

## Introduction

Dialog control represents a top-level window with a title and a border used to take some form of input from the user.

## Class declaration

Following is the declaration for **java.awt.Dialog** class:

```
public class Dialog
extends Window
```

## Field

Following are the fields for **java.awt.Dialog** class:

- **static Dialog.ModalityType DEFAULT\_MODALITY\_TYPE** -- Default modality type for modal dialogs.

## Class constructors

| S.N. | Constructor & Description                                                                                                                                                                                |
|------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1    | <b>DialogDialogowner</b><br><br>Constructs an initially invisible, modeless Dialog with the specified owner Dialog and an empty title.                                                                   |
| 2    | <b>DialogDialogowner, Stringtitle</b><br><br>Constructs an initially invisible, modeless Dialog with the specified owner Dialog and title.                                                               |
| 3    | <b>DialogDialogowner, Stringtitle, booleanmodal</b><br><br>Constructs an initially invisible Dialog with the specified owner Dialog, title, and modality.                                                |
| 4    | <b>DialogDialogowner, Stringtitle, booleanmodal, GraphicsConfigurationgc</b><br><br>Constructs an initially invisible Dialog with the specified owner Dialog, title, modality and GraphicsConfiguration. |
| 5    | <b>DialogFrameowner</b><br><br>Constructs an initially invisible, modeless Dialog with the specified owner Frame and an empty title.                                                                     |
| 6    | <b>DialogFrameowner, booleanmodal</b><br><br>Constructs an initially invisible Dialog with the specified owner Frame and modality and an empty title.                                                    |

|    |                                                                                                          |                                                                                                                      |
|----|----------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------|
| 7  | <b>Dialog</b> <i>Frameowner, Stringtitle</i>                                                             | Constructs an initially invisible, modeless Dialog with the specified owner Frame and title.                         |
| 8  | <b>Dialog</b> <i>Frameowner, Stringtitle, booleanmodal</i>                                               | Constructs an initially invisible Dialog with the specified owner Frame, title and modality.                         |
| 9  | <b>Dialog</b> <i>Frameowner, Stringtitle, booleanmodal, GraphicsConfigurationgc</i>                      | Constructs an initially invisible Dialog with the specified owner Frame, title, modality, and GraphicsConfiguration. |
| 10 | <b>Dialog</b> <i>Windowowner</i>                                                                         | Constructs an initially invisible, modeless Dialog with the specified owner Window and an empty title.               |
| 11 | <b>Dialog</b> <i>Windowowner, Dialog. ModalityTypemodalityType</i>                                       | Constructs an initially invisible Dialog with the specified owner Window and modality and an empty title.            |
| 12 | <b>Dialog</b> <i>Windowowner, Stringtitle</i>                                                            | Constructs an initially invisible, modeless Dialog with the specified owner Window and title.                        |
| 13 | <b>Dialog</b> <i>Windowowner, Stringtitle, Dialog. ModalityTypemodalityType</i>                          | Constructs an initially invisible Dialog with the specified owner Window, title and modality.                        |
| 14 | <b>Dialog</b> <i>Windowowner, Stringtitle, Dialog. ModalityTypemodalityType, GraphicsConfigurationgc</i> | Constructs an initially invisible Dialog with the specified owner Window, title, modality and GraphicsConfiguration  |

## Class methods

### S.N. Method & Description

|   |                       |                                                                             |
|---|-----------------------|-----------------------------------------------------------------------------|
| 1 | <b>void addNotify</b> | Makes this Dialog displayable by connecting it to a native screen resource. |
|---|-----------------------|-----------------------------------------------------------------------------|

2

### **AccessibleContext getAccessibleContext**

Gets the AccessibleContext associated with this Dialog.

3

### **Dialog.ModalityType getModalityType**

Returns the modality type of this dialog.

4

### **String getTitle**

Gets the title of the dialog.

5

### **void hide**

Deprecated. As of JDK version 1.5, replaced by setVisible*boolean*.

6

### **boolean isModal**

Indicates whether the dialog is modal.

7

### **boolean isResizable**

Indicates whether this dialog is resizable by the user.

