

Graphical User Interface using Swing

Part V - Dialogs

Samir Genaim

<http://docs.oracle.com/javase/tutorial/uiswing/>
<http://www.macs.hw.ac.uk/guidebook>

Dialog - showMessageDialog

frame can be **null** or a frame that already exists, e.g., the main window of the application that calls this method. The dialog belongs to this frame, which means that when the frame is minimised/destroyed/moved/etc so does the dialog. If **null** then the dialog is independent.



`JOptionPane.showMessageDialog(frame,
"Eggs are not supposed to be green.");`

All dialogs created by `JOptionPane.showMessageDialog` and such are modal, which means that all other windows in the program are blocked until this one is closed. We will see how to make non-modal dialogs later by inheriting directly from `JDialog`

see: `examples.swing.misc.DialogExample`

Dialog - showMessageDialog

```
JOptionPane.showMessageDialog(frame,  
    "Eggs are not supposed to be green.",  
    "Inane warning",  
    JOptionPane.WARNING_MESSAGE);
```

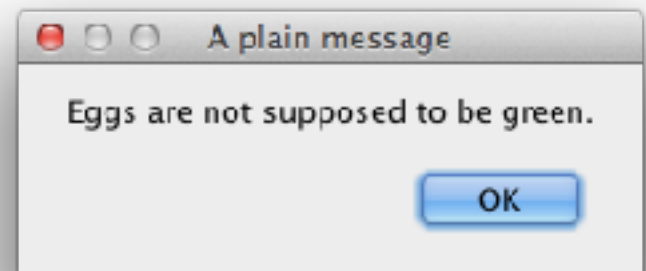


see: [examples.swing.misc.DialogExample](#)

Dialog - showMessageDialog

```
JOptionPane.showMessageDialog(frame,  
    "Eggs are not supposed to be green.",  
    "Inane warning",  
    JOptionPane.WARNING_MESSAGE);
```

JOptionPane.ERROR_MESSAGE
JOptionPane.PLAIN_MESSAGE

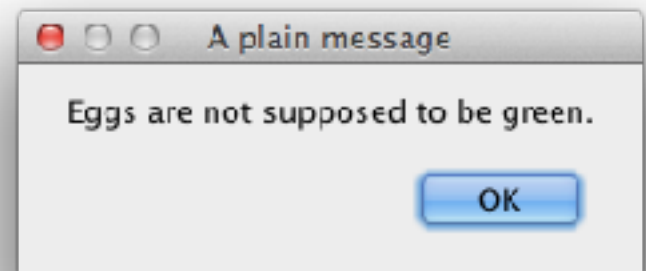


Dialog - showMessageDialog

```
JOptionPane.showMessageDialog(frame,  
    "Eggs are not supposed to be green.",  
    "Inane warning",  
    JOptionPane.WARNING_MESSAGE);
```

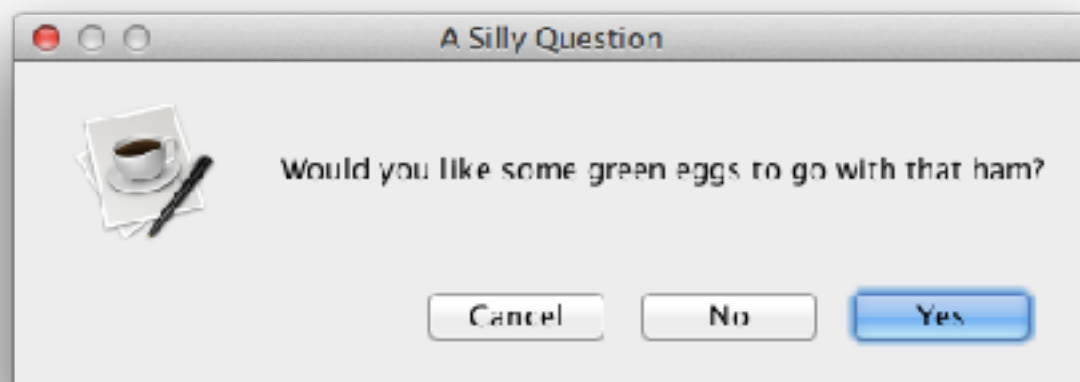
JOptionPane.ERROR_MESSAGE
JOptionPane.PLAIN_MESSAGE

```
ImageIcon icon = createImageIcon("calvin.jpg");  
JOptionPane.showMessageDialog(frame,  
    "Eggs are not supposed to be green.",  
    "Inane custom dialog",  
    JOptionPane.INFORMATION_MESSAGE,  
    icon);
```



