| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/HTMLDocument.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/html/HTML.UnknownTag.html)   [**NEXT CLASS**](http://docs.google.com/javax/swing/text/html/HTMLDocument.BlockElement.html) | [**FRAMES**](http://docs.google.com/index.html?javax/swing/text/html/HTMLDocument.html)    [**NO FRAMES**](http://docs.google.com/HTMLDocument.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: [NESTED](#3znysh7) | [FIELD](#3dy6vkm) | [CONSTR](#17dp8vu) | [METHOD](#3rdcrjn) | DETAIL: [FIELD](#44sinio) | [CONSTR](#z337ya) | [METHOD](#2xcytpi) |

## **javax.swing.text.html**

Class HTMLDocument

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

**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)

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

A document that models HTML. The purpose of this model is to support both browsing and editing. As a result, the structure described by an HTML document is not exactly replicated by default. The element structure that is modeled by default, is built by the class HTMLDocument.HTMLReader, which implements the HTMLEditorKit.ParserCallback protocol that the parser expects. To change the structure one can subclass HTMLReader, and reimplement the method [getReader(int)](http://docs.google.com/javax/swing/text/html/HTMLDocument.html#getReader(int)) to return the new reader implementation. The documentation for HTMLReader should be consulted for the details of the default structure created. The intent is that the document be non-lossy (although reproducing the HTML format may result in a different format).

The document models only HTML, and makes no attempt to store view attributes in it. The elements are identified by the StyleContext.NameAttribute attribute, which should always have a value of type HTML.Tag that identifies the kind of element. Some of the elements (such as comments) are synthesized. The HTMLFactory uses this attribute to determine what kind of view to build.

This document supports incremental loading. The TokenThreshold property controls how much of the parse is buffered before trying to update the element structure of the document. This property is set by the EditorKit so that subclasses can disable it.

The Base property determines the URL against which relative URLs are resolved. By default, this will be the Document.StreamDescriptionProperty if the value of the property is a URL. If a <BASE> tag is encountered, the base will become the URL specified by that tag. Because the base URL is a property, it can of course be set directly.

The default content storage mechanism for this document is a gap buffer (GapContent). Alternatives can be supplied by using the constructor that takes a Content implementation.

**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 | [**HTMLDocument.BlockElement**](http://docs.google.com/javax/swing/text/html/HTMLDocument.BlockElement.html)            An element that represents a structural *block* of HTML. |
| class | [**HTMLDocument.HTMLReader**](http://docs.google.com/javax/swing/text/html/HTMLDocument.HTMLReader.html)            An HTML reader to load an HTML document with an HTML element structure. |
| static class | [**HTMLDocument.Iterator**](http://docs.google.com/javax/swing/text/html/HTMLDocument.Iterator.html)            An iterator to iterate over a particular type of tag. |
| class | [**HTMLDocument.RunElement**](http://docs.google.com/javax/swing/text/html/HTMLDocument.RunElement.html)            An element that represents a chunk of text that has a set of HTML character level attributes assigned to it. |

| **Nested classes/interfaces inherited from class javax.swing.text.**[**DefaultStyledDocument**](http://docs.google.com/javax/swing/text/DefaultStyledDocument.html) |
| --- |
| [DefaultStyledDocument.AttributeUndoableEdit](http://docs.google.com/javax/swing/text/DefaultStyledDocument.AttributeUndoableEdit.html), [DefaultStyledDocument.ElementBuffer](http://docs.google.com/javax/swing/text/DefaultStyledDocument.ElementBuffer.html), [DefaultStyledDocument.ElementSpec](http://docs.google.com/javax/swing/text/DefaultStyledDocument.ElementSpec.html), [DefaultStyledDocument.SectionElement](http://docs.google.com/javax/swing/text/DefaultStyledDocument.SectionElement.html) |

| **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** | |
| --- | --- |
| static [String](http://docs.google.com/java/lang/String.html) | [**AdditionalComments**](http://docs.google.com/javax/swing/text/html/HTMLDocument.html#AdditionalComments)            Document property key value. |

