| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/PasswordView.html) | [**Tree**](http://docs.google.com/package-tree.html) | [**Deprecated**](http://docs.google.com/deprecated-list.html) | [**Index**](http://docs.google.com/index-files/index-1.html) | [**Help**](http://docs.google.com/help-doc.html) | | --- | --- | --- | --- | --- | --- | --- | --- | | | ***Java™ Platform***  ***Standard Ed. 6*** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| [**PREV CLASS**](http://docs.google.com/javax/swing/text/ParagraphView.html)   [**NEXT CLASS**](http://docs.google.com/javax/swing/text/PlainDocument.html) | [**FRAMES**](http://docs.google.com/index.html?javax/swing/text/PasswordView.html)    [**NO FRAMES**](http://docs.google.com/PasswordView.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | [FIELD](#2et92p0) | [CONSTR](#1t3h5sf) | [METHOD](#4d34og8) | DETAIL: FIELD | [CONSTR](#lnxbz9) | [METHOD](#1ksv4uv) |

## **javax.swing.text**

Class PasswordView

[java.lang.Object](http://docs.google.com/java/lang/Object.html)  
 [javax.swing.text.View](http://docs.google.com/javax/swing/text/View.html)  
 [javax.swing.text.PlainView](http://docs.google.com/javax/swing/text/PlainView.html)  
 [javax.swing.text.FieldView](http://docs.google.com/javax/swing/text/FieldView.html)  
 **javax.swing.text.PasswordView**

**All Implemented Interfaces:** [SwingConstants](http://docs.google.com/javax/swing/SwingConstants.html), [TabExpander](http://docs.google.com/javax/swing/text/TabExpander.html)

public class **PasswordView**extends [FieldView](http://docs.google.com/javax/swing/text/FieldView.html)

Implements a View suitable for use in JPasswordField UI implementations. This is basically a field ui that renders its contents as the echo character specified in the associated component (if it can narrow the component to a JPasswordField).

**See Also:**[View](http://docs.google.com/javax/swing/text/View.html)

| **Field Summary** | |
| --- | --- |

| **Fields inherited from class javax.swing.text.**[**PlainView**](http://docs.google.com/javax/swing/text/PlainView.html) |
| --- |
| [metrics](http://docs.google.com/javax/swing/text/PlainView.html#metrics) |

| **Fields inherited from class javax.swing.text.**[**View**](http://docs.google.com/javax/swing/text/View.html) |
| --- |
| [BadBreakWeight](http://docs.google.com/javax/swing/text/View.html#BadBreakWeight), [ExcellentBreakWeight](http://docs.google.com/javax/swing/text/View.html#ExcellentBreakWeight), [ForcedBreakWeight](http://docs.google.com/javax/swing/text/View.html#ForcedBreakWeight), [GoodBreakWeight](http://docs.google.com/javax/swing/text/View.html#GoodBreakWeight), [X\_AXIS](http://docs.google.com/javax/swing/text/View.html#X_AXIS), [Y\_AXIS](http://docs.google.com/javax/swing/text/View.html#Y_AXIS) |

