| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/DefaultStyledDocument.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/DefaultHighlighter.DefaultHighlightPainter.html)   [**NEXT CLASS**](http://docs.google.com/javax/swing/text/DefaultStyledDocument.AttributeUndoableEdit.html) | [**FRAMES**](http://docs.google.com/index.html?javax/swing/text/DefaultStyledDocument.html)    [**NO FRAMES**](http://docs.google.com/DefaultStyledDocument.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: [NESTED](#3znysh7) | [FIELD](#tyjcwt) | [CONSTR](#4d34og8) | [METHOD](#2s8eyo1) | DETAIL: [FIELD](#lnxbz9) | [CONSTR](#44sinio) | [METHOD](#1y810tw) |

## **javax.swing.text**

Class DefaultStyledDocument

[java.lang.Object](http://docs.google.com/java/lang/Object.html)  
 [javax.swing.text.AbstractDocument](http://docs.google.com/javax/swing/text/AbstractDocument.html)  
 **javax.swing.text.DefaultStyledDocument**

**All Implemented Interfaces:** [Serializable](http://docs.google.com/java/io/Serializable.html), [Document](http://docs.google.com/javax/swing/text/Document.html), [StyledDocument](http://docs.google.com/javax/swing/text/StyledDocument.html) **Direct Known Subclasses:** [HTMLDocument](http://docs.google.com/javax/swing/text/html/HTMLDocument.html)

public class **DefaultStyledDocument**extends [AbstractDocument](http://docs.google.com/javax/swing/text/AbstractDocument.html)implements [StyledDocument](http://docs.google.com/javax/swing/text/StyledDocument.html)

A document that can be marked up with character and paragraph styles in a manner similar to the Rich Text Format. The element structure for this document represents style crossings for style runs. These style runs are mapped into a paragraph element structure (which may reside in some other structure). The style runs break at paragraph boundaries since logical styles are assigned to paragraph boundaries.

**Warning:** Serialized objects of this class will not be compatible with future Swing releases. The current serialization support is appropriate for short term storage or RMI between applications running the same version of Swing. As of 1.4, support for long term storage of all JavaBeansTM has been added to the java.beans package. Please see [XMLEncoder](http://docs.google.com/java/beans/XMLEncoder.html).

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

| **Nested Class Summary** | |
| --- | --- |
| static class | [**DefaultStyledDocument.AttributeUndoableEdit**](http://docs.google.com/javax/swing/text/DefaultStyledDocument.AttributeUndoableEdit.html)            An UndoableEdit used to remember AttributeSet changes to an Element. |
| class | [**DefaultStyledDocument.ElementBuffer**](http://docs.google.com/javax/swing/text/DefaultStyledDocument.ElementBuffer.html)            Class to manage changes to the element hierarchy. |
| static class | [**DefaultStyledDocument.ElementSpec**](http://docs.google.com/javax/swing/text/DefaultStyledDocument.ElementSpec.html)            Specification for building elements. |
| protected  class | [**DefaultStyledDocument.SectionElement**](http://docs.google.com/javax/swing/text/DefaultStyledDocument.SectionElement.html)            Default root element for a document... |

| **Nested classes/interfaces inherited from class javax.swing.text.**[**AbstractDocument**](http://docs.google.com/javax/swing/text/AbstractDocument.html) |
| --- |
| [AbstractDocument.AbstractElement](http://docs.google.com/javax/swing/text/AbstractDocument.AbstractElement.html), [AbstractDocument.AttributeContext](http://docs.google.com/javax/swing/text/AbstractDocument.AttributeContext.html), [AbstractDocument.BranchElement](http://docs.google.com/javax/swing/text/AbstractDocument.BranchElement.html), [AbstractDocument.Content](http://docs.google.com/javax/swing/text/AbstractDocument.Content.html), [AbstractDocument.DefaultDocumentEvent](http://docs.google.com/javax/swing/text/AbstractDocument.DefaultDocumentEvent.html), [AbstractDocument.ElementEdit](http://docs.google.com/javax/swing/text/AbstractDocument.ElementEdit.html), [AbstractDocument.LeafElement](http://docs.google.com/javax/swing/text/AbstractDocument.LeafElement.html) |

| **Field Summary** | |
| --- | --- |
| protected  [DefaultStyledDocument.ElementBuffer](http://docs.google.com/javax/swing/text/DefaultStyledDocument.ElementBuffer.html) | [**buffer**](http://docs.google.com/javax/swing/text/DefaultStyledDocument.html#buffer) |
| static int | [**BUFFER\_SIZE\_DEFAULT**](http://docs.google.com/javax/swing/text/DefaultStyledDocument.html#BUFFER_SIZE_DEFAULT)            The default size of the initial content buffer. |

