| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/AbstractDocument.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   [**NEXT CLASS**](http://docs.google.com/javax/swing/text/AbstractDocument.AbstractElement.html) | [**FRAMES**](http://docs.google.com/index.html?javax/swing/text/AbstractDocument.html)    [**NO FRAMES**](http://docs.google.com/AbstractDocument.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: [NESTED](#3znysh7) | [FIELD](#2et92p0) | [CONSTR](#3dy6vkm) | [METHOD](#1t3h5sf) | DETAIL: [FIELD](#2s8eyo1) | [CONSTR](#2jxsxqh) | [METHOD](#1y810tw) |

## **javax.swing.text**

Class AbstractDocument

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

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

public abstract class **AbstractDocument**extends [Object](http://docs.google.com/java/lang/Object.html)implements [Document](http://docs.google.com/javax/swing/text/Document.html), [Serializable](http://docs.google.com/java/io/Serializable.html)

An implementation of the document interface to serve as a basis for implementing various kinds of documents. At this level there is very little policy, so there is a corresponding increase in difficulty of use.

This class implements a locking mechanism for the document. It allows multiple readers or one writer, and writers must wait until all observers of the document have been notified of a previous change before beginning another mutation to the document. The read lock is acquired and released using the render method. A write lock is aquired by the methods that mutate the document, and are held for the duration of the method call. Notification is done on the thread that produced the mutation, and the thread has full read access to the document for the duration of the notification, but other readers are kept out until the notification has finished. The notification is a beans event notification which does not allow any further mutations until all listeners have been notified.

Any models subclassed from this class and used in conjunction with a text component that has a look and feel implementation that is derived from BasicTextUI may be safely updated asynchronously, because all access to the View hierarchy is serialized by BasicTextUI if the document is of type AbstractDocument. The locking assumes that an independent thread will access the View hierarchy only from the DocumentListener methods, and that there will be only one event thread active at a time.

If concurrency support is desired, there are the following additional implications. The code path for any DocumentListener implementation and any UndoListener implementation must be threadsafe, and not access the component lock if trying to be safe from deadlocks. The repaint and revalidate methods on JComponent are safe.

AbstractDocument models an implied break at the end of the document. Among other things this allows you to position the caret after the last character. As a result of this, getLength returns one less than the length of the Content. If you create your own Content, be sure and initialize it to have an additional character. Refer to StringContent and GapContent for examples of this. Another implication of this is that Elements that model the implied end character will have an endOffset == (getLength() + 1). For example, in DefaultStyledDocument getParagraphElement(getLength()).getEndOffset() == getLength() + 1 .

**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).

| **Nested Class Summary** | |
| --- | --- |
| class | [**AbstractDocument.AbstractElement**](http://docs.google.com/javax/swing/text/AbstractDocument.AbstractElement.html)            Implements the abstract part of an element. |
| static interface | [**AbstractDocument.AttributeContext**](http://docs.google.com/javax/swing/text/AbstractDocument.AttributeContext.html)            An interface that can be used to allow MutableAttributeSet implementations to use pluggable attribute compression techniques. |
| class | [**AbstractDocument.BranchElement**](http://docs.google.com/javax/swing/text/AbstractDocument.BranchElement.html)            Implements a composite element that contains other elements. |
| static interface | [**AbstractDocument.Content**](http://docs.google.com/javax/swing/text/AbstractDocument.Content.html)            Interface to describe a sequence of character content that can be edited. |
| class | [**AbstractDocument.DefaultDocumentEvent**](http://docs.google.com/javax/swing/text/AbstractDocument.DefaultDocumentEvent.html)            Stores document changes as the document is being modified. |
| static class | [**AbstractDocument.ElementEdit**](http://docs.google.com/javax/swing/text/AbstractDocument.ElementEdit.html)            An implementation of ElementChange that can be added to the document event. |
| class | [**AbstractDocument.LeafElement**](http://docs.google.com/javax/swing/text/AbstractDocument.LeafElement.html)            Implements an element that directly represents content of some kind. |

| **Field Summary** | |
| --- | --- |
| protected static [String](http://docs.google.com/java/lang/String.html) | [**BAD\_LOCATION**](http://docs.google.com/javax/swing/text/AbstractDocument.html#BAD_LOCATION)            Error message to indicate a bad location. |
| static [String](http://docs.google.com/java/lang/String.html) | [**BidiElementName**](http://docs.google.com/javax/swing/text/AbstractDocument.html#BidiElementName)            Name of elements used to hold a unidirectional run |
| static [String](http://docs.google.com/java/lang/String.html) | [**ContentElementName**](http://docs.google.com/javax/swing/text/AbstractDocument.html#ContentElementName)            Name of elements used to represent content |
| static [String](http://docs.google.com/java/lang/String.html) | [**ElementNameAttribute**](http://docs.google.com/javax/swing/text/AbstractDocument.html#ElementNameAttribute)            Name of the attribute used to specify element names. |
| protected  [EventListenerList](http://docs.google.com/javax/swing/event/EventListenerList.html) | [**listenerList**](http://docs.google.com/javax/swing/text/AbstractDocument.html#listenerList)            The event listener list for the document. |
| static [String](http://docs.google.com/java/lang/String.html) | [**ParagraphElementName**](http://docs.google.com/javax/swing/text/AbstractDocument.html#ParagraphElementName)            Name of elements used to represent paragraphs |
| static [String](http://docs.google.com/java/lang/String.html) | [**SectionElementName**](http://docs.google.com/javax/swing/text/AbstractDocument.html#SectionElementName)            Name of elements used to hold sections (lines/paragraphs). |

