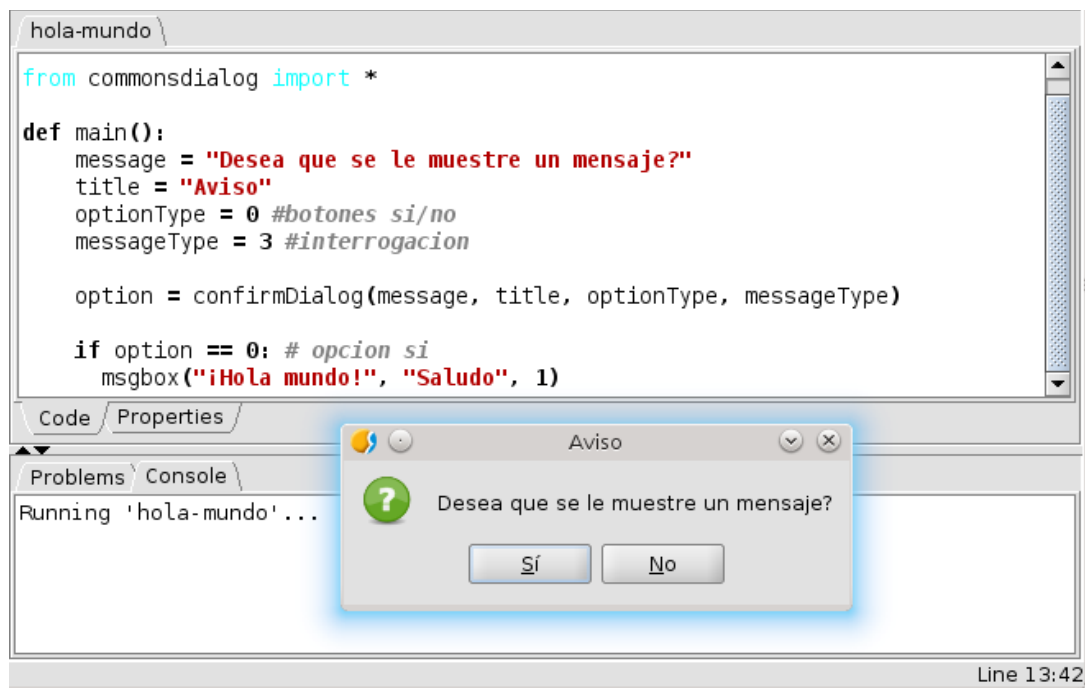
For example, there is the option where to consider whether showing or not a message to the user (example 2).

```
import commonsdialog

def main():
    message = "Desea que se le muestre un mensaje?"
    title = "Aviso"
    optionType = commonsdialog.YES_NO
    messageType = commonsdialog.QUESTION
    option = commonsdialog.confirmDialog(message, title, optionType,
                                         messageType)

    if option == 0: # opcion si
        commonsdialog.msgbox("¡Hola mundo!", "Saludo",
                             commonsdialog.IDEA)
```