| **Fields inherited from class javax.swing.text.**[**AbstractDocument**](http://docs.google.com/javax/swing/text/AbstractDocument.html) |
| --- |
| [BAD\_LOCATION](http://docs.google.com/javax/swing/text/AbstractDocument.html#BAD_LOCATION), [BidiElementName](http://docs.google.com/javax/swing/text/AbstractDocument.html#BidiElementName), [ContentElementName](http://docs.google.com/javax/swing/text/AbstractDocument.html#ContentElementName), [ElementNameAttribute](http://docs.google.com/javax/swing/text/AbstractDocument.html#ElementNameAttribute), [listenerList](http://docs.google.com/javax/swing/text/AbstractDocument.html#listenerList), [ParagraphElementName](http://docs.google.com/javax/swing/text/AbstractDocument.html#ParagraphElementName), [SectionElementName](http://docs.google.com/javax/swing/text/AbstractDocument.html#SectionElementName) |

| **Fields inherited from interface javax.swing.text.**[**Document**](http://docs.google.com/javax/swing/text/Document.html) |
| --- |
| [StreamDescriptionProperty](http://docs.google.com/javax/swing/text/Document.html#StreamDescriptionProperty), [TitleProperty](http://docs.google.com/javax/swing/text/Document.html#TitleProperty) |

| **Constructor Summary** | |
| --- | --- |
| [**DefaultStyledDocument**](http://docs.google.com/javax/swing/text/DefaultStyledDocument.html#DefaultStyledDocument())()            Constructs a default styled document. |
| [**DefaultStyledDocument**](http://docs.google.com/javax/swing/text/DefaultStyledDocument.html#DefaultStyledDocument(javax.swing.text.AbstractDocument.Content,%20javax.swing.text.StyleContext))([AbstractDocument.Content](http://docs.google.com/javax/swing/text/AbstractDocument.Content.html) c, [StyleContext](http://docs.google.com/javax/swing/text/StyleContext.html) styles)            Constructs a styled document. |
| [**DefaultStyledDocument**](http://docs.google.com/javax/swing/text/DefaultStyledDocument.html#DefaultStyledDocument(javax.swing.text.StyleContext))([StyleContext](http://docs.google.com/javax/swing/text/StyleContext.html) styles)            Constructs a styled document with the default content storage implementation and a shared set of styles. |