| **Fields inherited from interface javax.swing.**[**SwingConstants**](http://docs.google.com/javax/swing/SwingConstants.html) |
| --- |
| [BOTTOM](http://docs.google.com/javax/swing/SwingConstants.html#BOTTOM), [CENTER](http://docs.google.com/javax/swing/SwingConstants.html#CENTER), [EAST](http://docs.google.com/javax/swing/SwingConstants.html#EAST), [HORIZONTAL](http://docs.google.com/javax/swing/SwingConstants.html#HORIZONTAL), [LEADING](http://docs.google.com/javax/swing/SwingConstants.html#LEADING), [LEFT](http://docs.google.com/javax/swing/SwingConstants.html#LEFT), [NEXT](http://docs.google.com/javax/swing/SwingConstants.html#NEXT), [NORTH](http://docs.google.com/javax/swing/SwingConstants.html#NORTH), [NORTH\_EAST](http://docs.google.com/javax/swing/SwingConstants.html#NORTH_EAST), [NORTH\_WEST](http://docs.google.com/javax/swing/SwingConstants.html#NORTH_WEST), [PREVIOUS](http://docs.google.com/javax/swing/SwingConstants.html#PREVIOUS), [RIGHT](http://docs.google.com/javax/swing/SwingConstants.html#RIGHT), [SOUTH](http://docs.google.com/javax/swing/SwingConstants.html#SOUTH), [SOUTH\_EAST](http://docs.google.com/javax/swing/SwingConstants.html#SOUTH_EAST), [SOUTH\_WEST](http://docs.google.com/javax/swing/SwingConstants.html#SOUTH_WEST), [TOP](http://docs.google.com/javax/swing/SwingConstants.html#TOP), [TRAILING](http://docs.google.com/javax/swing/SwingConstants.html#TRAILING), [VERTICAL](http://docs.google.com/javax/swing/SwingConstants.html#VERTICAL), [WEST](http://docs.google.com/javax/swing/SwingConstants.html#WEST) |

| **Constructor Summary** | |
| --- | --- |
| [**PasswordView**](http://docs.google.com/javax/swing/text/PasswordView.html#PasswordView(javax.swing.text.Element))([Element](http://docs.google.com/javax/swing/text/Element.html) elem)            Constructs a new view wrapped on an element. |

| **Method Summary** | |
| --- | --- |
| protected  int | [**drawEchoCharacter**](http://docs.google.com/javax/swing/text/PasswordView.html#drawEchoCharacter(java.awt.Graphics,%20int,%20int,%20char))([Graphics](http://docs.google.com/java/awt/Graphics.html) g, int x, int y, char c)            Renders the echo character, or whatever graphic should be used to display the password characters. |
| protected  int | [**drawSelectedText**](http://docs.google.com/javax/swing/text/PasswordView.html#drawSelectedText(java.awt.Graphics,%20int,%20int,%20int,%20int))([Graphics](http://docs.google.com/java/awt/Graphics.html) g, int x, int y, int p0, int p1)            Renders the given range in the model as selected text. |
| protected  int | [**drawUnselectedText**](http://docs.google.com/javax/swing/text/PasswordView.html#drawUnselectedText(java.awt.Graphics,%20int,%20int,%20int,%20int))([Graphics](http://docs.google.com/java/awt/Graphics.html) g, int x, int y, int p0, int p1)            Renders the given range in the model as normal unselected text. |
| float | [**getPreferredSpan**](http://docs.google.com/javax/swing/text/PasswordView.html#getPreferredSpan(int))(int axis)            Determines the preferred span for this view along an axis. |
| [Shape](http://docs.google.com/java/awt/Shape.html) | [**modelToView**](http://docs.google.com/javax/swing/text/PasswordView.html#modelToView(int,%20java.awt.Shape,%20javax.swing.text.Position.Bias))(int pos, [Shape](http://docs.google.com/java/awt/Shape.html) a, [Position.Bias](http://docs.google.com/javax/swing/text/Position.Bias.html) b)            Provides a mapping from the document model coordinate space to the coordinate space of the view mapped to it. |
| int | [**viewToModel**](http://docs.google.com/javax/swing/text/PasswordView.html#viewToModel(float,%20float,%20java.awt.Shape,%20javax.swing.text.Position.Bias%5B%5D))(float fx, float fy, [Shape](http://docs.google.com/java/awt/Shape.html) a, [Position.Bias](http://docs.google.com/javax/swing/text/Position.Bias.html)[] bias)            Provides a mapping from the view coordinate space to the logical coordinate space of the model. |