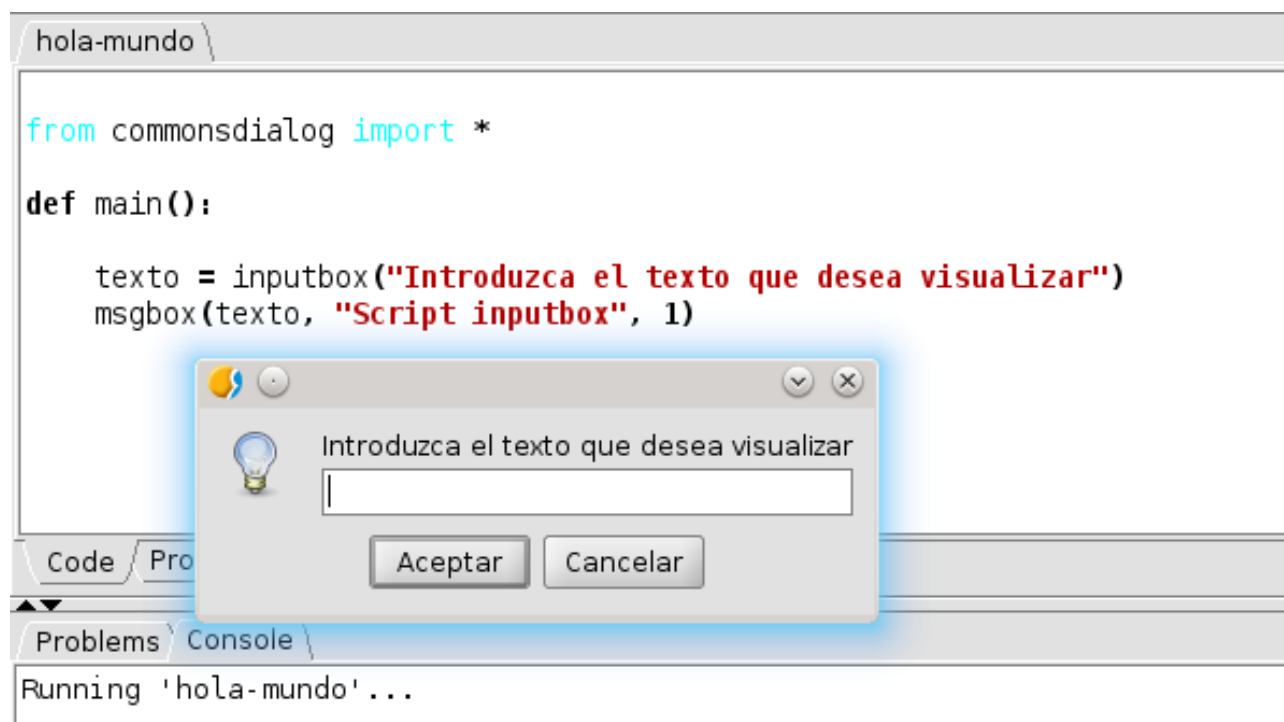
The result after running the script is shown in the following image



Dialog window confirmDialog

The dialog box, inputbox

The inputbox dialog allows us to obtain a value supplied by the user of the script and work with it. It displays a window, with a text box and two buttons; *OK* and *Cancel*. If the user clicks the *OK* button the value we get content in the box, if instead clicks *Cancel*, the return value will be *None*.



The dialog box., inputbox

The window function is invoked using *inputbox* found in the *commonsdialog* module, with the following syntax:

inputbox (message)

- message, string: Text displayed in the dialog box that should provide information on what the user should enter.

We can create a script that allows users enter a text that is displayed in a window. The code could be (example 3)

```
import commonsdialog

def main():
    titulo = "Script inputbox"
    texto = commonsdialog.inputbox(
```

```
"Enter the text you want to display")
commonsdialog.msgbox(texto, titulo, commonsdialog.IDEA)
```

When you run the script, a window will appear in which it can be introduced a text. It will be displayed when the user clicks the *OK* button.

If you do not want to show the message window in the case of having an empty text area, press *Cancel*, and simply check the value of the input box function returned.

The following script checks what the user entered. In the case if the user hasn't introduced a thing or has been pressed the *Cancel* button, a warning will be displayed, otherwise it will display the content of the text box (*Example 4*)

```
import commonsdialog
def main():
    texto = commonsdialog.inputbox(
        "Enter the text you want to display")
    if texto:
        commonsdialog.msgbox(texto, "Script inputbox",
            commonsdialog.IDEA)
    else:
        commonsdialog.msgbox(
            "You must enter a text to continue.",
            "Script inputbox",
            commonsdialog.FORBIDDEN)
```

2. Dialogues for files and directories selection

In the commonsdialog module, there are 3 functions for selecting files and directories.

The 3 functions use the same syntax, they support two optional parameters:

- title string (optional): Window's title
- initialPath string (optional): Directory to be used initially when the dialog is displayed.

If the parameter *InitialPath* is omitted or is invalid, try to show the user directory, if you can get it, open the directory where you installed gvSIG.

