| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | [**Class**](http://docs.google.com/javax/swing/text/TabExpander.html) | **Use** | [**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   NEXT | [**FRAMES**](http://docs.google.com/index.html?javax/swing/text//class-useTabExpander.html)    [**NO FRAMES**](http://docs.google.com/TabExpander.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |

**Uses of Interface**

**javax.swing.text.TabExpander**

| Packages that use [TabExpander](http://docs.google.com/javax/swing/text/TabExpander.html) | |
| --- | --- |
| [**javax.swing.text**](#3znysh7) | Provides classes and interfaces that deal with editable and noneditable text components. |

| Uses of [TabExpander](http://docs.google.com/javax/swing/text/TabExpander.html) in [javax.swing.text](http://docs.google.com/javax/swing/text/package-summary.html) | |
| --- | --- |

| Classes in [javax.swing.text](http://docs.google.com/javax/swing/text/package-summary.html) that implement [TabExpander](http://docs.google.com/javax/swing/text/TabExpander.html) | |
| --- | --- |
| class | [**FieldView**](http://docs.google.com/javax/swing/text/FieldView.html)            Extends the multi-line plain text view to be suitable for a single-line editor view. |
| class | [**ParagraphView**](http://docs.google.com/javax/swing/text/ParagraphView.html)            View of a simple line-wrapping paragraph that supports multiple fonts, colors, components, icons, etc. |
| class | [**PasswordView**](http://docs.google.com/javax/swing/text/PasswordView.html)            Implements a View suitable for use in JPasswordField UI implementations. |
| class | [**PlainView**](http://docs.google.com/javax/swing/text/PlainView.html)            Implements View interface for a simple multi-line text view that has text in one font and color. |
| class | [**WrappedPlainView**](http://docs.google.com/javax/swing/text/WrappedPlainView.html)            View of plain text (text with only one font and color) that does line-wrapping. |

| Methods in [javax.swing.text](http://docs.google.com/javax/swing/text/package-summary.html) that return [TabExpander](http://docs.google.com/javax/swing/text/TabExpander.html) | |
| --- | --- |
| [TabExpander](http://docs.google.com/javax/swing/text/TabExpander.html) | **GlyphView.**[**getTabExpander**](http://docs.google.com/javax/swing/text/GlyphView.html#getTabExpander())()            Fetch the TabExpander to use if tabs are present in this view. |

| Methods in [javax.swing.text](http://docs.google.com/javax/swing/text/package-summary.html) with parameters of type [TabExpander](http://docs.google.com/javax/swing/text/TabExpander.html) | |
| --- | --- |
| static int | **Utilities.**[**drawTabbedText**](http://docs.google.com/javax/swing/text/Utilities.html#drawTabbedText(javax.swing.text.Segment,%20int,%20int,%20java.awt.Graphics,%20javax.swing.text.TabExpander,%20int))([Segment](http://docs.google.com/javax/swing/text/Segment.html) s, int x, int y, [Graphics](http://docs.google.com/java/awt/Graphics.html) g, [TabExpander](http://docs.google.com/javax/swing/text/TabExpander.html) e, int startOffset)            Draws the given text, expanding any tabs that are contained using the given tab expansion technique. |
| static int | **Utilities.**[**getBreakLocation**](http://docs.google.com/javax/swing/text/Utilities.html#getBreakLocation(javax.swing.text.Segment,%20java.awt.FontMetrics,%20int,%20int,%20javax.swing.text.TabExpander,%20int))([Segment](http://docs.google.com/javax/swing/text/Segment.html) s, [FontMetrics](http://docs.google.com/java/awt/FontMetrics.html) metrics, int x0, int x, [TabExpander](http://docs.google.com/javax/swing/text/TabExpander.html) e, int startOffset)            Determine where to break the given text to fit within the given span. |
| abstract  float | **GlyphView.GlyphPainter.**[**getSpan**](http://docs.google.com/javax/swing/text/GlyphView.GlyphPainter.html#getSpan(javax.swing.text.GlyphView,%20int,%20int,%20javax.swing.text.TabExpander,%20float))([GlyphView](http://docs.google.com/javax/swing/text/GlyphView.html) v, int p0, int p1, [TabExpander](http://docs.google.com/javax/swing/text/TabExpander.html) e, float x)            Determine the span the glyphs given a start location (for tab expansion). |
| float | **TabableView.**[**getTabbedSpan**](http://docs.google.com/javax/swing/text/TabableView.html#getTabbedSpan(float,%20javax.swing.text.TabExpander))(float x, [TabExpander](http://docs.google.com/javax/swing/text/TabExpander.html) e)            Determines the desired span when using the given tab expansion implementation. |
| float | **GlyphView.**[**getTabbedSpan**](http://docs.google.com/javax/swing/text/GlyphView.html#getTabbedSpan(float,%20javax.swing.text.TabExpander))(float x, [TabExpander](http://docs.google.com/javax/swing/text/TabExpander.html) e)            Determines the desired span when using the given tab expansion implementation. |
| static int | **Utilities.**[**getTabbedTextOffset**](http://docs.google.com/javax/swing/text/Utilities.html#getTabbedTextOffset(javax.swing.text.Segment,%20java.awt.FontMetrics,%20int,%20int,%20javax.swing.text.TabExpander,%20int))([Segment](http://docs.google.com/javax/swing/text/Segment.html) s, [FontMetrics](http://docs.google.com/java/awt/FontMetrics.html) metrics, int x0, int x, [TabExpander](http://docs.google.com/javax/swing/text/TabExpander.html) e, int startOffset)            Determines the relative offset into the given text that best represents the given span in the view coordinate system. |
| static int | **Utilities.**[**getTabbedTextOffset**](http://docs.google.com/javax/swing/text/Utilities.html#getTabbedTextOffset(javax.swing.text.Segment,%20java.awt.FontMetrics,%20int,%20int,%20javax.swing.text.TabExpander,%20int,%20boolean))([Segment](http://docs.google.com/javax/swing/text/Segment.html) s, [FontMetrics](http://docs.google.com/java/awt/FontMetrics.html) metrics, int x0, int x, [TabExpander](http://docs.google.com/javax/swing/text/TabExpander.html) e, int startOffset, boolean round) |
| static int | **Utilities.**[**getTabbedTextWidth**](http://docs.google.com/javax/swing/text/Utilities.html#getTabbedTextWidth(javax.swing.text.Segment,%20java.awt.FontMetrics,%20int,%20javax.swing.text.TabExpander,%20int))([Segment](http://docs.google.com/javax/swing/text/Segment.html) s, [FontMetrics](http://docs.google.com/java/awt/FontMetrics.html) metrics, int x, [TabExpander](http://docs.google.com/javax/swing/text/TabExpander.html) e, int startOffset)            Determines the width of the given segment of text taking tabs into consideration. |

| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | [**Class**](http://docs.google.com/javax/swing/text/TabExpander.html) | **Use** | [**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   NEXT | [**FRAMES**](http://docs.google.com/index.html?javax/swing/text//class-useTabExpander.html)    [**NO FRAMES**](http://docs.google.com/TabExpander.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |

[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).