Dialog - showOptionDialog

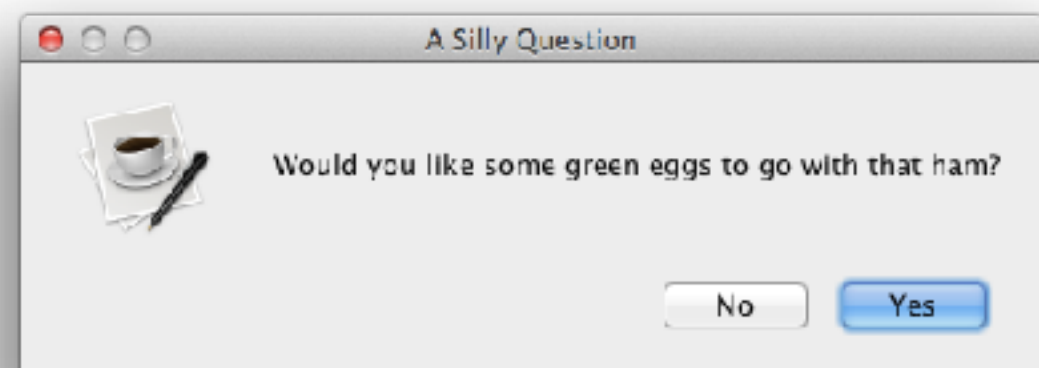
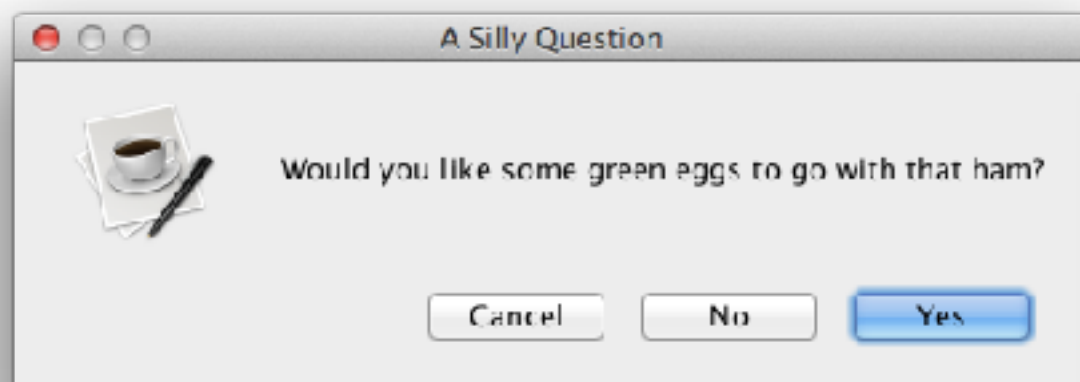
```
int n = JOptionPane.showOptionDialog(frame,  
    "Would you like some green eggs to go with that ham?",  
    "A Silly Question",  
    JOptionPane.YES_NO_CANCEL_OPTION,  
    JOptionPane.QUESTION_MESSAGE,  
    null,  
    null,  
    null);
```



Dialog - showOptionDialog

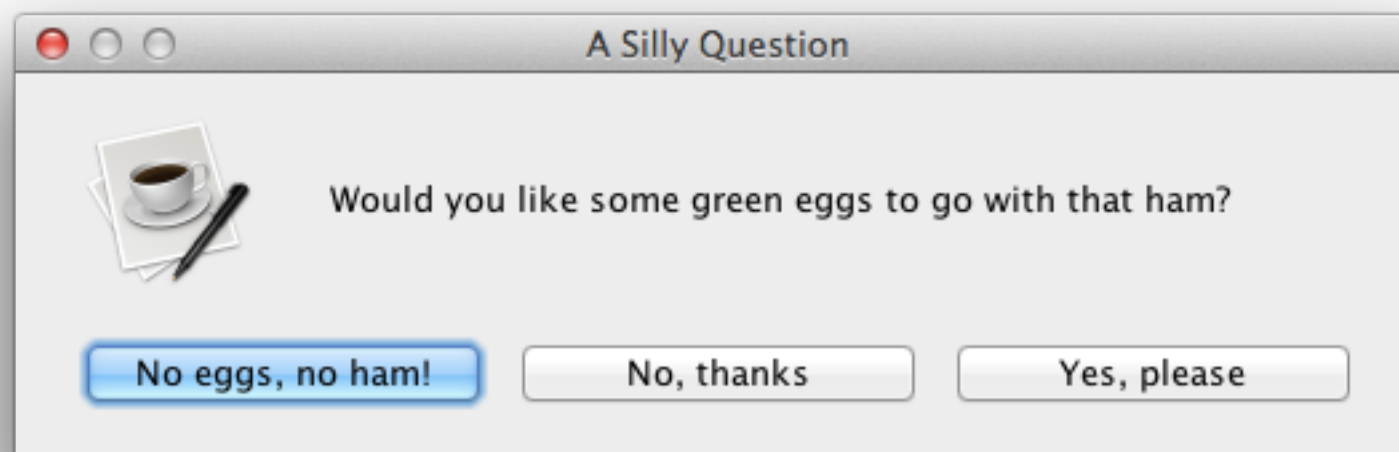
```
int n = JOptionPane.showOptionDialog(frame,  
    "Would you like some green eggs to go with that ham?",  
    "A Silly Question",  
    JOptionPane.YES_NO_CANCEL_OPTION,  
    JOptionPane.QUESTION_MESSAGE,  
    null,  
    null,  
    null);
```