| **Methods inherited from class javax.swing.text.**[**FieldView**](http://docs.google.com/javax/swing/text/FieldView.html) |
| --- |
| [adjustAllocation](http://docs.google.com/javax/swing/text/FieldView.html#adjustAllocation(java.awt.Shape)), [getFontMetrics](http://docs.google.com/javax/swing/text/FieldView.html#getFontMetrics()), [getResizeWeight](http://docs.google.com/javax/swing/text/FieldView.html#getResizeWeight(int)), [insertUpdate](http://docs.google.com/javax/swing/text/FieldView.html#insertUpdate(javax.swing.event.DocumentEvent,%20java.awt.Shape,%20javax.swing.text.ViewFactory)), [paint](http://docs.google.com/javax/swing/text/FieldView.html#paint(java.awt.Graphics,%20java.awt.Shape)), [removeUpdate](http://docs.google.com/javax/swing/text/FieldView.html#removeUpdate(javax.swing.event.DocumentEvent,%20java.awt.Shape,%20javax.swing.text.ViewFactory)) |

| **Methods inherited from class javax.swing.text.**[**PlainView**](http://docs.google.com/javax/swing/text/PlainView.html) |
| --- |
| [changedUpdate](http://docs.google.com/javax/swing/text/PlainView.html#changedUpdate(javax.swing.event.DocumentEvent,%20java.awt.Shape,%20javax.swing.text.ViewFactory)), [damageLineRange](http://docs.google.com/javax/swing/text/PlainView.html#damageLineRange(int,%20int,%20java.awt.Shape,%20java.awt.Component)), [drawLine](http://docs.google.com/javax/swing/text/PlainView.html#drawLine(int,%20java.awt.Graphics,%20int,%20int)), [getLineBuffer](http://docs.google.com/javax/swing/text/PlainView.html#getLineBuffer()), [getTabSize](http://docs.google.com/javax/swing/text/PlainView.html#getTabSize()), [lineToRect](http://docs.google.com/javax/swing/text/PlainView.html#lineToRect(java.awt.Shape,%20int)), [nextTabStop](http://docs.google.com/javax/swing/text/PlainView.html#nextTabStop(float,%20int)), [setSize](http://docs.google.com/javax/swing/text/PlainView.html#setSize(float,%20float)), [updateDamage](http://docs.google.com/javax/swing/text/PlainView.html#updateDamage(javax.swing.event.DocumentEvent,%20java.awt.Shape,%20javax.swing.text.ViewFactory)), [updateMetrics](http://docs.google.com/javax/swing/text/PlainView.html#updateMetrics()) |