| **Fields inherited from class javax.swing.text.**[**DefaultStyledDocument**](http://docs.google.com/javax/swing/text/DefaultStyledDocument.html) |
| --- |
| [buffer](http://docs.google.com/javax/swing/text/DefaultStyledDocument.html#buffer), [BUFFER\_SIZE\_DEFAULT](http://docs.google.com/javax/swing/text/DefaultStyledDocument.html#BUFFER_SIZE_DEFAULT) |

| **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** | |
| --- | --- |
| [**HTMLDocument**](http://docs.google.com/javax/swing/text/html/HTMLDocument.html#HTMLDocument())()            Constructs an HTML document using the default buffer size and a default StyleSheet. |
| [**HTMLDocument**](http://docs.google.com/javax/swing/text/html/HTMLDocument.html#HTMLDocument(javax.swing.text.AbstractDocument.Content,%20javax.swing.text.html.StyleSheet))([AbstractDocument.Content](http://docs.google.com/javax/swing/text/AbstractDocument.Content.html) c, [StyleSheet](http://docs.google.com/javax/swing/text/html/StyleSheet.html) styles)            Constructs an HTML document with the given content storage implementation and the given style/attribute storage mechanism. |
| [**HTMLDocument**](http://docs.google.com/javax/swing/text/html/HTMLDocument.html#HTMLDocument(javax.swing.text.html.StyleSheet))([StyleSheet](http://docs.google.com/javax/swing/text/html/StyleSheet.html) styles)            Constructs an HTML document with the default content storage implementation and the specified style/attribute storage mechanism. |