`JOptionPane.YES_NO_OPTION,`



Dialog - showOptionDialog

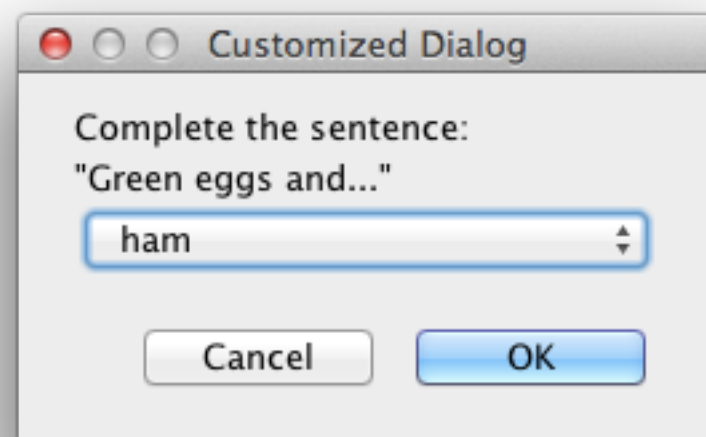
```
Object[] options1 = { "Yes, please", "No, thanks", "No eggs, no ham!" };  
int n = JOptionPane.showOptionDialog(frame,  
    "Would you like some green eggs to go with that ham?",  
    "A Silly Question",  
    JOptionPane.YES_NO_CANCEL_OPTION,  
    JOptionPane.QUESTION_MESSAGE,  
    null,  
    options,  
    options[2]);
```



see: [examples.swing.misc.DialogExample](#)

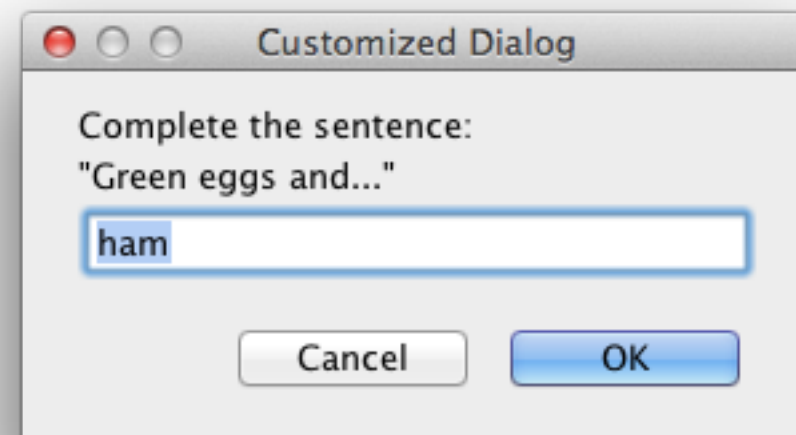
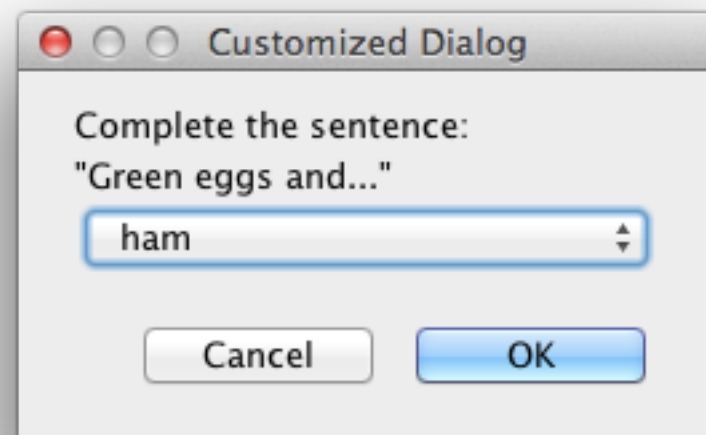
Dialog - showInputDialog

```
Object[] possibilities = { "ham", "spam", "yam" };  
String s = (String) JOptionPane.showInputDialog(frame,  
    "Complete the sentence:\n" + "\"Green eggs and...\"",  
    "Customized Dialog",  
    JOptionPane.PLAIN_MESSAGE,  
    null,  
    possibilities,  
    "ham"  
);
```



Dialog - showInputDialog

```
Object[] possibilities = { "ham", "spam", "yam" };  
String s = (String) JOptionPane.showInputDialog(frame,  
    "Complete the sentence:\n" + "\"Green eggs and...\"",  
    "Customized Dialog",  
    JOptionPane.PLAIN_MESSAGE,  
    null,  
    possibilities, null  
    "ham"  
);
```



Example: Confirm a Quit Request

```
private void initGUI() {  
    ...  
    JButton quit = new JButton("Quit");  
    quit.addActionListener(new ActionListener() {  
        public void actionPerformed(ActionEvent e) { quit(); }  
    });  
    mainPanel.add(quit);  
  
    ...  
    this.setDefaultCloseOperation(JFrame.DO_NOTHING_ON_CLOSE);  
    this.addWindowListener(new WindowListener() {  
        public void windowClosing(WindowEvent e) { quit(); }  
    });  
    ...  
}  
  
private void quit() {  
    int n = JOptionPane.showOptionDialog(this,  
        "Are sure you want to quit?", "Quit",  
        JOptionPane.YES_NO_OPTION, JOptionPane.QUESTION_MESSAGE, null,  
        null, null);  
  
    if (n == 0) {  
        // finalize your app  
        System.exit(0);  
    }  
}
```