| **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** | |
| --- | --- |
| protected | [**AbstractDocument**](http://docs.google.com/javax/swing/text/AbstractDocument.html#AbstractDocument(javax.swing.text.AbstractDocument.Content))([AbstractDocument.Content](http://docs.google.com/javax/swing/text/AbstractDocument.Content.html) data)            Constructs a new AbstractDocument, wrapped around some specified content storage mechanism. |
| protected | [**AbstractDocument**](http://docs.google.com/javax/swing/text/AbstractDocument.html#AbstractDocument(javax.swing.text.AbstractDocument.Content,%20javax.swing.text.AbstractDocument.AttributeContext))([AbstractDocument.Content](http://docs.google.com/javax/swing/text/AbstractDocument.Content.html) data, [AbstractDocument.AttributeContext](http://docs.google.com/javax/swing/text/AbstractDocument.AttributeContext.html) context)            Constructs a new AbstractDocument, wrapped around some specified content storage mechanism. |

| **Method Summary** | |
| --- | --- |
| void | [**addDocumentListener**](http://docs.google.com/javax/swing/text/AbstractDocument.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. |
| void | [**addUndoableEditListener**](http://docs.google.com/javax/swing/text/AbstractDocument.html#addUndoableEditListener(javax.swing.event.UndoableEditListener))([UndoableEditListener](http://docs.google.com/javax/swing/event/UndoableEditListener.html) listener)            Adds an undo listener for notification of any changes. |
| protected  [Element](http://docs.google.com/javax/swing/text/Element.html) | [**createBranchElement**](http://docs.google.com/javax/swing/text/AbstractDocument.html#createBranchElement(javax.swing.text.Element,%20javax.swing.text.AttributeSet))([Element](http://docs.google.com/javax/swing/text/Element.html) parent, [AttributeSet](http://docs.google.com/javax/swing/text/AttributeSet.html) a)            Creates a document branch element, that can contain other elements. |
| protected  [Element](http://docs.google.com/javax/swing/text/Element.html) | [**createLeafElement**](http://docs.google.com/javax/swing/text/AbstractDocument.html#createLeafElement(javax.swing.text.Element,%20javax.swing.text.AttributeSet,%20int,%20int))([Element](http://docs.google.com/javax/swing/text/Element.html) parent, [AttributeSet](http://docs.google.com/javax/swing/text/AttributeSet.html) a, int p0, int p1)            Creates a document leaf element. |
| [Position](http://docs.google.com/javax/swing/text/Position.html) | [**createPosition**](http://docs.google.com/javax/swing/text/AbstractDocument.html#createPosition(int))(int offs)            Returns a position that will track change as the document is altered. |
| void | [**dump**](http://docs.google.com/javax/swing/text/AbstractDocument.html#dump(java.io.PrintStream))([PrintStream](http://docs.google.com/java/io/PrintStream.html) out)            Gives a diagnostic dump. |
| protected  void | [**fireChangedUpdate**](http://docs.google.com/javax/swing/text/AbstractDocument.html#fireChangedUpdate(javax.swing.event.DocumentEvent))([DocumentEvent](http://docs.google.com/javax/swing/event/DocumentEvent.html) e)            Notifies all listeners that have registered interest for notification on this event type. |
| protected  void | [**fireInsertUpdate**](http://docs.google.com/javax/swing/text/AbstractDocument.html#fireInsertUpdate(javax.swing.event.DocumentEvent))([DocumentEvent](http://docs.google.com/javax/swing/event/DocumentEvent.html) e)            Notifies all listeners that have registered interest for notification on this event type. |
| protected  void | [**fireRemoveUpdate**](http://docs.google.com/javax/swing/text/AbstractDocument.html#fireRemoveUpdate(javax.swing.event.DocumentEvent))([DocumentEvent](http://docs.google.com/javax/swing/event/DocumentEvent.html) e)            Notifies all listeners that have registered interest for notification on this event type. |
| protected  void | [**fireUndoableEditUpdate**](http://docs.google.com/javax/swing/text/AbstractDocument.html#fireUndoableEditUpdate(javax.swing.event.UndoableEditEvent))([UndoableEditEvent](http://docs.google.com/javax/swing/event/UndoableEditEvent.html) e)            Notifies all listeners that have registered interest for notification on this event type. |
| int | [**getAsynchronousLoadPriority**](http://docs.google.com/javax/swing/text/AbstractDocument.html#getAsynchronousLoadPriority())()            Gets the asynchronous loading priority. |
| protected  [AbstractDocument.AttributeContext](http://docs.google.com/javax/swing/text/AbstractDocument.AttributeContext.html) | [**getAttributeContext**](http://docs.google.com/javax/swing/text/AbstractDocument.html#getAttributeContext())()            Fetches the context for managing attributes. |
| [Element](http://docs.google.com/javax/swing/text/Element.html) | [**getBidiRootElement**](http://docs.google.com/javax/swing/text/AbstractDocument.html#getBidiRootElement())()            Returns the root element of the bidirectional structure for this document. |
| protected  [AbstractDocument.Content](http://docs.google.com/javax/swing/text/AbstractDocument.Content.html) | [**getContent**](http://docs.google.com/javax/swing/text/AbstractDocument.html#getContent())()            Gets the content for the document. |
| protected  [Thread](http://docs.google.com/java/lang/Thread.html) | [**getCurrentWriter**](http://docs.google.com/javax/swing/text/AbstractDocument.html#getCurrentWriter())()            Fetches the current writing thread if there is one. |
| abstract  [Element](http://docs.google.com/javax/swing/text/Element.html) | [**getDefaultRootElement**](http://docs.google.com/javax/swing/text/AbstractDocument.html#getDefaultRootElement())()            Returns the root element that views should be based upon unless some other mechanism for assigning views to element structures is provided. |
| [DocumentFilter](http://docs.google.com/javax/swing/text/DocumentFilter.html) | [**getDocumentFilter**](http://docs.google.com/javax/swing/text/AbstractDocument.html#getDocumentFilter())()            Returns the DocumentFilter that is responsible for filtering of insertion/removal. |
| [DocumentListener](http://docs.google.com/javax/swing/event/DocumentListener.html)[] | [**getDocumentListeners**](http://docs.google.com/javax/swing/text/AbstractDocument.html#getDocumentListeners())()            Returns an array of all the document listeners registered on this document. |
| [Dictionary](http://docs.google.com/java/util/Dictionary.html)<[Object](http://docs.google.com/java/lang/Object.html),[Object](http://docs.google.com/java/lang/Object.html)> | [**getDocumentProperties**](http://docs.google.com/javax/swing/text/AbstractDocument.html#getDocumentProperties())()            Supports managing a set of properties. |
| [Position](http://docs.google.com/javax/swing/text/Position.html) | [**getEndPosition**](http://docs.google.com/javax/swing/text/AbstractDocument.html#getEndPosition())()            Returns a position that represents the end of the document. |