| **Method Summary** | |
| --- | --- |
| protected  void | [**create**](http://docs.google.com/javax/swing/text/html/HTMLDocument.html#create(javax.swing.text.DefaultStyledDocument.ElementSpec%5B%5D))([DefaultStyledDocument.ElementSpec](http://docs.google.com/javax/swing/text/DefaultStyledDocument.ElementSpec.html)[] data)            Replaces the contents of the document with the given element specifications. |
| protected  [Element](http://docs.google.com/javax/swing/text/Element.html) | [**createBranchElement**](http://docs.google.com/javax/swing/text/html/HTMLDocument.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  [AbstractDocument.AbstractElement](http://docs.google.com/javax/swing/text/AbstractDocument.AbstractElement.html) | [**createDefaultRoot**](http://docs.google.com/javax/swing/text/html/HTMLDocument.html#createDefaultRoot())()            Creates the root element to be used to represent the default document structure. |
| protected  [Element](http://docs.google.com/javax/swing/text/Element.html) | [**createLeafElement**](http://docs.google.com/javax/swing/text/html/HTMLDocument.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 that directly represents text (doesn't have any children). |
| protected  void | [**fireChangedUpdate**](http://docs.google.com/javax/swing/text/html/HTMLDocument.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 | [**fireUndoableEditUpdate**](http://docs.google.com/javax/swing/text/html/HTMLDocument.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. |
| [URL](http://docs.google.com/java/net/URL.html) | [**getBase**](http://docs.google.com/javax/swing/text/html/HTMLDocument.html#getBase())()            Returns the location to resolve relative URLs against. |
| [Element](http://docs.google.com/javax/swing/text/Element.html) | [**getElement**](http://docs.google.com/javax/swing/text/html/HTMLDocument.html#getElement(javax.swing.text.Element,%20java.lang.Object,%20java.lang.Object))([Element](http://docs.google.com/javax/swing/text/Element.html) e, [Object](http://docs.google.com/java/lang/Object.html) attribute, [Object](http://docs.google.com/java/lang/Object.html) value)            Returns the child element of e that contains the attribute, attribute with value value, or null if one isn't found. |
| [Element](http://docs.google.com/javax/swing/text/Element.html) | [**getElement**](http://docs.google.com/javax/swing/text/html/HTMLDocument.html#getElement(java.lang.String))([String](http://docs.google.com/java/lang/String.html) id)            Returns the element that has the given id Attribute. |
| [HTMLDocument.Iterator](http://docs.google.com/javax/swing/text/html/HTMLDocument.Iterator.html) | [**getIterator**](http://docs.google.com/javax/swing/text/html/HTMLDocument.html#getIterator(javax.swing.text.html.HTML.Tag))([HTML.Tag](http://docs.google.com/javax/swing/text/html/HTML.Tag.html) t)            Fetches an iterator for the specified HTML tag. |
| [HTMLEditorKit.Parser](http://docs.google.com/javax/swing/text/html/HTMLEditorKit.Parser.html) | [**getParser**](http://docs.google.com/javax/swing/text/html/HTMLDocument.html#getParser())()            Returns the parser that is used when inserting HTML into the existing document. |
| boolean | [**getPreservesUnknownTags**](http://docs.google.com/javax/swing/text/html/HTMLDocument.html#getPreservesUnknownTags())()            Returns the behavior the parser observes when encountering unknown tags. |
| [HTMLEditorKit.ParserCallback](http://docs.google.com/javax/swing/text/html/HTMLEditorKit.ParserCallback.html) | [**getReader**](http://docs.google.com/javax/swing/text/html/HTMLDocument.html#getReader(int))(int pos)            Fetches the reader for the parser to use when loading the document with HTML. |
| [HTMLEditorKit.ParserCallback](http://docs.google.com/javax/swing/text/html/HTMLEditorKit.ParserCallback.html) | [**getReader**](http://docs.google.com/javax/swing/text/html/HTMLDocument.html#getReader(int,%20int,%20int,%20javax.swing.text.html.HTML.Tag))(int pos, int popDepth, int pushDepth, [HTML.Tag](http://docs.google.com/javax/swing/text/html/HTML.Tag.html) insertTag)            Returns the reader for the parser to use to load the document with HTML. |
| [StyleSheet](http://docs.google.com/javax/swing/text/html/StyleSheet.html) | [**getStyleSheet**](http://docs.google.com/javax/swing/text/html/HTMLDocument.html#getStyleSheet())()            Fetches the StyleSheet with the document-specific display rules (CSS) that were specified in the HTML document itself. |
| int | [**getTokenThreshold**](http://docs.google.com/javax/swing/text/html/HTMLDocument.html#getTokenThreshold())()            Gets the number of tokens to buffer before trying to update the documents element structure. |
| protected  void | [**insert**](http://docs.google.com/javax/swing/text/html/HTMLDocument.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. |
| void | [**insertAfterEnd**](http://docs.google.com/javax/swing/text/html/HTMLDocument.html#insertAfterEnd(javax.swing.text.Element,%20java.lang.String))([Element](http://docs.google.com/javax/swing/text/Element.html) elem, [String](http://docs.google.com/java/lang/String.html) htmlText)            Inserts the HTML specified as a string after the the end of the given element. |
| void | [**insertAfterStart**](http://docs.google.com/javax/swing/text/html/HTMLDocument.html#insertAfterStart(javax.swing.text.Element,%20java.lang.String))([Element](http://docs.google.com/javax/swing/text/Element.html) elem, [String](http://docs.google.com/java/lang/String.html) htmlText)            Inserts the HTML specified as a string at the start of the element. |
| void | [**insertBeforeEnd**](http://docs.google.com/javax/swing/text/html/HTMLDocument.html#insertBeforeEnd(javax.swing.text.Element,%20java.lang.String))([Element](http://docs.google.com/javax/swing/text/Element.html) elem, [String](http://docs.google.com/java/lang/String.html) htmlText)            Inserts the HTML specified as a string at the end of the element. |
| void | [**insertBeforeStart**](http://docs.google.com/javax/swing/text/html/HTMLDocument.html#insertBeforeStart(javax.swing.text.Element,%20java.lang.String))([Element](http://docs.google.com/javax/swing/text/Element.html) elem, [String](http://docs.google.com/java/lang/String.html) htmlText)            Inserts the HTML specified as a string before the start of the given element. |
| protected  void | [**insertUpdate**](http://docs.google.com/javax/swing/text/html/HTMLDocument.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 | [**processHTMLFrameHyperlinkEvent**](http://docs.google.com/javax/swing/text/html/HTMLDocument.html#processHTMLFrameHyperlinkEvent(javax.swing.text.html.HTMLFrameHyperlinkEvent))([HTMLFrameHyperlinkEvent](http://docs.google.com/javax/swing/text/html/HTMLFrameHyperlinkEvent.html) e)            Processes HyperlinkEvents that are generated by documents in an HTML frame. |
| void | [**setBase**](http://docs.google.com/javax/swing/text/html/HTMLDocument.html#setBase(java.net.URL))([URL](http://docs.google.com/java/net/URL.html) u)            Sets the location to resolve relative URLs against. |
| void | [**setInnerHTML**](http://docs.google.com/javax/swing/text/html/HTMLDocument.html#setInnerHTML(javax.swing.text.Element,%20java.lang.String))([Element](http://docs.google.com/javax/swing/text/Element.html) elem, [String](http://docs.google.com/java/lang/String.html) htmlText)            Replaces the children of the given element with the contents specified as an HTML string. |
| void | [**setOuterHTML**](http://docs.google.com/javax/swing/text/html/HTMLDocument.html#setOuterHTML(javax.swing.text.Element,%20java.lang.String))([Element](http://docs.google.com/javax/swing/text/Element.html) elem, [String](http://docs.google.com/java/lang/String.html) htmlText)            Replaces the given element in the parent with the contents specified as an HTML string. |
| void | [**setParagraphAttributes**](http://docs.google.com/javax/swing/text/html/HTMLDocument.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. |
| void | [**setParser**](http://docs.google.com/javax/swing/text/html/HTMLDocument.html#setParser(javax.swing.text.html.HTMLEditorKit.Parser))([HTMLEditorKit.Parser](http://docs.google.com/javax/swing/text/html/HTMLEditorKit.Parser.html) parser)            Sets the parser that is used by the methods that insert html into the existing document, such as setInnerHTML, and setOuterHTML. |
| void | [**setPreservesUnknownTags**](http://docs.google.com/javax/swing/text/html/HTMLDocument.html#setPreservesUnknownTags(boolean))(boolean preservesTags)            Determines how unknown tags are handled by the parser. |
| void | [**setTokenThreshold**](http://docs.google.com/javax/swing/text/html/HTMLDocument.html#setTokenThreshold(int))(int n)            Sets the number of tokens to buffer before trying to update the documents element structure. |