Example: Confirm a Quit Request

```
private void initGUI() {
```

```
    ...  
    JButton quit = new JButton("Quit");  
    quit.addActionListener(new ActionListener() {  
        public void actionPerformed(ActionEvent e) { quit(); }  
    });  
    mainPanel.add(quit);  
  
    ...  
    this.setDefaultCloseOperation(JFrame.DO_NOTHING_ON_CLOSE);  
    this.addWindowListener(new WindowListener() {  
        public void windowClosing(WindowEvent e) { quit(); }  
    });  
    ...  
}
```

A Quit button that calls quit button when clicked

```
private void quit() {  
    int n = JOptionPane.showOptionDialog(this,  
        "Are sure you want to quit?", "Quit",  
        JOptionPane.YES_NO_OPTION, JOptionPane.QUESTION_MESSAGE, null,  
        null, null);  
  
    if (n == 0) {  
        // finalize your app  
        System.exit(0);  
    }  
}
```

Example: Confirm a Quit Request

```
private void initGUI() {
```

```
    ...  
    JButton quit = new JButton("Quit");  
    quit.addActionListener(new ActionListener() {  
        public void actionPerformed(ActionEvent e) { quit(); }  
    });  
    mainPanel.add(quit);
```

A Quit button that calls quit button when clicked

```
    ...  
    this.setDefaultCloseOperation(JFrame.DO_NOTHING_ON_CLOSE);  
    this.addWindowListener(new WindowListener() {  
        public void windowClosing(WindowEvent e) { quit(); }  
    });  
    ...
```

Call quit also when closing the window with a click on the x icon

```
private void quit() {  
    int n = JOptionPane.showOptionDialog(this,  
        "Are sure you want to quit?", "Quit",  
        JOptionPane.YES_NO_OPTION, JOptionPane.QUESTION_MESSAGE, null,  
        null, null);  
  
    if (n == 0) {  
        // finalize your app  
        System.exit(0);  
    }  
}
```

Dialog - FileChooser

```
public class FileChooserExample extends JFrame implements ActionListener {
```

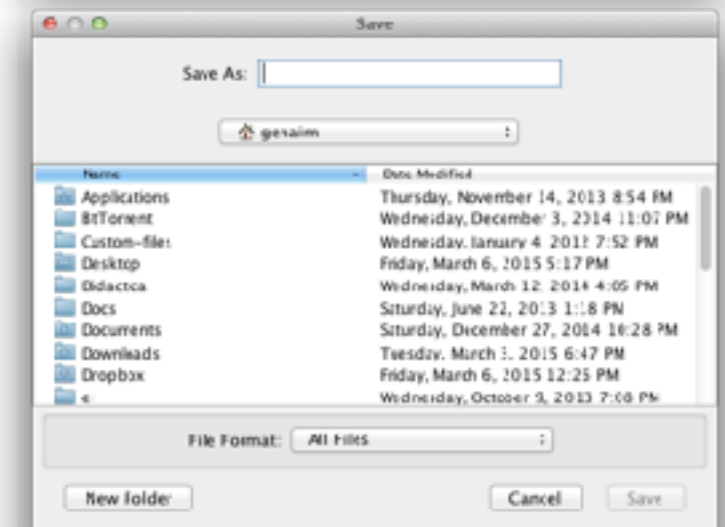
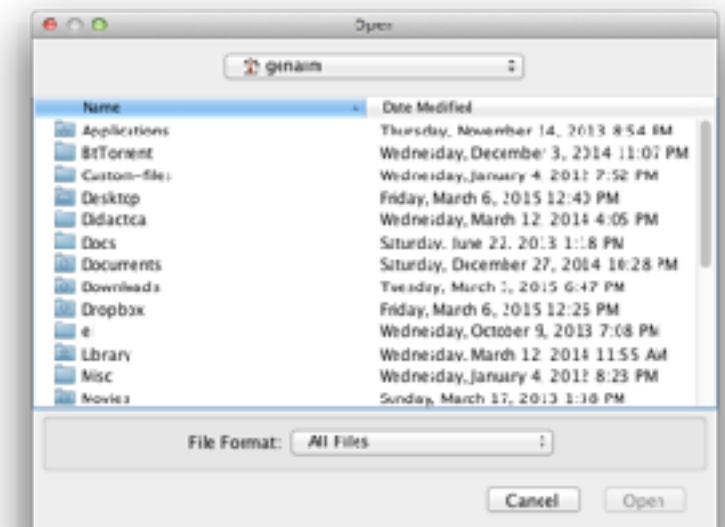
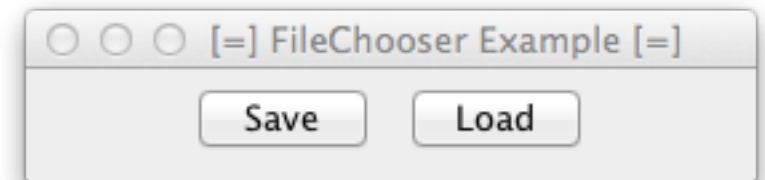
```
    JButton loadButton, saveButton;  
    JFileChooser fc;
```

```
    private void initGUI() {
```

```
        ...  
        fc = new JFileChooser();  
        ...  
    }
```

```
    public void actionPerformed(ActionEvent e) {  
        if (e.getSource() == loadButton) {  
            int returnVal = fc.showLoadDialog(this);  
            if (returnVal == JFileChooser.APPROVE_OPTION) {  
                File file = fc.getSelectedFile();  
                System.out.println("Loading: " + file.getName());  
            } else {  
                System.out.println("Load cancelled by user.");  
            }  
        } else if (e.getSource() == saveButton) {  
            int returnVal = fc.showSaveDialog(this);  
            ...  
        }  
    }
```