| **Methods inherited from class javax.swing.text.**[**View**](http://docs.google.com/javax/swing/text/View.html) |
| --- |
| [append](http://docs.google.com/javax/swing/text/View.html#append(javax.swing.text.View)), [breakView](http://docs.google.com/javax/swing/text/View.html#breakView(int,%20int,%20float,%20float)), [createFragment](http://docs.google.com/javax/swing/text/View.html#createFragment(int,%20int)), [forwardUpdate](http://docs.google.com/javax/swing/text/View.html#forwardUpdate(javax.swing.event.DocumentEvent.ElementChange,%20javax.swing.event.DocumentEvent,%20java.awt.Shape,%20javax.swing.text.ViewFactory)), [forwardUpdateToView](http://docs.google.com/javax/swing/text/View.html#forwardUpdateToView(javax.swing.text.View,%20javax.swing.event.DocumentEvent,%20java.awt.Shape,%20javax.swing.text.ViewFactory)), [getAlignment](http://docs.google.com/javax/swing/text/View.html#getAlignment(int)), [getAttributes](http://docs.google.com/javax/swing/text/View.html#getAttributes()), [getBreakWeight](http://docs.google.com/javax/swing/text/View.html#getBreakWeight(int,%20float,%20float)), [getChildAllocation](http://docs.google.com/javax/swing/text/View.html#getChildAllocation(int,%20java.awt.Shape)), [getContainer](http://docs.google.com/javax/swing/text/View.html#getContainer()), [getDocument](http://docs.google.com/javax/swing/text/View.html#getDocument()), [getElement](http://docs.google.com/javax/swing/text/View.html#getElement()), [getEndOffset](http://docs.google.com/javax/swing/text/View.html#getEndOffset()), [getGraphics](http://docs.google.com/javax/swing/text/View.html#getGraphics()), [getMaximumSpan](http://docs.google.com/javax/swing/text/View.html#getMaximumSpan(int)), [getMinimumSpan](http://docs.google.com/javax/swing/text/View.html#getMinimumSpan(int)), [getNextVisualPositionFrom](http://docs.google.com/javax/swing/text/View.html#getNextVisualPositionFrom(int,%20javax.swing.text.Position.Bias,%20java.awt.Shape,%20int,%20javax.swing.text.Position.Bias%5B%5D)), [getParent](http://docs.google.com/javax/swing/text/View.html#getParent()), [getStartOffset](http://docs.google.com/javax/swing/text/View.html#getStartOffset()), [getToolTipText](http://docs.google.com/javax/swing/text/View.html#getToolTipText(float,%20float,%20java.awt.Shape)), [getView](http://docs.google.com/javax/swing/text/View.html#getView(int)), [getViewCount](http://docs.google.com/javax/swing/text/View.html#getViewCount()), [getViewFactory](http://docs.google.com/javax/swing/text/View.html#getViewFactory()), [getViewIndex](http://docs.google.com/javax/swing/text/View.html#getViewIndex(float,%20float,%20java.awt.Shape)), [getViewIndex](http://docs.google.com/javax/swing/text/View.html#getViewIndex(int,%20javax.swing.text.Position.Bias)), [insert](http://docs.google.com/javax/swing/text/View.html#insert(int,%20javax.swing.text.View)), [isVisible](http://docs.google.com/javax/swing/text/View.html#isVisible()), [modelToView](http://docs.google.com/javax/swing/text/View.html#modelToView(int,%20javax.swing.text.Position.Bias,%20int,%20javax.swing.text.Position.Bias,%20java.awt.Shape)), [modelToView](http://docs.google.com/javax/swing/text/View.html#modelToView(int,%20java.awt.Shape)), [preferenceChanged](http://docs.google.com/javax/swing/text/View.html#preferenceChanged(javax.swing.text.View,%20boolean,%20boolean)), [remove](http://docs.google.com/javax/swing/text/View.html#remove(int)), [removeAll](http://docs.google.com/javax/swing/text/View.html#removeAll()), [replace](http://docs.google.com/javax/swing/text/View.html#replace(int,%20int,%20javax.swing.text.View%5B%5D)), [setParent](http://docs.google.com/javax/swing/text/View.html#setParent(javax.swing.text.View)), [updateChildren](http://docs.google.com/javax/swing/text/View.html#updateChildren(javax.swing.event.DocumentEvent.ElementChange,%20javax.swing.event.DocumentEvent,%20javax.swing.text.ViewFactory)), [updateLayout](http://docs.google.com/javax/swing/text/View.html#updateLayout(javax.swing.event.DocumentEvent.ElementChange,%20javax.swing.event.DocumentEvent,%20java.awt.Shape)), [viewToModel](http://docs.google.com/javax/swing/text/View.html#viewToModel(float,%20float,%20java.awt.Shape)) |

| **Methods inherited from class java.lang.**[**Object**](http://docs.google.com/java/lang/Object.html) |
| --- |
| [clone](http://docs.google.com/java/lang/Object.html#clone()), [equals](http://docs.google.com/java/lang/Object.html#equals(java.lang.Object)), [finalize](http://docs.google.com/java/lang/Object.html#finalize()), [getClass](http://docs.google.com/java/lang/Object.html#getClass()), [hashCode](http://docs.google.com/java/lang/Object.html#hashCode()), [notify](http://docs.google.com/java/lang/Object.html#notify()), [notifyAll](http://docs.google.com/java/lang/Object.html#notifyAll()), [toString](http://docs.google.com/java/lang/Object.html#toString()), [wait](http://docs.google.com/java/lang/Object.html#wait()), [wait](http://docs.google.com/java/lang/Object.html#wait(long)), [wait](http://docs.google.com/java/lang/Object.html#wait(long,%20int)) |

| **Constructor Detail** |
| --- |

### PasswordView

public **PasswordView**([Element](http://docs.google.com/javax/swing/text/Element.html) elem)

Constructs a new view wrapped on an element.

