

AWT TEXTAREA CLASS

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

Copyright © tutorialspoint.com

Introduction

The TextArea control in AWT provide us multiline editor area. The user can type here as much as he wants. When the text in the text area become larger than the viewable area the scroll bar is automatically appears which help us to scroll the text up & down and right & left.

Class declaration

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

```
public class TextArea
    extends TextComponent
```

Field

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

- **static int SCROLLBARS_BOTH** -- Create and display both vertical and horizontal scrollbars.
- **static int SCROLLBARS_HORIZONTAL_ONLY** -- Create and display horizontal scrollbar only.
- **static int SCROLLBARS_NONE** -- Do not create or display any scrollbars for the text area.
- **static int SCROLLBARS_VERTICAL_ONLY** -- Create and display vertical scrollbar only.

Class constructors

S.N.	Constructor & Description
1	TextArea Constructs a new text area with the empty string as text.
2	TextArea <i>introws, intcolumns</i> Constructs a new text area with the specified number of rows and columns and the empty string as text.
3	TextArea <i>Stringtext</i> Constructs a new text area with the specified text.
4	TextArea <i>Stringtext, introws, intcolumns</i> Constructs a new text area with the specified text, and with the specified number of rows and columns.
5	TextArea <i>Stringtext, introws, intcolumns, intscrollbars</i> Constructs a new text area with the specified text, and with the rows, columns, and scroll bar visibility as specified.

Class methods

S.N.	Method & Description
1	void addNotify Creates the TextArea's peer.
2	void appendStringstr Appends the given text to the text area's current text.
3	void appendTextStringstr Deprecated. As of JDK version 1.1, replaced by <i>appendString</i> .
4	AccessibleContext getAccessibleContext Returns the AccessibleContext associated with this TextArea.
5	int getColumns Returns the number of columns in this text area.
6	Dimension getMinimumSize Determines the minimum size of this text area.
7	Dimension getMinimumSizeintrows, intcolumns Determines the minimum size of a text area with the specified number of rows and columns.
8	Dimension getPreferredSize Determines the preferred size of this text area.
9	Dimension getPreferredSizeintrows, intcolumns Determines the preferred size of a text area with the specified number of rows and columns.
10	int getRows Returns the number of rows in the text area.

11

int getScrollbarVisibility

Returns an enumerated value that indicates which scroll bars the text area uses.

12

void insertStringstr, intpos

Inserts the specified text at the specified position in this text area.

13

void insertTextStringstr, intpos

Deprecated. As of JDK version 1.1, replaced by insertString, int.

14

Dimension minimumSize

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

15

Dimension minimumSizeintrows, intcolumns

Deprecated. As of JDK version 1.1, replaced by getMinimumSizeint, int.

16

protected String paramString

Returns a string representing the state of this TextArea.

17

Dimension preferredSize

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

18

Dimension preferredSizeintrows, intcolumns

Deprecated. As of JDK version 1.1, replaced by getPreferredSizeint, int.

19

void replaceRangeStringstr, intstart, intend

Replaces text between the indicated start and end positions with the specified replacement text.

20

void replaceTextStringstr, intstart, intend

Deprecated. As of JDK version 1.1, replaced by replaceRangeString, int, int.

21

void setColumnsintcolumns

Sets the number of columns for this text area.

22

void setRowsintrows

Sets the number of rows for this text area.

Methods inherited

This class inherits methods from the following classes:

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

TextArea 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.showTextAreaDemo();
    }

    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 showTextAreaDemo(){
        headerLabel.setText("Control in action: TextArea");

        Label commentLabel= new Label("Comments: ", Label.RIGHT);
```

```

final TextArea commentTextArea = new TextArea("This is a AWT tutorial "
+"to make GUI application in Java.",5,30);

Button showButton = new Button("Show");

showButton.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        statusLabel.setText( commentTextArea.getText());
    }
});

controlPanel.add(commentLabel);
controlPanel.add(commentTextArea);
controlPanel.add(showButton);
mainFrame.setVisible(true);
}
}

```

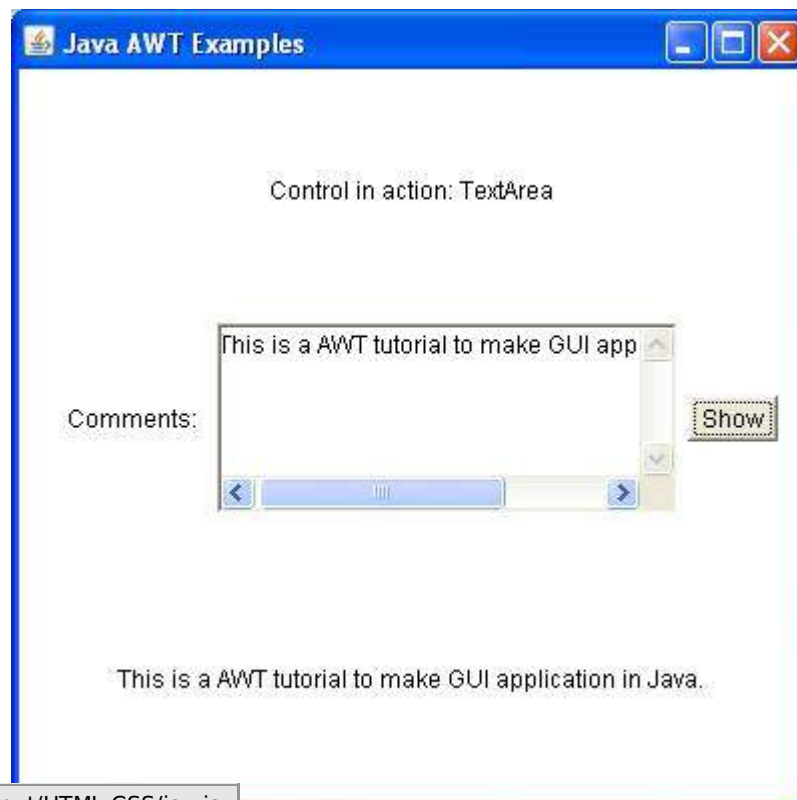
Compile the program using command prompt. Go to **D:/ > AWT** and type the following command.

```
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