| **Methods inherited from class javax.swing.text.**[**DefaultStyledDocument**](http://docs.google.com/javax/swing/text/DefaultStyledDocument.html) |
| --- |
| [addDocumentListener](http://docs.google.com/javax/swing/text/DefaultStyledDocument.html#addDocumentListener(javax.swing.event.DocumentListener)), [addStyle](http://docs.google.com/javax/swing/text/DefaultStyledDocument.html#addStyle(java.lang.String,%20javax.swing.text.Style)), [getBackground](http://docs.google.com/javax/swing/text/DefaultStyledDocument.html#getBackground(javax.swing.text.AttributeSet)), [getCharacterElement](http://docs.google.com/javax/swing/text/DefaultStyledDocument.html#getCharacterElement(int)), [getDefaultRootElement](http://docs.google.com/javax/swing/text/DefaultStyledDocument.html#getDefaultRootElement()), [getFont](http://docs.google.com/javax/swing/text/DefaultStyledDocument.html#getFont(javax.swing.text.AttributeSet)), [getForeground](http://docs.google.com/javax/swing/text/DefaultStyledDocument.html#getForeground(javax.swing.text.AttributeSet)), [getLogicalStyle](http://docs.google.com/javax/swing/text/DefaultStyledDocument.html#getLogicalStyle(int)), [getParagraphElement](http://docs.google.com/javax/swing/text/DefaultStyledDocument.html#getParagraphElement(int)), [getStyle](http://docs.google.com/javax/swing/text/DefaultStyledDocument.html#getStyle(java.lang.String)), [getStyleNames](http://docs.google.com/javax/swing/text/DefaultStyledDocument.html#getStyleNames()), [removeDocumentListener](http://docs.google.com/javax/swing/text/DefaultStyledDocument.html#removeDocumentListener(javax.swing.event.DocumentListener)), [removeStyle](http://docs.google.com/javax/swing/text/DefaultStyledDocument.html#removeStyle(java.lang.String)), [removeUpdate](http://docs.google.com/javax/swing/text/DefaultStyledDocument.html#removeUpdate(javax.swing.text.AbstractDocument.DefaultDocumentEvent)), [setCharacterAttributes](http://docs.google.com/javax/swing/text/DefaultStyledDocument.html#setCharacterAttributes(int,%20int,%20javax.swing.text.AttributeSet,%20boolean)), [setLogicalStyle](http://docs.google.com/javax/swing/text/DefaultStyledDocument.html#setLogicalStyle(int,%20javax.swing.text.Style)), [styleChanged](http://docs.google.com/javax/swing/text/DefaultStyledDocument.html#styleChanged(javax.swing.text.Style)) |