8

### **boolean isUndecorated**

Indicates whether this dialog is undecorated.

9

### **protected String paramString**

Returns a string representing the state of this dialog.

10

### **void setModalbooleanmodal**

Specifies whether this dialog should be modal.

11

### **void setModalityTypeDialog. ModalityTypetype**

Sets the modality type for this dialog.

12

### **void setResizablebooleanresizable**

Sets whether this dialog is resizable by the user.

13

### **void setTitleStringtitle**

Sets the title of the Dialog.

- 14      **void setUndecorated***booleanundecorated*  
Disables or enables decorations for this dialog.
- 15      **void setVisible***booleanb*  
Shows or hides this Dialog depending on the value of parameter b.
- 16      **void show**  
Deprecated. As of JDK version 1.5, replaced by *setVisibleboolean*.
- 17      **void toBack**  
If this Window is visible, sends this Window to the back and may cause it to lose focus or activation if it is the focused or active Window.

## Methods inherited

This class inherits methods from the following classes:

- java.awt.Window
- java.awt.Component
- java.lang.Object

## Dialog Example

Create the following java program using any editor of your choice in say **D:/ > AWT > com > tutorialspoint > gui >**

*AwtControlDemo.java*

```
package com.tutorialspoint.gui;

import java.awt.*;
import java.awt.event.*;

public class AwtControlDemo {

    private Frame mainFrame;
    private Label headerLabel;
    private Label statusLabel;
    private Panel controlPanel;

    public AwtControlDemo(){
        prepareGUI();
    }

    public static void main(String[] args){
        AwtControlDemo awtControlDemo = new AwtControlDemo();
        awtControlDemo.showDialogDemo();
    }

    private void prepareGUI(){
        mainFrame = new Frame("Java AWT Examples");
        mainFrame.setSize(400,400);
        mainFrame.setLayout(new GridLayout(3, 1));
        mainFrame.addWindowListener(new WindowAdapter() {
```

```

        public void windowClosing(WindowEvent windowEvent){
            System.exit(0);
        }
    });
    headerLabel = new Label();
    headerLabel.setAlignment(Label.CENTER);
    statusLabel = new Label();
    statusLabel.setAlignment(Label.CENTER);
    statusLabel.setSize(350,100);

    controlPanel = new Panel();
    controlPanel.setLayout(new FlowLayout());

    mainFrame.add(headerLabel);
    mainFrame.add(controlPanel);
    mainFrame.add(statusLabel);
    mainFrame.setVisible(true);
}

private void showDialogDemo(){
    headerLabel.setText("Control in action: Dialog");
    Button showAboutDialogButton = new Button("Show About Dialog");
    showAboutDialogButton.addActionListener(new ActionListener() {
        @Override
        public void actionPerformed(ActionEvent e) {
            AboutDialog aboutDialog = new AboutDialog(mainFrame);
            aboutDialog.setVisible(true);
        }
    });

    controlPanel.add(showAboutDialogButton);
    mainFrame.setVisible(true);
}

class AboutDialog extends Dialog {
    public AboutDialog(Frame parent){
        super(parent, true);
        setBackground(Color.gray);
        setLayout(new BorderLayout());
        Panel panel = new Panel();
        panel.add(new Button("Close"));
        add("South", panel);
        setSize(200,200);

        addWindowListener(new WindowAdapter() {
            public void windowClosing(WindowEvent windowEvent){
                dispose();
            }
        });
    }

    public boolean action(Event evt, Object arg){
        if(arg.equals("Close")){
            dispose();
            return true;
        }
        return false;
    }

    public void paint(Graphics g){
        g.setColor(Color.white);
        g.drawString("TutorialsPoint.Com", 25,70 );
        g.drawString("Version 1.0", 60, 90);
    }
}
}

```

Compile the program using command prompt. Go to **D:/ > AWT** and type the following command.

```
D:\AWT>javac com\tutorialspoint\gui\AwtControlDemo.java
```

If no error comes that means compilation is successful. Run the program using following command.

```
D:\AWT>java com.tutorialspoint.gui.AwtControlDemo
```

Verify the following output



Loading [MathJax]/jax/output/HTML-CSS/jax.js