see: [examples.swing.misc.FileChooserExample](#)



Dialog - FileChooser

```
public class FileChooserExample extends JFrame implements ActionListener {
```

```
    JButton loadButton, saveButton;  
    JFileChooser fc;
```

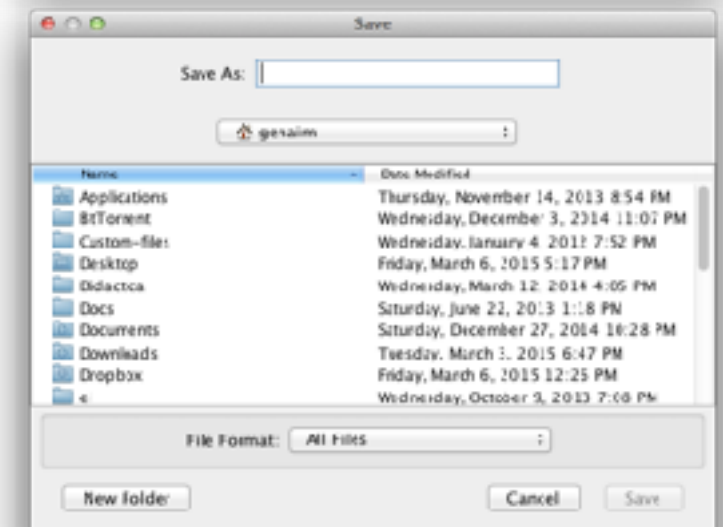
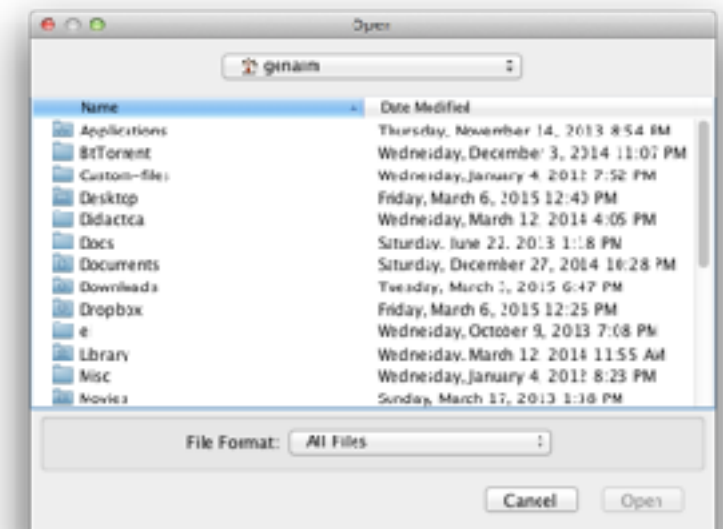
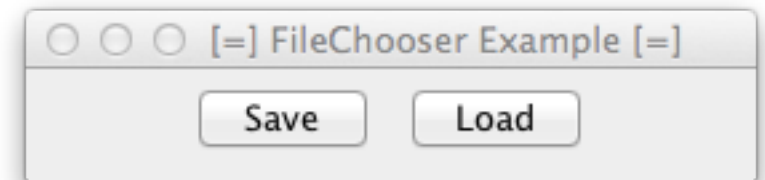
```
    private void initGUI() {
```

```
        ...  
        fc = new JFileChooser();
```

```
        ...  
    }
```

```
    public void actionPerformed(ActionEvent e) {  
        if (e.getSource() == loadButton) {  
            int returnVal = fc.showLoadDialog(this);  
            if (returnVal == JFileChooser.APPROVE_OPTION) {  
                File file = fc.getSelectedFile();  
                System.out.println("Loading: " + file.getName());  
            } else {  
                System.out.println("Load cancelled by user.");  
            }  
        } else if (e.getSource() == saveButton) {  
            int returnVal = fc.showSaveDialog(this);  
            ...  
        }  
    }
```

see: [examples.swing.misc.FileChooserExample](#)



Dialog - FileChooser

```
public class FileChooserExample extends JFrame implements ActionListener {
```