| **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)), [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)), [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)), [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** |
| --- |

### AdditionalComments

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

Document property key value. The value for the key will be a Vector of Strings that are comments not found in the body.

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

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

### HTMLDocument

public **HTMLDocument**()

Constructs an HTML document using the default buffer size and a default StyleSheet. This is a convenience method for the constructor HTMLDocument(Content, StyleSheet).

### HTMLDocument

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

Constructs an HTML document with the default content storage implementation and the specified style/attribute storage mechanism. This is a convenience method for the constructor HTMLDocument(Content, StyleSheet).

**Parameters:**styles - the styles

### HTMLDocument

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

Constructs an HTML document with the given content storage implementation and the given style/attribute storage mechanism.

**Parameters:**c - the container for the contentstyles - the styles

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

### getReader

public [HTMLEditorKit.ParserCallback](http://docs.google.com/javax/swing/text/html/HTMLEditorKit.ParserCallback.html) **getReader**(int pos)

Fetches the reader for the parser to use when loading the document with HTML. This is implemented to return an instance of HTMLDocument.HTMLReader. Subclasses can reimplement this method to change how the document gets structured if desired. (For example, to handle custom tags, or structurally represent character style elements.)

**Parameters:**pos - the starting position **Returns:**the reader used by the parser to load the document

### getReader

public [HTMLEditorKit.ParserCallback](http://docs.google.com/javax/swing/text/html/HTMLEditorKit.ParserCallback.html) **getReader**(int pos,  
 int popDepth,  
 int pushDepth,  
 [HTML.Tag](http://docs.google.com/javax/swing/text/html/HTML.Tag.html) insertTag)

Returns the reader for the parser to use to load the document with HTML. This is implemented to return an instance of HTMLDocument.HTMLReader. Subclasses can reimplement this method to change how the document gets structured if desired. (For example, to handle custom tags, or structurally represent character style elements.)

This is a convenience method for getReader(int, int, int, HTML.Tag, TRUE).

**Parameters:**popDepth - the number of ElementSpec.EndTagTypes to generate before insertingpushDepth - the number of ElementSpec.StartTagTypes with a direction of ElementSpec.JoinNextDirection that should be generated before inserting, but after the end tags have been generatedinsertTag - the first tag to start inserting into document **Returns:**the reader used by the parser to load the document

### getBase

public [URL](http://docs.google.com/java/net/URL.html) **getBase**()

Returns the location to resolve relative URLs against. By default this will be the document's URL if the document was loaded from a URL. If a base tag is found and can be parsed, it will be used as the base location.