| int | [**getLength**](http://docs.google.com/javax/swing/text/AbstractDocument.html#getLength())()            Returns the length of the data. |
| | <T extends [EventListener](http://docs.google.com/java/util/EventListener.html)>  T[] | | --- | | [**getListeners**](http://docs.google.com/javax/swing/text/AbstractDocument.html#getListeners(java.lang.Class))([Class](http://docs.google.com/java/lang/Class.html)<T> listenerType)            Returns an array of all the objects currently registered as *Foo*Listeners upon this document. |
| abstract  [Element](http://docs.google.com/javax/swing/text/Element.html) | [**getParagraphElement**](http://docs.google.com/javax/swing/text/AbstractDocument.html#getParagraphElement(int))(int pos)            Get the paragraph element containing the given position. |
| [Object](http://docs.google.com/java/lang/Object.html) | [**getProperty**](http://docs.google.com/javax/swing/text/AbstractDocument.html#getProperty(java.lang.Object))([Object](http://docs.google.com/java/lang/Object.html) key)            A convenience method for looking up a property value. |
| [Element](http://docs.google.com/javax/swing/text/Element.html)[] | [**getRootElements**](http://docs.google.com/javax/swing/text/AbstractDocument.html#getRootElements())()            Gets all root elements defined. |
| [Position](http://docs.google.com/javax/swing/text/Position.html) | [**getStartPosition**](http://docs.google.com/javax/swing/text/AbstractDocument.html#getStartPosition())()            Returns a position that represents the start of the document. |
| [String](http://docs.google.com/java/lang/String.html) | [**getText**](http://docs.google.com/javax/swing/text/AbstractDocument.html#getText(int,%20int))(int offset, int length)            Gets a sequence of text from the document. |
| void | [**getText**](http://docs.google.com/javax/swing/text/AbstractDocument.html#getText(int,%20int,%20javax.swing.text.Segment))(int offset, int length, [Segment](http://docs.google.com/javax/swing/text/Segment.html) txt)            Fetches the text contained within the given portion of the document. |
| [UndoableEditListener](http://docs.google.com/javax/swing/event/UndoableEditListener.html)[] | [**getUndoableEditListeners**](http://docs.google.com/javax/swing/text/AbstractDocument.html#getUndoableEditListeners())()            Returns an array of all the undoable edit listeners registered on this document. |
| void | [**insertString**](http://docs.google.com/javax/swing/text/AbstractDocument.html#insertString(int,%20java.lang.String,%20javax.swing.text.AttributeSet))(int offs, [String](http://docs.google.com/java/lang/String.html) str, [AttributeSet](http://docs.google.com/javax/swing/text/AttributeSet.html) a)            Inserts some content into the document. |
| protected  void | [**insertUpdate**](http://docs.google.com/javax/swing/text/AbstractDocument.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. |
| protected  void | [**postRemoveUpdate**](http://docs.google.com/javax/swing/text/AbstractDocument.html#postRemoveUpdate(javax.swing.text.AbstractDocument.DefaultDocumentEvent))([AbstractDocument.DefaultDocumentEvent](http://docs.google.com/javax/swing/text/AbstractDocument.DefaultDocumentEvent.html) chng)            Updates any document structure as a result of text removal. |
| void | [**putProperty**](http://docs.google.com/javax/swing/text/AbstractDocument.html#putProperty(java.lang.Object,%20java.lang.Object))([Object](http://docs.google.com/java/lang/Object.html) key, [Object](http://docs.google.com/java/lang/Object.html) value)            A convenience method for storing up a property value. |
| void | [**readLock**](http://docs.google.com/javax/swing/text/AbstractDocument.html#readLock())()            Acquires a lock to begin reading some state from the document. |
| void | [**readUnlock**](http://docs.google.com/javax/swing/text/AbstractDocument.html#readUnlock())()            Does a read unlock. |
| void | [**remove**](http://docs.google.com/javax/swing/text/AbstractDocument.html#remove(int,%20int))(int offs, int len)            Removes some content from the document. |
| void | [**removeDocumentListener**](http://docs.google.com/javax/swing/text/AbstractDocument.html#removeDocumentListener(javax.swing.event.DocumentListener))([DocumentListener](http://docs.google.com/javax/swing/event/DocumentListener.html) listener)            Removes a document listener. |
| void | [**removeUndoableEditListener**](http://docs.google.com/javax/swing/text/AbstractDocument.html#removeUndoableEditListener(javax.swing.event.UndoableEditListener))([UndoableEditListener](http://docs.google.com/javax/swing/event/UndoableEditListener.html) listener)            Removes an undo listener. |
| protected  void | [**removeUpdate**](http://docs.google.com/javax/swing/text/AbstractDocument.html#removeUpdate(javax.swing.text.AbstractDocument.DefaultDocumentEvent))([AbstractDocument.DefaultDocumentEvent](http://docs.google.com/javax/swing/text/AbstractDocument.DefaultDocumentEvent.html) chng)            Updates any document structure as a result of text removal. |
| void | [**render**](http://docs.google.com/javax/swing/text/AbstractDocument.html#render(java.lang.Runnable))([Runnable](http://docs.google.com/java/lang/Runnable.html) r)            This allows the model to be safely rendered in the presence of currency, if the model supports being updated asynchronously. |
| void | [**replace**](http://docs.google.com/javax/swing/text/AbstractDocument.html#replace(int,%20int,%20java.lang.String,%20javax.swing.text.AttributeSet))(int offset, int length, [String](http://docs.google.com/java/lang/String.html) text, [AttributeSet](http://docs.google.com/javax/swing/text/AttributeSet.html) attrs)            Deletes the region of text from offset to offset + length, and replaces it with text. |
| void | [**setAsynchronousLoadPriority**](http://docs.google.com/javax/swing/text/AbstractDocument.html#setAsynchronousLoadPriority(int))(int p)            Sets the asynchronous loading priority. |
| void | [**setDocumentFilter**](http://docs.google.com/javax/swing/text/AbstractDocument.html#setDocumentFilter(javax.swing.text.DocumentFilter))([DocumentFilter](http://docs.google.com/javax/swing/text/DocumentFilter.html) filter)            Sets the DocumentFilter. |
| void | [**setDocumentProperties**](http://docs.google.com/javax/swing/text/AbstractDocument.html#setDocumentProperties(java.util.Dictionary))([Dictionary](http://docs.google.com/java/util/Dictionary.html)<[Object](http://docs.google.com/java/lang/Object.html),[Object](http://docs.google.com/java/lang/Object.html)> x)            Replaces the document properties dictionary for this document. |
| protected  void | [**writeLock**](http://docs.google.com/javax/swing/text/AbstractDocument.html#writeLock())()            Acquires a lock to begin mutating the document this lock protects. |
| protected  void | [**writeUnlock**](http://docs.google.com/javax/swing/text/AbstractDocument.html#writeUnlock())()            Releases a write lock previously obtained via writeLock. |

