

AWT TEXTFIELD CLASS

http://www.tutorialspoint.com/awt/awt_textfield.htm

Copyright © tutorialspoint.com

Introduction

The textfield component allows the user to edit single line of text. When the user types a key in the text field the event is sent to the TextField. The key event may be key pressed, Key released or key typed. The key event is passed to the registered KeyListener. It is also possible to for an ActionEvent if the ActionEvent is enabled on the textfield then ActionEvent may be fired by pressing the return key.

Class declaration

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

```
public class TextField
    extends TextComponent
```

Class constructors

S.N.	Constructor & Description
1	TextField Constructs a new text field.
2	TextField(int columns) Constructs a new empty text field with the specified number of columns.
3	TextField(String text) Constructs a new text field initialized with the specified text.
4	TextField(String text, int columns) Constructs a new text field initialized with the specified text to be displayed, and wide enough to hold the specified number of columns.

Class methods

S.N.	Method & Description
1	void addActionListener(ActionListener l) Adds the specified action listener to receive action events from this text field.
2	void addNotify Creates the TextField's peer.

3

boolean echoCharIsSet

Indicates whether or not this text field has a character set for echoing.

4

AccessibleContext getAccessibleContext

Gets the AccessibleContext associated with this TextField.

5

ActionListener[] getActionListeners

Returns an array of all the action listeners registered on this textfield.

6

int getColumns

Gets the number of columns in this text field.

7

char getEchoChar

Gets the character that is to be used for echoing.

8

<T extends EventListener> T[] getListenersClass < T > listenerType

Returns an array of all the objects currently registered as FooListeners upon this TextField.

9

Dimension getMinimumSize

Gets the minumum dimensions for this text field.

10

Dimension getMinimumSize(int columns) Gets the minumum dimensions for a text field with the specified number of columns.

11

Dimension getPreferredSize

Gets the preferred size of this text field.

12

Dimension getPreferredSize(int columns)

Gets the preferred size of this text field with the specified number of columns.

13

Dimension minimumSize

Deprecated. As of JDK version 1.1, replaced by getMinimumSize.

14

Dimension minimumSize(int columns)

Deprecated. As of JDK version 1.1, replaced by `getMinimumSize`*int*.

- 15 **protected String paramString**
Returns a string representing the state of this TextField.
- 16 **Dimension preferredSize**
Deprecated. As of JDK version 1.1, replaced by `getPreferredSize`.
- 17 **Dimension preferredSize***int columns*
Deprecated. As of JDK version 1.1, replaced by `getPreferredSize`*int*.
- 18 **protected void processActionEvent***ActionEvent e*
Processes action events occurring on this text field by dispatching them to any registered `ActionListener` objects.
- 19 **protected void processEvent***AWTEvent e*
Processes events on this text field.
- 20 **void removeActionListener***ActionListener l*
Removes the specified action listener so that it no longer receives action events from this text field.
- 21 **void setColumns***int columns*
Sets the number of columns in this text field.
- 22 **void setEchoChar***char c*
Sets the echo character for this text field.
- 23 **void setEchoCharacter***char c*
Deprecated. As of JDK version 1.1, replaced by `setEchoChar`*char*.
- 24 **void setText***String t*
Sets the text that is presented by this text component to be the specified text.

Methods inherited

This class inherits methods from the following classes:

- java.awt.TextComponent
- java.awt.Component
- java.lang.Object

TextField 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.showTextFieldDemo();
    }

    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 showTextFieldDemo(){
        headerLabel.setText("Control in action: TextField");

        Label nameLabel= new Label("User ID: ", Label.RIGHT);
        Label passwordLabel = new Label("Password: ", Label.CENTER);
        final TextField userText = new TextField(6);
        final TextField passwordText = new TextField(6);
        passwordText.setEchoChar('*');

        Button loginButton = new Button("Login");

        loginButton.addActionListener(new ActionListener() {
```

```

        public void actionPerformed(ActionEvent e) {
            String data = "Username: " + userText.getText();
            data += ", Password: " + passwordText.getText();
            statusLabel.setText(data);
        }
    });

    controlPanel.add(namelabel);
    controlPanel.add(userText);
    controlPanel.add(passwordLabel);
    controlPanel.add(passwordText);
    controlPanel.add(loginButton);
    mainFrame.setVisible(true);
}
}

```

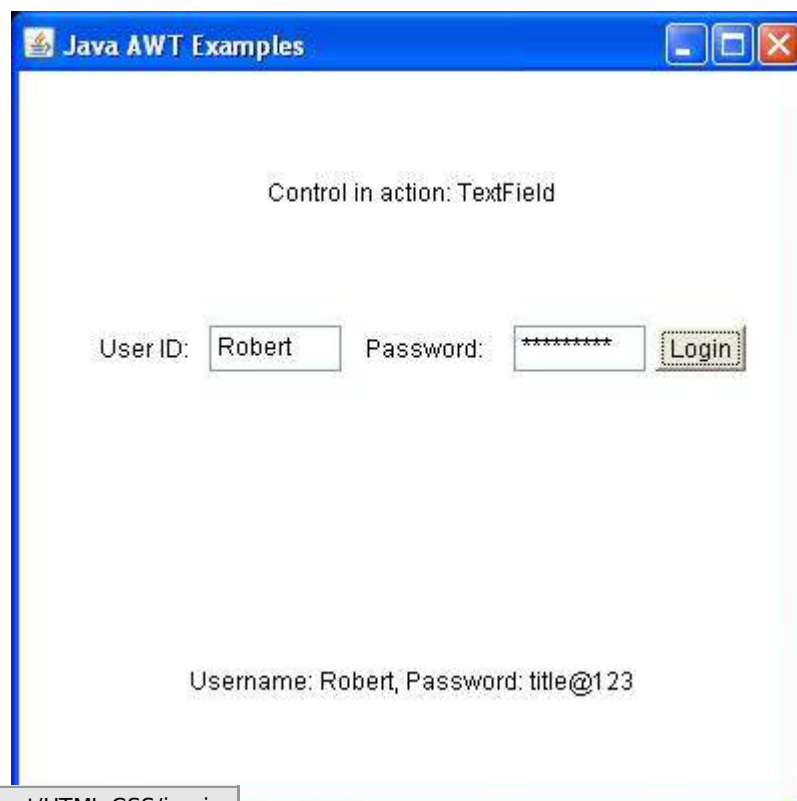
Compile the program using command prompt. Go to **D:/ > AWT** and type the following command.

```
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