**Parameters:**elem - the element

| **Method Detail** |
| --- |

### drawUnselectedText

protected int **drawUnselectedText**([Graphics](http://docs.google.com/java/awt/Graphics.html) g,  
 int x,  
 int y,  
 int p0,  
 int p1)  
 throws [BadLocationException](http://docs.google.com/javax/swing/text/BadLocationException.html)

Renders the given range in the model as normal unselected text. This sets the foreground color and echos the characters using the value returned by getEchoChar().

**Overrides:**[drawUnselectedText](http://docs.google.com/javax/swing/text/PlainView.html#drawUnselectedText(java.awt.Graphics,%20int,%20int,%20int,%20int)) in class [PlainView](http://docs.google.com/javax/swing/text/PlainView.html) **Parameters:**g - the graphics contextx - the starting X coordinate >= 0y - the starting Y coordinate >= 0p0 - the starting offset in the model >= 0p1 - the ending offset in the model >= p0 **Returns:**the X location of the end of the range >= 0 **Throws:** [BadLocationException](http://docs.google.com/javax/swing/text/BadLocationException.html) - if p0 or p1 are out of range

### drawSelectedText

protected int **drawSelectedText**([Graphics](http://docs.google.com/java/awt/Graphics.html) g,  
 int x,  
 int y,  
 int p0,  
 int p1)  
 throws [BadLocationException](http://docs.google.com/javax/swing/text/BadLocationException.html)

Renders the given range in the model as selected text. This is implemented to render the text in the color specified in the hosting component. It assumes the highlighter will render the selected background. Uses the result of getEchoChar() to display the characters.

**Overrides:**[drawSelectedText](http://docs.google.com/javax/swing/text/PlainView.html#drawSelectedText(java.awt.Graphics,%20int,%20int,%20int,%20int)) in class [PlainView](http://docs.google.com/javax/swing/text/PlainView.html) **Parameters:**g - the graphics contextx - the starting X coordinate >= 0y - the starting Y coordinate >= 0p0 - the starting offset in the model >= 0p1 - the ending offset in the model >= p0 **Returns:**the X location of the end of the range >= 0 **Throws:** [BadLocationException](http://docs.google.com/javax/swing/text/BadLocationException.html) - if p0 or p1 are out of range

### drawEchoCharacter

protected int **drawEchoCharacter**([Graphics](http://docs.google.com/java/awt/Graphics.html) g,  
 int x,  
 int y,  
 char c)

Renders the echo character, or whatever graphic should be used to display the password characters. The color in the Graphics object is set to the appropriate foreground color for selected or unselected text.

**Parameters:**g - the graphics contextx - the starting X coordinate >= 0y - the starting Y coordinate >= 0c - the echo character **Returns:**the updated X position >= 0

### modelToView

public [Shape](http://docs.google.com/java/awt/Shape.html) **modelToView**(int pos,  
 [Shape](http://docs.google.com/java/awt/Shape.html) a,  
 [Position.Bias](http://docs.google.com/javax/swing/text/Position.Bias.html) b)  
 throws [BadLocationException](http://docs.google.com/javax/swing/text/BadLocationException.html)

Provides a mapping from the document model coordinate space to the coordinate space of the view mapped to it.