| **Method Summary** | |
| --- | --- |
| void | [**addDocumentListener**](http://docs.google.com/javax/swing/text/DefaultStyledDocument.html#addDocumentListener(javax.swing.event.DocumentListener))([DocumentListener](http://docs.google.com/javax/swing/event/DocumentListener.html) listener)            Adds a document listener for notification of any changes. |
| [Style](http://docs.google.com/javax/swing/text/Style.html) | [**addStyle**](http://docs.google.com/javax/swing/text/DefaultStyledDocument.html#addStyle(java.lang.String,%20javax.swing.text.Style))([String](http://docs.google.com/java/lang/String.html) nm, [Style](http://docs.google.com/javax/swing/text/Style.html) parent)            Adds a new style into the logical style hierarchy. |
| protected  void | [**create**](http://docs.google.com/javax/swing/text/DefaultStyledDocument.html#create(javax.swing.text.DefaultStyledDocument.ElementSpec%5B%5D))([DefaultStyledDocument.ElementSpec](http://docs.google.com/javax/swing/text/DefaultStyledDocument.ElementSpec.html)[] data)            Initialize the document to reflect the given element structure (i.e. |
| protected  [AbstractDocument.AbstractElement](http://docs.google.com/javax/swing/text/AbstractDocument.AbstractElement.html) | [**createDefaultRoot**](http://docs.google.com/javax/swing/text/DefaultStyledDocument.html#createDefaultRoot())()            Creates the root element to be used to represent the default document structure. |
| [Color](http://docs.google.com/java/awt/Color.html) | [**getBackground**](http://docs.google.com/javax/swing/text/DefaultStyledDocument.html#getBackground(javax.swing.text.AttributeSet))([AttributeSet](http://docs.google.com/javax/swing/text/AttributeSet.html) attr)            Gets the background color from an attribute set. |
| [Element](http://docs.google.com/javax/swing/text/Element.html) | [**getCharacterElement**](http://docs.google.com/javax/swing/text/DefaultStyledDocument.html#getCharacterElement(int))(int pos)            Gets a character element based on a position. |
| [Element](http://docs.google.com/javax/swing/text/Element.html) | [**getDefaultRootElement**](http://docs.google.com/javax/swing/text/DefaultStyledDocument.html#getDefaultRootElement())()            Gets the default root element. |
| [Font](http://docs.google.com/java/awt/Font.html) | [**getFont**](http://docs.google.com/javax/swing/text/DefaultStyledDocument.html#getFont(javax.swing.text.AttributeSet))([AttributeSet](http://docs.google.com/javax/swing/text/AttributeSet.html) attr)            Gets the font from an attribute set. |
| [Color](http://docs.google.com/java/awt/Color.html) | [**getForeground**](http://docs.google.com/javax/swing/text/DefaultStyledDocument.html#getForeground(javax.swing.text.AttributeSet))([AttributeSet](http://docs.google.com/javax/swing/text/AttributeSet.html) attr)            Gets the foreground color from an attribute set. |
| [Style](http://docs.google.com/javax/swing/text/Style.html) | [**getLogicalStyle**](http://docs.google.com/javax/swing/text/DefaultStyledDocument.html#getLogicalStyle(int))(int p)            Fetches the logical style assigned to the paragraph represented by the given position. |
| [Element](http://docs.google.com/javax/swing/text/Element.html) | [**getParagraphElement**](http://docs.google.com/javax/swing/text/DefaultStyledDocument.html#getParagraphElement(int))(int pos)            Gets the paragraph element at the offset pos. |
| [Style](http://docs.google.com/javax/swing/text/Style.html) | [**getStyle**](http://docs.google.com/javax/swing/text/DefaultStyledDocument.html#getStyle(java.lang.String))([String](http://docs.google.com/java/lang/String.html) nm)            Fetches a named style previously added. |
| [Enumeration](http://docs.google.com/java/util/Enumeration.html)<?> | [**getStyleNames**](http://docs.google.com/javax/swing/text/DefaultStyledDocument.html#getStyleNames())()            Fetches the list of of style names. |
| protected  void | [**insert**](http://docs.google.com/javax/swing/text/DefaultStyledDocument.html#insert(int,%20javax.swing.text.DefaultStyledDocument.ElementSpec%5B%5D))(int offset, [DefaultStyledDocument.ElementSpec](http://docs.google.com/javax/swing/text/DefaultStyledDocument.ElementSpec.html)[] data)            Inserts new elements in bulk. |
| protected  void | [**insertUpdate**](http://docs.google.com/javax/swing/text/DefaultStyledDocument.html#insertUpdate(javax.swing.text.AbstractDocument.DefaultDocumentEvent,%20javax.swing.text.AttributeSet))([AbstractDocument.DefaultDocumentEvent](http://docs.google.com/javax/swing/text/AbstractDocument.DefaultDocumentEvent.html) chng, [AttributeSet](http://docs.google.com/javax/swing/text/AttributeSet.html) attr)            Updates document structure as a result of text insertion. |
| void | [**removeDocumentListener**](http://docs.google.com/javax/swing/text/DefaultStyledDocument.html#removeDocumentListener(javax.swing.event.DocumentListener))([DocumentListener](http://docs.google.com/javax/swing/event/DocumentListener.html) listener)            Removes a document listener. |
| void | [**removeStyle**](http://docs.google.com/javax/swing/text/DefaultStyledDocument.html#removeStyle(java.lang.String))([String](http://docs.google.com/java/lang/String.html) nm)            Removes a named style previously added to the document. |
| protected  void | [**removeUpdate**](http://docs.google.com/javax/swing/text/DefaultStyledDocument.html#removeUpdate(javax.swing.text.AbstractDocument.DefaultDocumentEvent))([AbstractDocument.DefaultDocumentEvent](http://docs.google.com/javax/swing/text/AbstractDocument.DefaultDocumentEvent.html) chng)            Updates document structure as a result of text removal. |
| void | [**setCharacterAttributes**](http://docs.google.com/javax/swing/text/DefaultStyledDocument.html#setCharacterAttributes(int,%20int,%20javax.swing.text.AttributeSet,%20boolean))(int offset, int length, [AttributeSet](http://docs.google.com/javax/swing/text/AttributeSet.html) s, boolean replace)            Sets attributes for some part of the document. |
| void | [**setLogicalStyle**](http://docs.google.com/javax/swing/text/DefaultStyledDocument.html#setLogicalStyle(int,%20javax.swing.text.Style))(int pos, [Style](http://docs.google.com/javax/swing/text/Style.html) s)            Sets the logical style to use for the paragraph at the given position. |
| void | [**setParagraphAttributes**](http://docs.google.com/javax/swing/text/DefaultStyledDocument.html#setParagraphAttributes(int,%20int,%20javax.swing.text.AttributeSet,%20boolean))(int offset, int length, [AttributeSet](http://docs.google.com/javax/swing/text/AttributeSet.html) s, boolean replace)            Sets attributes for a paragraph. |
| protected  void | [**styleChanged**](http://docs.google.com/javax/swing/text/DefaultStyledDocument.html#styleChanged(javax.swing.text.Style))([Style](http://docs.google.com/javax/swing/text/Style.html) style)            Called when any of this document's styles have changed. |