**Returns:**the base location

### setBase

public void **setBase**([URL](http://docs.google.com/java/net/URL.html) u)

Sets the location to resolve relative URLs against. By default this will be the document's URL if the document was loaded from a URL. If a base tag is found and can be parsed, it will be used as the base location.

This also sets the base of the StyleSheet to be u as well as the base of the document.

**Parameters:**u - the desired base URL

### 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 how elements get created in the document. The parsing determines what structure is needed and creates the specification as a set of tokens that describe the edit while leaving the document free of a write-lock. This method can then be called in bursts by the reader to acquire a write-lock for a shorter duration (i.e. while the document is actually being altered).

**Overrides:**[insert](http://docs.google.com/javax/swing/text/DefaultStyledDocument.html#insert(int,%20javax.swing.text.DefaultStyledDocument.ElementSpec%5B%5D)) in class [DefaultStyledDocument](http://docs.google.com/javax/swing/text/DefaultStyledDocument.html) **Parameters:**offset - the starting offsetdata - the element data **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.

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

### create

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

Replaces the contents of the document with the given element specifications. This is called before insert if the loading is done in bursts. This is the only method called if loading the document entirely in one burst.

**Overrides:**[create](http://docs.google.com/javax/swing/text/DefaultStyledDocument.html#create(javax.swing.text.DefaultStyledDocument.ElementSpec%5B%5D)) in class [DefaultStyledDocument](http://docs.google.com/javax/swing/text/DefaultStyledDocument.html) **Parameters:**data - the new contents of the document

### 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)**Overrides:**[setParagraphAttributes](http://docs.google.com/javax/swing/text/DefaultStyledDocument.html#setParagraphAttributes(int,%20int,%20javax.swing.text.AttributeSet,%20boolean)) in class [DefaultStyledDocument](http://docs.google.com/javax/swing/text/DefaultStyledDocument.html) **Parameters:**offset - the offset into the paragraph (must be at least 0)length - the number of characters affected (must be at least 0)s - the attributesreplace - whether to replace existing attributes, or merge them

### getStyleSheet

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

Fetches the StyleSheet with the document-specific display rules (CSS) that were specified in the HTML document itself.

**Returns:**the StyleSheet

### getIterator

public [HTMLDocument.Iterator](http://docs.google.com/javax/swing/text/html/HTMLDocument.Iterator.html) **getIterator**([HTML.Tag](http://docs.google.com/javax/swing/text/html/HTML.Tag.html) t)

Fetches an iterator for the specified HTML tag. This can be used for things like iterating over the set of anchors contained, or iterating over the input elements.

**Parameters:**t - the requested HTML.Tag **Returns:**the Iterator for the given HTML tag**See Also:**[HTML.Tag](http://docs.google.com/javax/swing/text/html/HTML.Tag.html)

### 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 that directly represents text (doesn't have any children). This is implemented to return an element of type HTMLDocument.RunElement.

**Overrides:**[createLeafElement](http://docs.google.com/javax/swing/text/AbstractDocument.html#createLeafElement(javax.swing.text.Element,%20javax.swing.text.AttributeSet,%20int,%20int)) in class [AbstractDocument](http://docs.google.com/javax/swing/text/AbstractDocument.html) **Parameters:**parent - the parent elementa - the attributes for the elementp0 - the beginning of the range (must be at least 0)p1 - the end of the range (must be at least 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. This is implemented to return an element of type HTMLDocument.BlockElement.

**Overrides:**[createBranchElement](http://docs.google.com/javax/swing/text/AbstractDocument.html#createBranchElement(javax.swing.text.Element,%20javax.swing.text.AttributeSet)) in class [AbstractDocument](http://docs.google.com/javax/swing/text/AbstractDocument.html) **Parameters:**parent - the parent elementa - the attributes **Returns:**the element

### 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.

**Overrides:**[createDefaultRoot](http://docs.google.com/javax/swing/text/DefaultStyledDocument.html#createDefaultRoot()) in class [DefaultStyledDocument](http://docs.google.com/javax/swing/text/DefaultStyledDocument.html) **Returns:**the element base

### setTokenThreshold

public void **setTokenThreshold**(int n)

Sets the number of tokens to buffer before trying to update the documents element structure.