| **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)) |

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

### listenerList

protected [EventListenerList](http://docs.google.com/javax/swing/event/EventListenerList.html) **listenerList**

The event listener list for the document.

### BAD\_LOCATION

protected static final [String](http://docs.google.com/java/lang/String.html) **BAD\_LOCATION**

Error message to indicate a bad location.

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

### ParagraphElementName

public static final [String](http://docs.google.com/java/lang/String.html) **ParagraphElementName**

Name of elements used to represent paragraphs

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

### ContentElementName

public static final [String](http://docs.google.com/java/lang/String.html) **ContentElementName**

Name of elements used to represent content

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

### SectionElementName

public static final [String](http://docs.google.com/java/lang/String.html) **SectionElementName**

Name of elements used to hold sections (lines/paragraphs).

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

### BidiElementName

public static final [String](http://docs.google.com/java/lang/String.html) **BidiElementName**

Name of elements used to hold a unidirectional run

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

### ElementNameAttribute

public static final [String](http://docs.google.com/java/lang/String.html) **ElementNameAttribute**

Name of the attribute used to specify element names.

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

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

### AbstractDocument

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

Constructs a new AbstractDocument, wrapped around some specified content storage mechanism.

**Parameters:**data - the content

### AbstractDocument

protected **AbstractDocument**([AbstractDocument.Content](http://docs.google.com/javax/swing/text/AbstractDocument.Content.html) data,  
 [AbstractDocument.AttributeContext](http://docs.google.com/javax/swing/text/AbstractDocument.AttributeContext.html) context)

Constructs a new AbstractDocument, wrapped around some specified content storage mechanism.

**Parameters:**data - the contentcontext - the attribute context

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

### getDocumentProperties

public [Dictionary](http://docs.google.com/java/util/Dictionary.html)<[Object](http://docs.google.com/java/lang/Object.html),[Object](http://docs.google.com/java/lang/Object.html)> **getDocumentProperties**()

Supports managing a set of properties. Callers can use the documentProperties dictionary to annotate the document with document-wide properties.

**Returns:**a non-null Dictionary**See Also:**[setDocumentProperties(java.util.Dictionary)](http://docs.google.com/javax/swing/text/AbstractDocument.html#setDocumentProperties(java.util.Dictionary))

### setDocumentProperties

public void **setDocumentProperties**([Dictionary](http://docs.google.com/java/util/Dictionary.html)<[Object](http://docs.google.com/java/lang/Object.html),[Object](http://docs.google.com/java/lang/Object.html)> x)

Replaces the document properties dictionary for this document.

**Parameters:**x - the new dictionary**See Also:**[getDocumentProperties()](http://docs.google.com/javax/swing/text/AbstractDocument.html#getDocumentProperties())

### fireInsertUpdate

protected void **fireInsertUpdate**([DocumentEvent](http://docs.google.com/javax/swing/event/DocumentEvent.html) e)

Notifies all listeners that have registered interest for notification on this event type. The event instance is lazily created using the parameters passed into the fire method.