| **Methods inherited from class javax.swing.text.**[**AbstractDocument**](http://docs.google.com/javax/swing/text/AbstractDocument.html) |
| --- |
| [addUndoableEditListener](http://docs.google.com/javax/swing/text/AbstractDocument.html#addUndoableEditListener(javax.swing.event.UndoableEditListener)), [createBranchElement](http://docs.google.com/javax/swing/text/AbstractDocument.html#createBranchElement(javax.swing.text.Element,%20javax.swing.text.AttributeSet)), [createLeafElement](http://docs.google.com/javax/swing/text/AbstractDocument.html#createLeafElement(javax.swing.text.Element,%20javax.swing.text.AttributeSet,%20int,%20int)), [createPosition](http://docs.google.com/javax/swing/text/AbstractDocument.html#createPosition(int)), [dump](http://docs.google.com/javax/swing/text/AbstractDocument.html#dump(java.io.PrintStream)), [fireChangedUpdate](http://docs.google.com/javax/swing/text/AbstractDocument.html#fireChangedUpdate(javax.swing.event.DocumentEvent)), [fireInsertUpdate](http://docs.google.com/javax/swing/text/AbstractDocument.html#fireInsertUpdate(javax.swing.event.DocumentEvent)), [fireRemoveUpdate](http://docs.google.com/javax/swing/text/AbstractDocument.html#fireRemoveUpdate(javax.swing.event.DocumentEvent)), [fireUndoableEditUpdate](http://docs.google.com/javax/swing/text/AbstractDocument.html#fireUndoableEditUpdate(javax.swing.event.UndoableEditEvent)), [getAsynchronousLoadPriority](http://docs.google.com/javax/swing/text/AbstractDocument.html#getAsynchronousLoadPriority()), [getAttributeContext](http://docs.google.com/javax/swing/text/AbstractDocument.html#getAttributeContext()), [getBidiRootElement](http://docs.google.com/javax/swing/text/AbstractDocument.html#getBidiRootElement()), [getContent](http://docs.google.com/javax/swing/text/AbstractDocument.html#getContent()), [getCurrentWriter](http://docs.google.com/javax/swing/text/AbstractDocument.html#getCurrentWriter()), [getDocumentFilter](http://docs.google.com/javax/swing/text/AbstractDocument.html#getDocumentFilter()), [getDocumentListeners](http://docs.google.com/javax/swing/text/AbstractDocument.html#getDocumentListeners()), [getDocumentProperties](http://docs.google.com/javax/swing/text/AbstractDocument.html#getDocumentProperties()), [getEndPosition](http://docs.google.com/javax/swing/text/AbstractDocument.html#getEndPosition()), [getLength](http://docs.google.com/javax/swing/text/AbstractDocument.html#getLength()), [getListeners](http://docs.google.com/javax/swing/text/AbstractDocument.html#getListeners(java.lang.Class)), [getProperty](http://docs.google.com/javax/swing/text/AbstractDocument.html#getProperty(java.lang.Object)), [getRootElements](http://docs.google.com/javax/swing/text/AbstractDocument.html#getRootElements()), [getStartPosition](http://docs.google.com/javax/swing/text/AbstractDocument.html#getStartPosition()), [getText](http://docs.google.com/javax/swing/text/AbstractDocument.html#getText(int,%20int)), [getText](http://docs.google.com/javax/swing/text/AbstractDocument.html#getText(int,%20int,%20javax.swing.text.Segment)), [getUndoableEditListeners](http://docs.google.com/javax/swing/text/AbstractDocument.html#getUndoableEditListeners()), [insertString](http://docs.google.com/javax/swing/text/AbstractDocument.html#insertString(int,%20java.lang.String,%20javax.swing.text.AttributeSet)), [postRemoveUpdate](http://docs.google.com/javax/swing/text/AbstractDocument.html#postRemoveUpdate(javax.swing.text.AbstractDocument.DefaultDocumentEvent)), [putProperty](http://docs.google.com/javax/swing/text/AbstractDocument.html#putProperty(java.lang.Object,%20java.lang.Object)), [readLock](http://docs.google.com/javax/swing/text/AbstractDocument.html#readLock()), [readUnlock](http://docs.google.com/javax/swing/text/AbstractDocument.html#readUnlock()), [remove](http://docs.google.com/javax/swing/text/AbstractDocument.html#remove(int,%20int)), [removeUndoableEditListener](http://docs.google.com/javax/swing/text/AbstractDocument.html#removeUndoableEditListener(javax.swing.event.UndoableEditListener)), [render](http://docs.google.com/javax/swing/text/AbstractDocument.html#render(java.lang.Runnable)), [replace](http://docs.google.com/javax/swing/text/AbstractDocument.html#replace(int,%20int,%20java.lang.String,%20javax.swing.text.AttributeSet)), [setAsynchronousLoadPriority](http://docs.google.com/javax/swing/text/AbstractDocument.html#setAsynchronousLoadPriority(int)), [setDocumentFilter](http://docs.google.com/javax/swing/text/AbstractDocument.html#setDocumentFilter(javax.swing.text.DocumentFilter)), [setDocumentProperties](http://docs.google.com/javax/swing/text/AbstractDocument.html#setDocumentProperties(java.util.Dictionary)), [writeLock](http://docs.google.com/javax/swing/text/AbstractDocument.html#writeLock()), [writeUnlock](http://docs.google.com/javax/swing/text/AbstractDocument.html#writeUnlock()) |