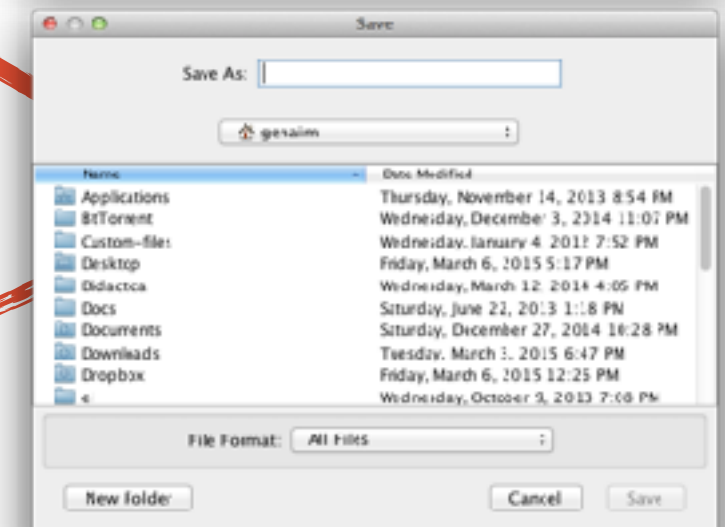
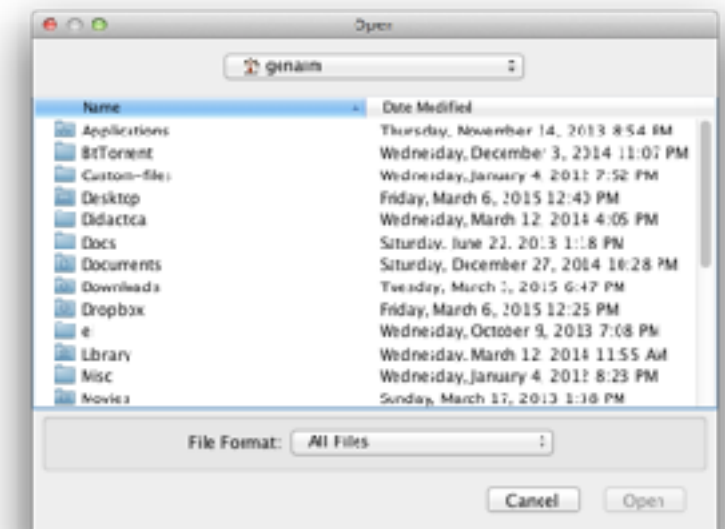
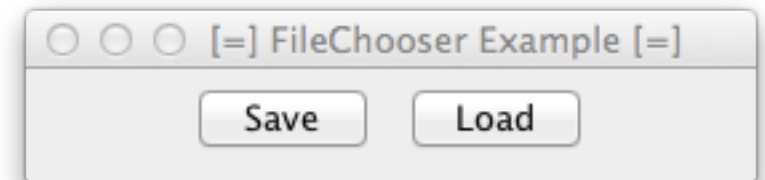
```
    JButton loadButton, saveButton;  
    JFileChooser fc;
```

```
    private void initGUI() {
```

```
        ...  
        fc = new JFileChooser();
```

```
        ...  
    }
```

```
    public void actionPerformed(ActionEvent e) {  
        if (e.getSource() == loadButton) {  
            int returnVal = fc.showLoadDialog(this);  
            if (returnVal == JFileChooser.APPROVE_OPTION) {  
                File file = fc.getSelectedFile();  
                System.out.println("Loading: " + file.getName());  
            } else {  
                System.out.println("Load cancelled by user.");  
            }  
        } else if (e.getSource() == saveButton) {  
            int returnVal = fc.showSaveDialog(this);  
            ...  
        }  
    }
```



see: [examples.swing.misc.FileChooserExample](#)

Dialog - FileChooser

```
public class FileChooserExample extends JFrame implements ActionListener {
```

```
    JButton loadButton, saveButton;  
    JFileChooser fc;
```