**Parameters:**e - the event**See Also:**[EventListenerList](http://docs.google.com/javax/swing/event/EventListenerList.html)

### fireChangedUpdate

protected void **fireChangedUpdate**([DocumentEvent](http://docs.google.com/javax/swing/event/DocumentEvent.html) e)

Notifies all listeners that have registered interest for notification on this event type. The event instance is lazily created using the parameters passed into the fire method.

**Parameters:**e - the event**See Also:**[EventListenerList](http://docs.google.com/javax/swing/event/EventListenerList.html)

### fireRemoveUpdate

protected void **fireRemoveUpdate**([DocumentEvent](http://docs.google.com/javax/swing/event/DocumentEvent.html) e)

Notifies all listeners that have registered interest for notification on this event type. The event instance is lazily created using the parameters passed into the fire method.

**Parameters:**e - the event**See Also:**[EventListenerList](http://docs.google.com/javax/swing/event/EventListenerList.html)

### fireUndoableEditUpdate

protected void **fireUndoableEditUpdate**([UndoableEditEvent](http://docs.google.com/javax/swing/event/UndoableEditEvent.html) e)

Notifies all listeners that have registered interest for notification on this event type. The event instance is lazily created using the parameters passed into the fire method.

**Parameters:**e - the event**See Also:**[EventListenerList](http://docs.google.com/javax/swing/event/EventListenerList.html)

### getListeners

public <T extends [EventListener](http://docs.google.com/java/util/EventListener.html)> T[] **getListeners**([Class](http://docs.google.com/java/lang/Class.html)<T> listenerType)

Returns an array of all the objects currently registered as *Foo*Listeners upon this document. *Foo*Listeners are registered using the add*Foo*Listener method.

You can specify the listenerType argument with a class literal, such as *Foo*Listener.class. For example, you can query a document d for its document listeners with the following code:

DocumentListener[] mls = (DocumentListener[])(d.getListeners(DocumentListener.class));

If no such listeners exist, this method returns an empty array.

**Parameters:**listenerType - the type of listeners requested; this parameter should specify an interface that descends from java.util.EventListener **Returns:**an array of all objects registered as *Foo*Listeners on this component, or an empty array if no such listeners have been added **Throws:** [ClassCastException](http://docs.google.com/java/lang/ClassCastException.html) - if listenerType doesn't specify a class or interface that implements java.util.EventListener**Since:** 1.3 **See Also:**[getDocumentListeners()](http://docs.google.com/javax/swing/text/AbstractDocument.html#getDocumentListeners()), [getUndoableEditListeners()](http://docs.google.com/javax/swing/text/AbstractDocument.html#getUndoableEditListeners())

### getAsynchronousLoadPriority

public int **getAsynchronousLoadPriority**()

Gets the asynchronous loading priority. If less than zero, the document should not be loaded asynchronously.

**Returns:**the asynchronous loading priority, or -1 if the document should not be loaded asynchronously

### setAsynchronousLoadPriority

public void **setAsynchronousLoadPriority**(int p)

Sets the asynchronous loading priority.

**Parameters:**p - the new asynchronous loading priority; a value less than zero indicates that the document should not be loaded asynchronously

### setDocumentFilter

public void **setDocumentFilter**([DocumentFilter](http://docs.google.com/javax/swing/text/DocumentFilter.html) filter)

Sets the DocumentFilter. The DocumentFilter is passed insert and remove to conditionally allow inserting/deleting of the text. A null value indicates that no filtering will occur.

**Parameters:**filter - the DocumentFilter used to constrain text**Since:** 1.4 **See Also:**[getDocumentFilter()](http://docs.google.com/javax/swing/text/AbstractDocument.html#getDocumentFilter())

### getDocumentFilter

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

Returns the DocumentFilter that is responsible for filtering of insertion/removal. A null return value implies no filtering is to occur.

**Returns:**the DocumentFilter**Since:** 1.4 **See Also:**[setDocumentFilter(javax.swing.text.DocumentFilter)](http://docs.google.com/javax/swing/text/AbstractDocument.html#setDocumentFilter(javax.swing.text.DocumentFilter))

### render

public void **render**([Runnable](http://docs.google.com/java/lang/Runnable.html) r)

This allows the model to be safely rendered in the presence of currency, if the model supports being updated asynchronously. The given runnable will be executed in a way that allows it to safely read the model with no changes while the runnable is being executed. The runnable itself may *not* make any mutations.

This is implemented to aquire a read lock for the duration of the runnables execution. There may be multiple runnables executing at the same time, and all writers will be blocked while there are active rendering runnables. If the runnable throws an exception, its lock will be safely released. There is no protection against a runnable that never exits, which will effectively leave the document locked for it's lifetime.

If the given runnable attempts to make any mutations in this implementation, a deadlock will occur. There is no tracking of individual rendering threads to enable detecting this situation, but a subclass could incur the overhead of tracking them and throwing an error.

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:**[render](http://docs.google.com/javax/swing/text/Document.html#render(java.lang.Runnable)) in interface [Document](http://docs.google.com/javax/swing/text/Document.html) **Parameters:**r - the renderer to execute

### getLength

public int **getLength**()

Returns the length of the data. This is the number of characters of content that represents the users data.