| **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)) |

| **Methods inherited from interface javax.swing.text.**[**Document**](http://docs.google.com/javax/swing/text/Document.html) |
| --- |
| [addUndoableEditListener](http://docs.google.com/javax/swing/text/Document.html#addUndoableEditListener(javax.swing.event.UndoableEditListener)), [createPosition](http://docs.google.com/javax/swing/text/Document.html#createPosition(int)), [getEndPosition](http://docs.google.com/javax/swing/text/Document.html#getEndPosition()), [getLength](http://docs.google.com/javax/swing/text/Document.html#getLength()), [getProperty](http://docs.google.com/javax/swing/text/Document.html#getProperty(java.lang.Object)), [getRootElements](http://docs.google.com/javax/swing/text/Document.html#getRootElements()), [getStartPosition](http://docs.google.com/javax/swing/text/Document.html#getStartPosition()), [getText](http://docs.google.com/javax/swing/text/Document.html#getText(int,%20int)), [getText](http://docs.google.com/javax/swing/text/Document.html#getText(int,%20int,%20javax.swing.text.Segment)), [insertString](http://docs.google.com/javax/swing/text/Document.html#insertString(int,%20java.lang.String,%20javax.swing.text.AttributeSet)), [putProperty](http://docs.google.com/javax/swing/text/Document.html#putProperty(java.lang.Object,%20java.lang.Object)), [remove](http://docs.google.com/javax/swing/text/Document.html#remove(int,%20int)), [removeUndoableEditListener](http://docs.google.com/javax/swing/text/Document.html#removeUndoableEditListener(javax.swing.event.UndoableEditListener)), [render](http://docs.google.com/javax/swing/text/Document.html#render(java.lang.Runnable)) |

| **Field Detail** |
| --- |

### BUFFER\_SIZE\_DEFAULT

public static final int **BUFFER\_SIZE\_DEFAULT**

The default size of the initial content buffer.

**See Also:**[Constant Field Values](http://docs.google.com/constant-values.html#javax.swing.text.DefaultStyledDocument.BUFFER_SIZE_DEFAULT)

### buffer

protected [DefaultStyledDocument.ElementBuffer](http://docs.google.com/javax/swing/text/DefaultStyledDocument.ElementBuffer.html) **buffer**

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

### DefaultStyledDocument

public **DefaultStyledDocument**([AbstractDocument.Content](http://docs.google.com/javax/swing/text/AbstractDocument.Content.html) c,  
 [StyleContext](http://docs.google.com/javax/swing/text/StyleContext.html) styles)

Constructs a styled document.

**Parameters:**c - the container for the contentstyles - resources and style definitions which may be shared across documents

### DefaultStyledDocument

public **DefaultStyledDocument**([StyleContext](http://docs.google.com/javax/swing/text/StyleContext.html) styles)

Constructs a styled document with the default content storage implementation and a shared set of styles.

**Parameters:**styles - the styles

### DefaultStyledDocument

public **DefaultStyledDocument**()

Constructs a default styled document. This buffers input content by a size of *BUFFER\_SIZE\_DEFAULT* and has a style context that is scoped by the lifetime of the document and is not shared with other documents.

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

### getDefaultRootElement

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

Gets the default root element.