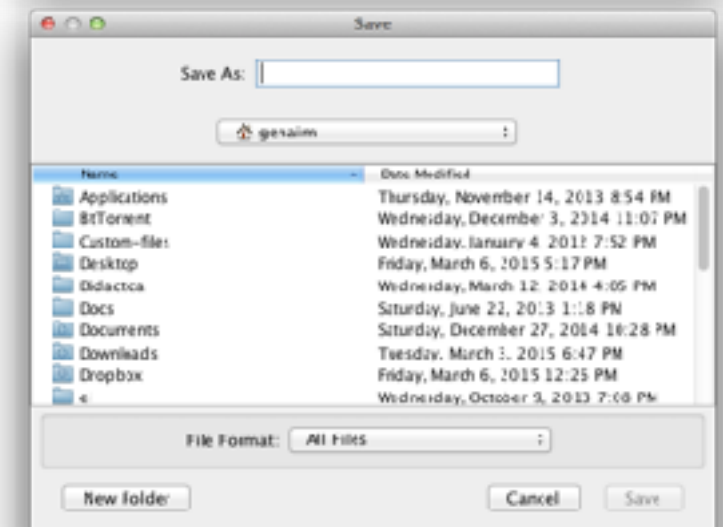
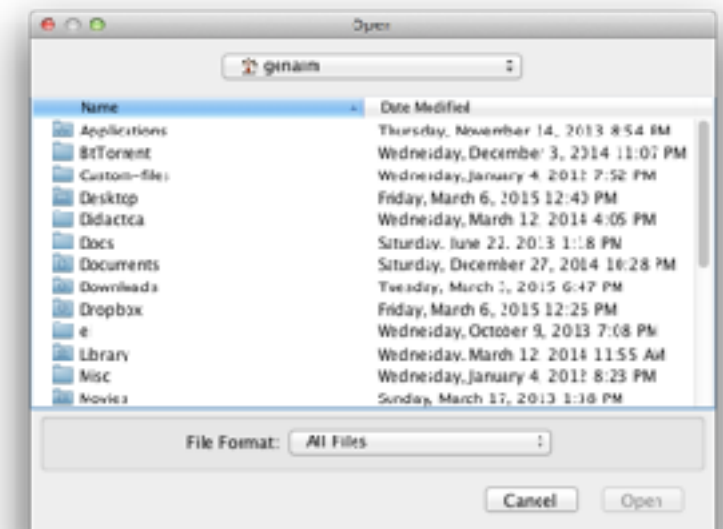
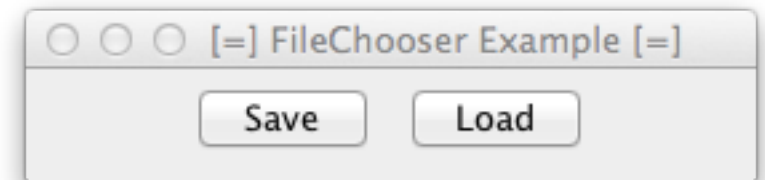
```
    private void initGUI() {
```

```
        ...  
        fc = new JFileChooser();
```

```
        ...  
    }
```

```
    public void actionPerformed(ActionEvent e) {  
        if (e.getSource() == loadButton) {  
            int returnVal = fc.showLoadDialog(this);  
            if (returnVal == JFileChooser.APPROVE_OPTION) {  
                File file = fc.getSelectedFile();  
                System.out.println("Loading: " + file.getName());  
            } else {  
                System.out.println("Load cancelled by user.");  
            }  
        } else if (e.getSource() == saveButton) {  
            int returnVal = fc.showSaveDialog(this);  
            ...  
        }  
    }
```

see: [examples.swing.misc.FileChooserExample](#)



Dialog - FileChooser

```
public class FileChooserExample extends JFrame implements ActionListener {
```

```
    JButton loadButton, saveButton  
    JFileChooser fc;
```

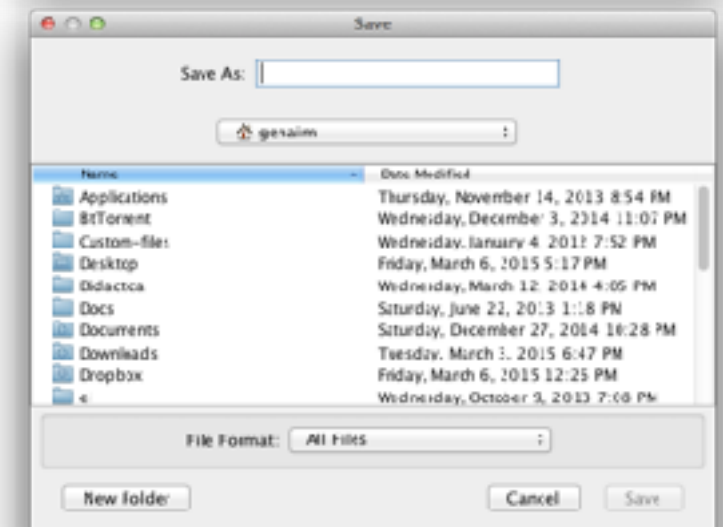
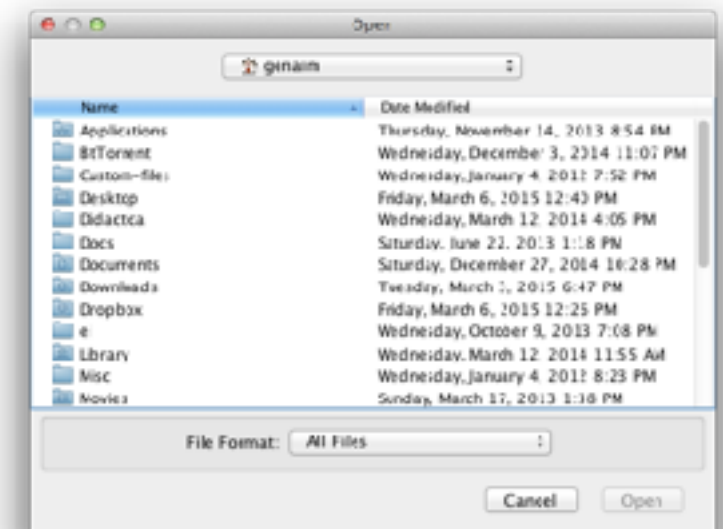
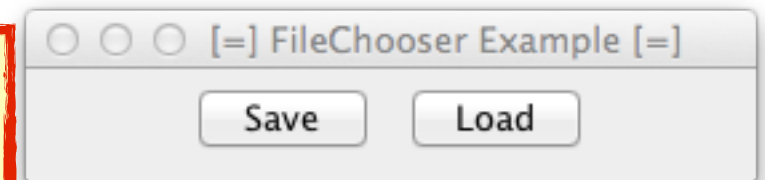
```
    private void initGUI() {
```

```
        ...  
        fc = new JFileChooser();
```

```
        ...  
    }
```