**Specified by:**[getLength](http://docs.google.com/javax/swing/text/Document.html#getLength()) in interface [Document](http://docs.google.com/javax/swing/text/Document.html) **Returns:**the length >= 0**See Also:**[Document.getLength()](http://docs.google.com/javax/swing/text/Document.html#getLength())

### 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) **Parameters:**listener - the DocumentListener to add**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) **Parameters:**listener - the DocumentListener to remove**See Also:**[Document.removeDocumentListener(javax.swing.event.DocumentListener)](http://docs.google.com/javax/swing/text/Document.html#removeDocumentListener(javax.swing.event.DocumentListener))

### getDocumentListeners

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

Returns an array of all the document listeners registered on this document.

**Returns:**all of this document's DocumentListeners or an empty array if no document listeners are currently registered**Since:** 1.4 **See Also:**[addDocumentListener(javax.swing.event.DocumentListener)](http://docs.google.com/javax/swing/text/AbstractDocument.html#addDocumentListener(javax.swing.event.DocumentListener)), [removeDocumentListener(javax.swing.event.DocumentListener)](http://docs.google.com/javax/swing/text/AbstractDocument.html#removeDocumentListener(javax.swing.event.DocumentListener))

### addUndoableEditListener

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

Adds an undo listener for notification of any changes. Undo/Redo operations performed on the UndoableEdit will cause the appropriate DocumentEvent to be fired to keep the view(s) in sync with the model.

**Specified by:**[addUndoableEditListener](http://docs.google.com/javax/swing/text/Document.html#addUndoableEditListener(javax.swing.event.UndoableEditListener)) in interface [Document](http://docs.google.com/javax/swing/text/Document.html) **Parameters:**listener - the UndoableEditListener to add**See Also:**[Document.addUndoableEditListener(javax.swing.event.UndoableEditListener)](http://docs.google.com/javax/swing/text/Document.html#addUndoableEditListener(javax.swing.event.UndoableEditListener))

### removeUndoableEditListener

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

Removes an undo listener.

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

### getUndoableEditListeners

public [UndoableEditListener](http://docs.google.com/javax/swing/event/UndoableEditListener.html)[] **getUndoableEditListeners**()

Returns an array of all the undoable edit listeners registered on this document.

**Returns:**all of this document's UndoableEditListeners or an empty array if no undoable edit listeners are currently registered**Since:** 1.4 **See Also:**[addUndoableEditListener(javax.swing.event.UndoableEditListener)](http://docs.google.com/javax/swing/text/AbstractDocument.html#addUndoableEditListener(javax.swing.event.UndoableEditListener)), [removeUndoableEditListener(javax.swing.event.UndoableEditListener)](http://docs.google.com/javax/swing/text/AbstractDocument.html#removeUndoableEditListener(javax.swing.event.UndoableEditListener))

### getProperty

public final [Object](http://docs.google.com/java/lang/Object.html) **getProperty**([Object](http://docs.google.com/java/lang/Object.html) key)

A convenience method for looking up a property value. It is equivalent to:

getDocumentProperties().get(key);

**Specified by:**[getProperty](http://docs.google.com/javax/swing/text/Document.html#getProperty(java.lang.Object)) in interface [Document](http://docs.google.com/javax/swing/text/Document.html) **Parameters:**key - the non-null property key **Returns:**the value of this property or null**See Also:**[getDocumentProperties()](http://docs.google.com/javax/swing/text/AbstractDocument.html#getDocumentProperties())

### putProperty

public final void **putProperty**([Object](http://docs.google.com/java/lang/Object.html) key,  
 [Object](http://docs.google.com/java/lang/Object.html) value)

A convenience method for storing up a property value. It is equivalent to:

getDocumentProperties().put(key, value);

If value is null this method will remove the property.

**Specified by:**[putProperty](http://docs.google.com/javax/swing/text/Document.html#putProperty(java.lang.Object,%20java.lang.Object)) in interface [Document](http://docs.google.com/javax/swing/text/Document.html) **Parameters:**key - the non-null keyvalue - the property value**See Also:**[getDocumentProperties()](http://docs.google.com/javax/swing/text/AbstractDocument.html#getDocumentProperties())

### remove

public void **remove**(int offs,  
 int len)  
 throws [BadLocationException](http://docs.google.com/javax/swing/text/BadLocationException.html)

Removes some content from the document. Removing content causes a write lock to be held while the actual changes are taking place. Observers are notified of the change on the thread that called this method.

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:**[remove](http://docs.google.com/javax/swing/text/Document.html#remove(int,%20int)) in interface [Document](http://docs.google.com/javax/swing/text/Document.html) **Parameters:**offs - the starting offset >= 0len - the number of characters to remove >= 0 **Throws:** [BadLocationException](http://docs.google.com/javax/swing/text/BadLocationException.html) - the given remove position is not a valid position within the document**See Also:**[Document.remove(int, int)](http://docs.google.com/javax/swing/text/Document.html#remove(int,%20int))

### replace

public void **replace**(int offset,  
 int length,  
 [String](http://docs.google.com/java/lang/String.html) text,  
 [AttributeSet](http://docs.google.com/javax/swing/text/AttributeSet.html) attrs)  
 throws [BadLocationException](http://docs.google.com/javax/swing/text/BadLocationException.html)

Deletes the region of text from offset to offset + length, and replaces it with text. It is up to the implementation as to how this is implemented, some implementations may treat this as two distinct operations: a remove followed by an insert, others may treat the replace as one atomic operation.