**Specified by:**[getDefaultRootElement](http://docs.google.com/javax/swing/text/Document.html#getDefaultRootElement()) in interface [Document](http://docs.google.com/javax/swing/text/Document.html)**Specified by:**[getDefaultRootElement](http://docs.google.com/javax/swing/text/AbstractDocument.html#getDefaultRootElement()) in class [AbstractDocument](http://docs.google.com/javax/swing/text/AbstractDocument.html) **Returns:**the root**See Also:**[Document.getDefaultRootElement()](http://docs.google.com/javax/swing/text/Document.html#getDefaultRootElement())

### create

protected void **create**([DefaultStyledDocument.ElementSpec](http://docs.google.com/javax/swing/text/DefaultStyledDocument.ElementSpec.html)[] data)

Initialize the document to reflect the given element structure (i.e. the structure reported by the getDefaultRootElement method. If the document contained any data it will first be removed.

### insert

protected void **insert**(int offset,  
 [DefaultStyledDocument.ElementSpec](http://docs.google.com/javax/swing/text/DefaultStyledDocument.ElementSpec.html)[] data)  
 throws [BadLocationException](http://docs.google.com/javax/swing/text/BadLocationException.html)

Inserts new elements in bulk. This is useful to allow parsing with the document in an unlocked state and prepare an element structure modification. This method takes an array of tokens that describe how to update an element structure so the time within a write lock can be greatly reduced in an asynchronous update situation.

This method is thread safe, although most Swing methods are not. Please see [How to Use Threads](http://java.sun.com/docs/books/tutorial/uiswing/misc/threads.html) for more information.

**Parameters:**offset - the starting offset >= 0data - the element data **Throws:** [BadLocationException](http://docs.google.com/javax/swing/text/BadLocationException.html) - for an invalid starting offset

### addStyle

public [Style](http://docs.google.com/javax/swing/text/Style.html) **addStyle**([String](http://docs.google.com/java/lang/String.html) nm,  
 [Style](http://docs.google.com/javax/swing/text/Style.html) parent)

Adds a new style into the logical style hierarchy. Style attributes resolve from bottom up so an attribute specified in a child will override an attribute specified in the parent.

**Specified by:**[addStyle](http://docs.google.com/javax/swing/text/StyledDocument.html#addStyle(java.lang.String,%20javax.swing.text.Style)) in interface [StyledDocument](http://docs.google.com/javax/swing/text/StyledDocument.html) **Parameters:**nm - the name of the style (must be unique within the collection of named styles). The name may be null if the style is unnamed, but the caller is responsible for managing the reference returned as an unnamed style can't be fetched by name. An unnamed style may be useful for things like character attribute overrides such as found in a style run.parent - the parent style. This may be null if unspecified attributes need not be resolved in some other style. **Returns:**the style

### removeStyle

public void **removeStyle**([String](http://docs.google.com/java/lang/String.html) nm)

Removes a named style previously added to the document.

**Specified by:**[removeStyle](http://docs.google.com/javax/swing/text/StyledDocument.html#removeStyle(java.lang.String)) in interface [StyledDocument](http://docs.google.com/javax/swing/text/StyledDocument.html) **Parameters:**nm - the name of the style to remove

### getStyle

public [Style](http://docs.google.com/javax/swing/text/Style.html) **getStyle**([String](http://docs.google.com/java/lang/String.html) nm)

Fetches a named style previously added.

**Specified by:**[getStyle](http://docs.google.com/javax/swing/text/StyledDocument.html#getStyle(java.lang.String)) in interface [StyledDocument](http://docs.google.com/javax/swing/text/StyledDocument.html) **Parameters:**nm - the name of the style **Returns:**the style

### getStyleNames

public [Enumeration](http://docs.google.com/java/util/Enumeration.html)<?> **getStyleNames**()

Fetches the list of of style names.

**Returns:**all the style names

### setLogicalStyle

public void **setLogicalStyle**(int pos,  
 [Style](http://docs.google.com/javax/swing/text/Style.html) s)

Sets the logical style to use for the paragraph at the given position. If attributes aren't explicitly set for character and paragraph attributes they will resolve through the logical style assigned to the paragraph, which in turn may resolve through some hierarchy completely independent of the element hierarchy in the document.

This method is thread safe, although most Swing methods are not. Please see [How to Use Threads](http://java.sun.com/docs/books/tutorial/uiswing/misc/threads.html) for more information.

**Specified by:**[setLogicalStyle](http://docs.google.com/javax/swing/text/StyledDocument.html#setLogicalStyle(int,%20javax.swing.text.Style)) in interface [StyledDocument](http://docs.google.com/javax/swing/text/StyledDocument.html) **Parameters:**pos - the offset from the start of the document >= 0s - the logical style to assign to the paragraph, null if none

### getLogicalStyle

public [Style](http://docs.google.com/javax/swing/text/Style.html) **getLogicalStyle**(int p)

Fetches the logical style assigned to the paragraph represented by the given position.