**Overrides:**[modelToView](http://docs.google.com/javax/swing/text/FieldView.html#modelToView(int,%20java.awt.Shape,%20javax.swing.text.Position.Bias)) in class [FieldView](http://docs.google.com/javax/swing/text/FieldView.html) **Parameters:**pos - the position to convert >= 0a - the allocated region to render intob - the bias toward the previous character or the next character represented by the offset, in case the position is a boundary of two views; b will have one of these values:

* Position.Bias.Forward
* Position.Bias.Backward

**Returns:**the bounding box of the given position **Throws:** [BadLocationException](http://docs.google.com/javax/swing/text/BadLocationException.html) - if the given position does not represent a valid location in the associated document**See Also:**[View.modelToView(int, java.awt.Shape, javax.swing.text.Position.Bias)](http://docs.google.com/javax/swing/text/View.html#modelToView(int,%20java.awt.Shape,%20javax.swing.text.Position.Bias))

### viewToModel

public int **viewToModel**(float fx,  
 float fy,  
 [Shape](http://docs.google.com/java/awt/Shape.html) a,  
 [Position.Bias](http://docs.google.com/javax/swing/text/Position.Bias.html)[] bias)

Provides a mapping from the view coordinate space to the logical coordinate space of the model.

**Overrides:**[viewToModel](http://docs.google.com/javax/swing/text/FieldView.html#viewToModel(float,%20float,%20java.awt.Shape,%20javax.swing.text.Position.Bias%5B%5D)) in class [FieldView](http://docs.google.com/javax/swing/text/FieldView.html) **Parameters:**fx - the X coordinate >= 0.0ffy - the Y coordinate >= 0.0fa - the allocated region to render into **Returns:**the location within the model that best represents the given point in the view**See Also:**[View.viewToModel(float, float, java.awt.Shape, javax.swing.text.Position.Bias[])](http://docs.google.com/javax/swing/text/View.html#viewToModel(float,%20float,%20java.awt.Shape,%20javax.swing.text.Position.Bias%5B%5D))

### getPreferredSpan

public float **getPreferredSpan**(int axis)

Determines the preferred span for this view along an axis.

**Overrides:**[getPreferredSpan](http://docs.google.com/javax/swing/text/FieldView.html#getPreferredSpan(int)) in class [FieldView](http://docs.google.com/javax/swing/text/FieldView.html) **Parameters:**axis - may be either View.X\_AXIS or View.Y\_AXIS **Returns:**the span the view would like to be rendered into >= 0. Typically the view is told to render into the span that is returned, although there is no guarantee. The parent may choose to resize or break the view.**See Also:**[View.getPreferredSpan(int)](http://docs.google.com/javax/swing/text/View.html#getPreferredSpan(int))

| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/PasswordView.html) | [**Tree**](http://docs.google.com/package-tree.html) | [**Deprecated**](http://docs.google.com/deprecated-list.html) | [**Index**](http://docs.google.com/index-files/index-1.html) | [**Help**](http://docs.google.com/help-doc.html) | | --- | --- | --- | --- | --- | --- | --- | --- | | | ***Java™ Platform***  ***Standard Ed. 6*** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| [**PREV CLASS**](http://docs.google.com/javax/swing/text/ParagraphView.html)   [**NEXT CLASS**](http://docs.google.com/javax/swing/text/PlainDocument.html) | [**FRAMES**](http://docs.google.com/index.html?javax/swing/text/PasswordView.html)    [**NO FRAMES**](http://docs.google.com/PasswordView.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | [FIELD](#2et92p0) | [CONSTR](#1t3h5sf) | [METHOD](#4d34og8) | DETAIL: FIELD | [CONSTR](#lnxbz9) | [METHOD](#1ksv4uv) |

[Submit a bug or feature](http://bugs.sun.com/services/bugreport/index.jsp)

For further API reference and developer documentation, see [Java SE Developer Documentation](http://docs.google.com/webnotes/devdocs-vs-specs.html). That documentation contains more detailed, developer-targeted descriptions, with conceptual overviews, definitions of terms, workarounds, and working code examples.

Copyright 2006 Sun Microsystems, Inc. All rights reserved. Use is subject to [license terms](http://docs.google.com/legal/license.html). Also see the [documentation redistribution policy](http://java.sun.com/docs/redist.html).