**Parameters:**offset - index of child elementlength - length of text to delete, may be 0 indicating don't delete anythingtext - text to insert, null indicates no text to insertattrs - AttributeSet indicating attributes of inserted text, null is legal, and typically treated as an empty attributeset, but exact interpretation is left to the subclass **Throws:** [BadLocationException](http://docs.google.com/javax/swing/text/BadLocationException.html) - the given position is not a valid position within the document**Since:** 1.4

### insertString

public void **insertString**(int offs,  
 [String](http://docs.google.com/java/lang/String.html) str,  
 [AttributeSet](http://docs.google.com/javax/swing/text/AttributeSet.html) a)  
 throws [BadLocationException](http://docs.google.com/javax/swing/text/BadLocationException.html)

Inserts some content into the document. Inserting content causes a write lock to be held while the actual changes are taking place, followed by notification to the observers on the thread that grabbed the write lock.

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:**[insertString](http://docs.google.com/javax/swing/text/Document.html#insertString(int,%20java.lang.String,%20javax.swing.text.AttributeSet)) in interface [Document](http://docs.google.com/javax/swing/text/Document.html) **Parameters:**offs - the starting offset >= 0str - the string to insert; does nothing with null/empty stringsa - the attributes for the inserted content **Throws:** [BadLocationException](http://docs.google.com/javax/swing/text/BadLocationException.html) - the given insert position is not a valid position within the document**See Also:**[Document.insertString(int, java.lang.String, javax.swing.text.AttributeSet)](http://docs.google.com/javax/swing/text/Document.html#insertString(int,%20java.lang.String,%20javax.swing.text.AttributeSet))

### getText

public [String](http://docs.google.com/java/lang/String.html) **getText**(int offset,  
 int length)  
 throws [BadLocationException](http://docs.google.com/javax/swing/text/BadLocationException.html)

Gets a sequence of text from the document.

**Specified by:**[getText](http://docs.google.com/javax/swing/text/Document.html#getText(int,%20int)) in interface [Document](http://docs.google.com/javax/swing/text/Document.html) **Parameters:**offset - the starting offset >= 0length - the number of characters to retrieve >= 0 **Returns:**the text **Throws:** [BadLocationException](http://docs.google.com/javax/swing/text/BadLocationException.html) - the range given includes a position that is not a valid position within the document**See Also:**[Document.getText(int, int)](http://docs.google.com/javax/swing/text/Document.html#getText(int,%20int))

### getText

public void **getText**(int offset,  
 int length,  
 [Segment](http://docs.google.com/javax/swing/text/Segment.html) txt)  
 throws [BadLocationException](http://docs.google.com/javax/swing/text/BadLocationException.html)

Fetches the text contained within the given portion of the document.

If the partialReturn property on the txt parameter is false, the data returned in the Segment will be the entire length requested and may or may not be a copy depending upon how the data was stored. If the partialReturn property is true, only the amount of text that can be returned without creating a copy is returned. Using partial returns will give better performance for situations where large parts of the document are being scanned. The following is an example of using the partial return to access the entire document:

  int nleft = doc.getDocumentLength();  
   Segment text = new Segment();  
   int offs = 0;  
   text.setPartialReturn(true);   
   while (nleft > 0) {  
   doc.getText(offs, nleft, text);  
   // do something with text  
   nleft -= text.count;  
   offs += text.count;  
   }

**Specified by:**[getText](http://docs.google.com/javax/swing/text/Document.html#getText(int,%20int,%20javax.swing.text.Segment)) in interface [Document](http://docs.google.com/javax/swing/text/Document.html) **Parameters:**offset - the starting offset >= 0length - the number of characters to retrieve >= 0txt - the Segment object to retrieve the text into **Throws:** [BadLocationException](http://docs.google.com/javax/swing/text/BadLocationException.html) - the range given includes a position that is not a valid position within the document

### createPosition

public [Position](http://docs.google.com/javax/swing/text/Position.html) **createPosition**(int offs)  
 throws [BadLocationException](http://docs.google.com/javax/swing/text/BadLocationException.html)

Returns a position that will track change as the document is altered.

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:**[createPosition](http://docs.google.com/javax/swing/text/Document.html#createPosition(int)) in interface [Document](http://docs.google.com/javax/swing/text/Document.html) **Parameters:**offs - the position in the model >= 0 **Returns:**the 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:**[Document.createPosition(int)](http://docs.google.com/javax/swing/text/Document.html#createPosition(int))

### getStartPosition

public final [Position](http://docs.google.com/javax/swing/text/Position.html) **getStartPosition**()

Returns a position that represents the start of the document. The position returned can be counted on to track change and stay located at the beginning of the document.

**Specified by:**[getStartPosition](http://docs.google.com/javax/swing/text/Document.html#getStartPosition()) in interface [Document](http://docs.google.com/javax/swing/text/Document.html) **Returns:**the position

### getEndPosition

public final [Position](http://docs.google.com/javax/swing/text/Position.html) **getEndPosition**()

Returns a position that represents the end of the document. The position returned can be counted on to track change and stay located at the end of the document.

**Specified by:**[getEndPosition](http://docs.google.com/javax/swing/text/Document.html#getEndPosition()) in interface [Document](http://docs.google.com/javax/swing/text/Document.html) **Returns:**the position

### getRootElements

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

Gets all root elements defined. Typically, there will only be one so the default implementation is to return the default root element.

**Specified by:**[getRootElements](http://docs.google.com/javax/swing/text/Document.html#getRootElements()) in interface [Document](http://docs.google.com/javax/swing/text/Document.html) **Returns:**the root element

### getDefaultRootElement

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

Returns the root element that views should be based upon unless some other mechanism for assigning views to element structures is provided.