**Specified by:**[getLogicalStyle](http://docs.google.com/javax/swing/text/StyledDocument.html#getLogicalStyle(int)) in interface [StyledDocument](http://docs.google.com/javax/swing/text/StyledDocument.html) **Parameters:**p - the location to translate to a paragraph and determine the logical style assigned >= 0. This is an offset from the start of the document. **Returns:**the style, null if none

### setCharacterAttributes

public void **setCharacterAttributes**(int offset,  
 int length,  
 [AttributeSet](http://docs.google.com/javax/swing/text/AttributeSet.html) s,  
 boolean replace)

Sets attributes for some part of the document. A write lock is held by this operation while changes are being made, and a DocumentEvent is sent to the listeners after the change has been successfully completed.

This method is thread safe, although most Swing methods are not. Please see [How to Use Threads](http://java.sun.com/docs/books/tutorial/uiswing/misc/threads.html) for more information.

**Specified by:**[setCharacterAttributes](http://docs.google.com/javax/swing/text/StyledDocument.html#setCharacterAttributes(int,%20int,%20javax.swing.text.AttributeSet,%20boolean)) in interface [StyledDocument](http://docs.google.com/javax/swing/text/StyledDocument.html) **Parameters:**offset - the offset in the document >= 0length - the length >= 0s - the attributesreplace - true if the previous attributes should be replaced before setting the new attributes

### setParagraphAttributes

public void **setParagraphAttributes**(int offset,  
 int length,  
 [AttributeSet](http://docs.google.com/javax/swing/text/AttributeSet.html) s,  
 boolean replace)

Sets attributes for a paragraph.

This method is thread safe, although most Swing methods are not. Please see [How to Use Threads](http://java.sun.com/docs/books/tutorial/uiswing/misc/threads.html) for more information.

**Specified by:**[setParagraphAttributes](http://docs.google.com/javax/swing/text/StyledDocument.html#setParagraphAttributes(int,%20int,%20javax.swing.text.AttributeSet,%20boolean)) in interface [StyledDocument](http://docs.google.com/javax/swing/text/StyledDocument.html) **Parameters:**offset - the offset into the paragraph >= 0length - the number of characters affected >= 0s - the attributesreplace - whether to replace existing attributes, or merge them

### getParagraphElement

public [Element](http://docs.google.com/javax/swing/text/Element.html) **getParagraphElement**(int pos)

Gets the paragraph element at the offset pos. A paragraph consists of at least one child Element, which is usually a leaf.

**Specified by:**[getParagraphElement](http://docs.google.com/javax/swing/text/StyledDocument.html#getParagraphElement(int)) in interface [StyledDocument](http://docs.google.com/javax/swing/text/StyledDocument.html)**Specified by:**[getParagraphElement](http://docs.google.com/javax/swing/text/AbstractDocument.html#getParagraphElement(int)) in class [AbstractDocument](http://docs.google.com/javax/swing/text/AbstractDocument.html) **Parameters:**pos - the starting offset >= 0 **Returns:**the element

### getCharacterElement

public [Element](http://docs.google.com/javax/swing/text/Element.html) **getCharacterElement**(int pos)

Gets a character element based on a position.

**Specified by:**[getCharacterElement](http://docs.google.com/javax/swing/text/StyledDocument.html#getCharacterElement(int)) in interface [StyledDocument](http://docs.google.com/javax/swing/text/StyledDocument.html) **Parameters:**pos - the position in the document >= 0 **Returns:**the element

### insertUpdate

protected void **insertUpdate**([AbstractDocument.DefaultDocumentEvent](http://docs.google.com/javax/swing/text/AbstractDocument.DefaultDocumentEvent.html) chng,  
 [AttributeSet](http://docs.google.com/javax/swing/text/AttributeSet.html) attr)

Updates document structure as a result of text insertion. This will happen within a write lock. This implementation simply parses the inserted content for line breaks and builds up a set of instructions for the element buffer.

**Overrides:**[insertUpdate](http://docs.google.com/javax/swing/text/AbstractDocument.html#insertUpdate(javax.swing.text.AbstractDocument.DefaultDocumentEvent,%20javax.swing.text.AttributeSet)) in class [AbstractDocument](http://docs.google.com/javax/swing/text/AbstractDocument.html) **Parameters:**chng - a description of the document changeattr - the attributes

### removeUpdate

protected void **removeUpdate**([AbstractDocument.DefaultDocumentEvent](http://docs.google.com/javax/swing/text/AbstractDocument.DefaultDocumentEvent.html) chng)

Updates document structure as a result of text removal.