Function OpenFileDialog

This function displays a dialog box to select a file. It returns the absolute path of a file or None in the case the window is closed or it has been pressed the *cancel* button.

The syntax is:

```
openFileDialog([title, initialPath])
```

- title string (optional): Window's title.

- `initialPath` string (optional): Directory to be used initially when the dialog is displayed.

For example, we want to get the path of a shape that is saved in the mapping defined file in gvSIG. If no file is selected, it will be displayed a warning message (*example 5*).

You can set this folder in gvSIG from the preferences dialog, in the branch General / Folder.

```
import gvsig
import commonsdialog

def main():
    path_inicial = gvsig.getDataFolder()
    file_path = commonsdialog.openFileDialog("open File",
                                             path_inicial)
    if not file_path:
        commonsdialog.msgbox("Select a file", "Aviso",
                             commonsdialog.WARNING)
    return

#
#Code
#
```

SaveFileDialog Function

This function displays a dialog box to select a file. It returns the absolute path of a file or None in the case the windows is closed or it has been press the *cancel* button.

The syntax is:

```
saveFileDialog([title, initialPath])
```

- `title` string (opcional): Window's title.
- `initialPath` string (optional): Directory to be used initially when the dialog is displayed.

Suppose we want to obtain a path where we will store a file in the system temporary directory. The code would be (*Example 6*):

```
import tempfile
import commonsdialog
```

```

path_inicial = tempfile.gettempdir()
file_path = commonsdialog.saveFileDialog("Save File",
    path_inicial)

if not file_path:
    commonsdialog.msgbox("Select a file", "warning",
        commonsdialog.WARNING)
    return

#
#Code
#

```

Function openDirectoryDialog

This function displays a dialog box allowing you to select a directory. Returns the absolute path of a file or None is selected if you close the window or press the *cancel* button.

The syntax is:

```
openFileDialog([title, initialPath])
```

- title string (opcional): Window's title.
- initialPath string (optional): Directory to be used initially when the dialog is displayed.

For example, we want to get the path of a directory and want to show in the beggining the user directory in the system. If directory is not selected, it will display a warning message (*ejemplo 7*).

```

import commonsdialog

file_path = commonsdialog.openFolderDialog("open File")

if not file_path:
    commonsdialog.msgbox("Select a file", "Warning",
        commonsdialog.WARNING)
    return

#
#Code
#

```

gvSIG Association

Plaza Don Juan de Villarrasa 14-5,
46001, Valencia (Spain)

Registro Nacional de Asociaciones (National Register of Associations): 596206

e-mail : info@gvsig.com

Web: www.gvsig.com

Project website: <http://www.gvsig.org>

Documentation made by Víctor Acevedo. Translated by Elisabet Adeva

Distribution Lists

There are three mailing lists in order to facilitate communication between all stakeholders in gvSIG project. The first two are for users and developers, mainly oriented to the Hispanic community, with the use of the Castilian language preferred. The third list, is a international list, aimed at other communities where is preferd the use of English.

- **Members List.** . Here you can leave your opinions on the operation of the software: what things would you like to be developed, doubts related to the use of gvSIG and anything you believe should be in discussion and accommodates in the list of users. The link to subscribe to the user list is the following:

http://listserv.gva.es/mailman/listinfo/gvsig_usuarios

- **Developers list.** It is geared/oriented to anyone interested in knowing how gvSIG is developed. The link to subscribe to this list is:

http://listserv.gva.es/mailman/listinfo/gvsig_desarrolladores

- **International list.** It is intended for both users and developers witch do not speak Spanish. The preferably language used is English. The link to subscribe to this list is:

http://listserv.gva.es/mailman/listinfo/gvsig_internacional

All names of programs, operating systems, computer hardware etc., that appear in this course are trademarks of their respective companies or organizations.

© 2013 gvSIG Association

This manual is distributed under the Creative Commons Attribution-ShareAlike 3.0 Unported (<http://creativecommons.org/licenses/by-sa/3.0/deed.es>) - See conditions in Annexes