**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) **Returns:**the root element**See Also:**[Document.getDefaultRootElement()](http://docs.google.com/javax/swing/text/Document.html#getDefaultRootElement())

### getBidiRootElement

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

Returns the root element of the bidirectional structure for this document. Its children represent character runs with a given Unicode bidi level.

### getParagraphElement

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

Get the paragraph element containing the given position. Sub-classes must define for themselves what exactly constitutes a paragraph. They should keep in mind however that a paragraph should at least be the unit of text over which to run the Unicode bidirectional algorithm.

**Parameters:**pos - the starting offset >= 0 **Returns:**the element

### getAttributeContext

protected final [AbstractDocument.AttributeContext](http://docs.google.com/javax/swing/text/AbstractDocument.AttributeContext.html) **getAttributeContext**()

Fetches the context for managing attributes. This method effectively establishes the strategy used for compressing AttributeSet information.

**Returns:**the context

### 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. If a subclass of this class reimplements this method, it should delegate to the superclass as well.

**Parameters:**chng - a description of the changeattr - the attributes for the change

### removeUpdate

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

Updates any document structure as a result of text removal. This method is called before the text is actually removed from the Content. This will happen within a write lock. If a subclass of this class reimplements this method, it should delegate to the superclass as well.

**Parameters:**chng - a description of the change

### postRemoveUpdate

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

Updates any document structure as a result of text removal. This method is called after the text has been removed from the Content. This will happen within a write lock. If a subclass of this class reimplements this method, it should delegate to the superclass as well.

**Parameters:**chng - a description of the change

### dump

public void **dump**([PrintStream](http://docs.google.com/java/io/PrintStream.html) out)

Gives a diagnostic dump.

**Parameters:**out - the output stream

### getContent

protected final [AbstractDocument.Content](http://docs.google.com/javax/swing/text/AbstractDocument.Content.html) **getContent**()

Gets the content for the document.

**Returns:**the content

### createLeafElement

protected [Element](http://docs.google.com/javax/swing/text/Element.html) **createLeafElement**([Element](http://docs.google.com/javax/swing/text/Element.html) parent,  
 [AttributeSet](http://docs.google.com/javax/swing/text/AttributeSet.html) a,  
 int p0,  
 int p1)

Creates a document leaf element. Hook through which elements are created to represent the document structure. Because this implementation keeps structure and content separate, elements grow automatically when content is extended so splits of existing elements follow. The document itself gets to decide how to generate elements to give flexibility in the type of elements used.

**Parameters:**parent - the parent elementa - the attributes for the elementp0 - the beginning of the range >= 0p1 - the end of the range >= p0 **Returns:**the new element

### createBranchElement

protected [Element](http://docs.google.com/javax/swing/text/Element.html) **createBranchElement**([Element](http://docs.google.com/javax/swing/text/Element.html) parent,  
 [AttributeSet](http://docs.google.com/javax/swing/text/AttributeSet.html) a)

Creates a document branch element, that can contain other elements.

**Parameters:**parent - the parent elementa - the attributes **Returns:**the element

### getCurrentWriter

protected final [Thread](http://docs.google.com/java/lang/Thread.html) **getCurrentWriter**()

Fetches the current writing thread if there is one. This can be used to distinguish whether a method is being called as part of an existing modification or if a lock needs to be acquired and a new transaction started.

**Returns:**the thread actively modifying the document or null if there are no modifications in progress

### writeLock

protected final void **writeLock**()

Acquires a lock to begin mutating the document this lock protects. There can be no writing, notification of changes, or reading going on in order to gain the lock. Additionally a thread is allowed to gain more than one writeLock, as long as it doesn't attempt to gain additional writeLocks from within document notification. Attempting to gain a writeLock from within a DocumentListener notification will result in an IllegalStateException. The ability to obtain more than one writeLock per thread allows subclasses to gain a writeLock, perform a number of operations, then release the lock.

Calls to writeLock must be balanced with calls to writeUnlock, else the Document will be left in a locked state so that no reading or writing can be done.

**Throws:** [IllegalStateException](http://docs.google.com/java/lang/IllegalStateException.html) - thrown on illegal lock attempt. If the document is implemented properly, this can only happen if a document listener attempts to mutate the document. This situation violates the bean event model where order of delivery is not guaranteed and all listeners should be notified before further mutations are allowed.

### writeUnlock

protected final void **writeUnlock**()

Releases a write lock previously obtained via writeLock. After decrementing the lock count if there are no oustanding locks this will allow a new writer, or readers.

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

### readLock

public final void **readLock**()

Acquires a lock to begin reading some state from the document. There can be multiple readers at the same time. Writing blocks the readers until notification of the change to the listeners has been completed. This method should be used very carefully to avoid unintended compromise of the document. It should always be balanced with a readUnlock.

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

### readUnlock

public final void **readUnlock**()

Does a read unlock. This signals that one of the readers is done. If there are no more readers then writing can begin again. This should be balanced with a readLock, and should occur in a finally statement so that the balance is guaranteed. The following is an example.

  readLock();  
   try {  
   // do something  
   } finally {  
   readUnlock();  
   }

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

| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/AbstractDocument.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   [**NEXT CLASS**](http://docs.google.com/javax/swing/text/AbstractDocument.AbstractElement.html) | [**FRAMES**](http://docs.google.com/index.html?javax/swing/text/AbstractDocument.html)    [**NO FRAMES**](http://docs.google.com/AbstractDocument.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: [NESTED](#3znysh7) | [FIELD](#2et92p0) | [CONSTR](#3dy6vkm) | [METHOD](#1t3h5sf) | DETAIL: [FIELD](#2s8eyo1) | [CONSTR](#2jxsxqh) | [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).