**Overrides:**[removeUpdate](http://docs.google.com/javax/swing/text/AbstractDocument.html#removeUpdate(javax.swing.text.AbstractDocument.DefaultDocumentEvent)) in class [AbstractDocument](http://docs.google.com/javax/swing/text/AbstractDocument.html) **Parameters:**chng - a description of the document change

### createDefaultRoot

protected [AbstractDocument.AbstractElement](http://docs.google.com/javax/swing/text/AbstractDocument.AbstractElement.html) **createDefaultRoot**()

Creates the root element to be used to represent the default document structure.

**Returns:**the element base

### getForeground

public [Color](http://docs.google.com/java/awt/Color.html) **getForeground**([AttributeSet](http://docs.google.com/javax/swing/text/AttributeSet.html) attr)

Gets the foreground color from an attribute set.

**Specified by:**[getForeground](http://docs.google.com/javax/swing/text/StyledDocument.html#getForeground(javax.swing.text.AttributeSet)) in interface [StyledDocument](http://docs.google.com/javax/swing/text/StyledDocument.html) **Parameters:**attr - the attribute set **Returns:**the color

### getBackground

public [Color](http://docs.google.com/java/awt/Color.html) **getBackground**([AttributeSet](http://docs.google.com/javax/swing/text/AttributeSet.html) attr)

Gets the background color from an attribute set.

**Specified by:**[getBackground](http://docs.google.com/javax/swing/text/StyledDocument.html#getBackground(javax.swing.text.AttributeSet)) in interface [StyledDocument](http://docs.google.com/javax/swing/text/StyledDocument.html) **Parameters:**attr - the attribute set **Returns:**the color

### getFont

public [Font](http://docs.google.com/java/awt/Font.html) **getFont**([AttributeSet](http://docs.google.com/javax/swing/text/AttributeSet.html) attr)

Gets the font from an attribute set.

**Specified by:**[getFont](http://docs.google.com/javax/swing/text/StyledDocument.html#getFont(javax.swing.text.AttributeSet)) in interface [StyledDocument](http://docs.google.com/javax/swing/text/StyledDocument.html) **Parameters:**attr - the attribute set **Returns:**the font

### styleChanged

protected void **styleChanged**([Style](http://docs.google.com/javax/swing/text/Style.html) style)

Called when any of this document's styles have changed. Subclasses may wish to be intelligent about what gets damaged.

**Parameters:**style - The Style that has changed.

### addDocumentListener

public void **addDocumentListener**([DocumentListener](http://docs.google.com/javax/swing/event/DocumentListener.html) listener)

Adds a document listener for notification of any changes.

**Specified by:**[addDocumentListener](http://docs.google.com/javax/swing/text/Document.html#addDocumentListener(javax.swing.event.DocumentListener)) in interface [Document](http://docs.google.com/javax/swing/text/Document.html)**Overrides:**[addDocumentListener](http://docs.google.com/javax/swing/text/AbstractDocument.html#addDocumentListener(javax.swing.event.DocumentListener)) in class [AbstractDocument](http://docs.google.com/javax/swing/text/AbstractDocument.html) **Parameters:**listener - the listener**See Also:**[Document.addDocumentListener(javax.swing.event.DocumentListener)](http://docs.google.com/javax/swing/text/Document.html#addDocumentListener(javax.swing.event.DocumentListener))

### removeDocumentListener

public void **removeDocumentListener**([DocumentListener](http://docs.google.com/javax/swing/event/DocumentListener.html) listener)

Removes a document listener.

**Specified by:**[removeDocumentListener](http://docs.google.com/javax/swing/text/Document.html#removeDocumentListener(javax.swing.event.DocumentListener)) in interface [Document](http://docs.google.com/javax/swing/text/Document.html)**Overrides:**[removeDocumentListener](http://docs.google.com/javax/swing/text/AbstractDocument.html#removeDocumentListener(javax.swing.event.DocumentListener)) in class [AbstractDocument](http://docs.google.com/javax/swing/text/AbstractDocument.html) **Parameters:**listener - the listener**See Also:**[Document.removeDocumentListener(javax.swing.event.DocumentListener)](http://docs.google.com/javax/swing/text/Document.html#removeDocumentListener(javax.swing.event.DocumentListener))

| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/DefaultStyledDocument.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/DefaultHighlighter.DefaultHighlightPainter.html)   [**NEXT CLASS**](http://docs.google.com/javax/swing/text/DefaultStyledDocument.AttributeUndoableEdit.html) | [**FRAMES**](http://docs.google.com/index.html?javax/swing/text/DefaultStyledDocument.html)    [**NO FRAMES**](http://docs.google.com/DefaultStyledDocument.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: [NESTED](#3znysh7) | [FIELD](#tyjcwt) | [CONSTR](#4d34og8) | [METHOD](#2s8eyo1) | DETAIL: [FIELD](#lnxbz9) | [CONSTR](#44sinio) | [METHOD](#1y810tw) |

[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).