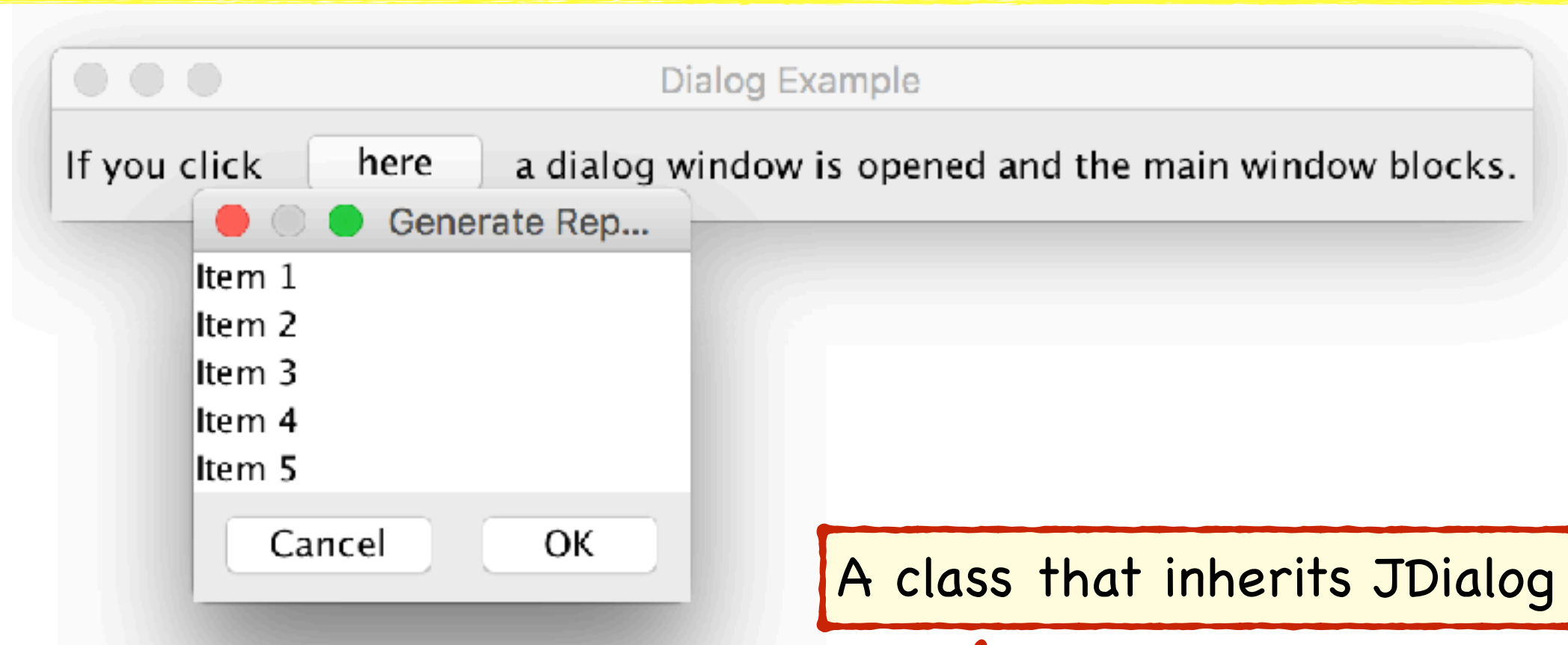
```
    public void actionPerformed(ActionEvent e) {  
        if (e.getSource() == loadButton) {  
            int returnVal = fc.showLoadDialog(this);  
            if (returnVal == JFileChooser.APPROVE_OPTION) {  
                File file = fc.getSelectedFile();  
                System.out.println("Loading: " + file.getName());  
            } else {  
                System.out.println("Load cancelled by user.");  
            }  
        } else if (e.getSource() == saveButton) {  
            int returnVal = fc.showSaveDialog(this);  
            ...  
        }  
    }
```

Declaring a JFileChooser as a **field** and creating it only **once**, it remembers in which directory it was every time we open the dialog, etc.



see: [examples.swing.misc.FileChooserExample](#)

Your Own Dialog!



see:

`examples.swing.misc.DialogWindow`

`examples.swing.misc.DialogWindowExample`

A class that inherits JFrame
and uses DialogWindow

A class that inherits JDialog