**Parameters:**n - the number of tokens to buffer

### getTokenThreshold

public int **getTokenThreshold**()

Gets the number of tokens to buffer before trying to update the documents element structure. The default value is Integer.MAX\_VALUE.

**Returns:**the number of tokens to buffer

### setPreservesUnknownTags

public void **setPreservesUnknownTags**(boolean preservesTags)

Determines how unknown tags are handled by the parser. If set to true, unknown tags are put in the model, otherwise they are dropped.

**Parameters:**preservesTags - true if unknown tags should be saved in the model, otherwise tags are dropped**See Also:**[HTML.Tag](http://docs.google.com/javax/swing/text/html/HTML.Tag.html)

### getPreservesUnknownTags

public boolean **getPreservesUnknownTags**()

Returns the behavior the parser observes when encountering unknown tags.

**Returns:**true if unknown tags are to be preserved when parsing**See Also:**[HTML.Tag](http://docs.google.com/javax/swing/text/html/HTML.Tag.html)

### processHTMLFrameHyperlinkEvent

public void **processHTMLFrameHyperlinkEvent**([HTMLFrameHyperlinkEvent](http://docs.google.com/javax/swing/text/html/HTMLFrameHyperlinkEvent.html) e)

Processes HyperlinkEvents that are generated by documents in an HTML frame. The HyperlinkEvent type, as the parameter suggests, is HTMLFrameHyperlinkEvent. In addition to the typical information contained in a HyperlinkEvent, this event contains the element that corresponds to the frame in which the click happened (the source element) and the target name. The target name has 4 possible values:

* \_self
* \_parent
* \_top
* a named frame

If target is \_self, the action is to change the value of the HTML.Attribute.SRC attribute and fires a ChangedUpdate event.

If the target is \_parent, then it deletes the parent element, which is a <FRAMESET> element, and inserts a new <FRAME> element, and sets its HTML.Attribute.SRC attribute to have a value equal to the destination URL and fire a RemovedUpdate and InsertUpdate.

If the target is \_top, this method does nothing. In the implementation of the view for a frame, namely the FrameView, the processing of \_top is handled. Given that \_top implies replacing the entire document, it made sense to handle this outside of the document that it will replace.

If the target is a named frame, then the element hierarchy is searched for an element with a name equal to the target, its HTML.Attribute.SRC attribute is updated and a ChangedUpdate event is fired.

**Parameters:**e - the event

### setParser

public void **setParser**([HTMLEditorKit.Parser](http://docs.google.com/javax/swing/text/html/HTMLEditorKit.Parser.html) parser)

Sets the parser that is used by the methods that insert html into the existing document, such as setInnerHTML, and setOuterHTML.

HTMLEditorKit.createDefaultDocument will set the parser for you. If you create an HTMLDocument by hand, be sure and set the parser accordingly.

**Parameters:**parser - the parser to be used for text insertion**Since:** 1.3

### getParser

public [HTMLEditorKit.Parser](http://docs.google.com/javax/swing/text/html/HTMLEditorKit.Parser.html) **getParser**()

Returns the parser that is used when inserting HTML into the existing document.

**Returns:**the parser used for text insertion**Since:** 1.3

### setInnerHTML

public void **setInnerHTML**([Element](http://docs.google.com/javax/swing/text/Element.html) elem,  
 [String](http://docs.google.com/java/lang/String.html) htmlText)  
 throws [BadLocationException](http://docs.google.com/javax/swing/text/BadLocationException.html),  
 [IOException](http://docs.google.com/java/io/IOException.html)

Replaces the children of the given element with the contents specified as an HTML string.

This will be seen as at least two events, n inserts followed by a remove.

For this to work correcty, the document must have an HTMLEditorKit.Parser set. This will be the case if the document was created from an HTMLEditorKit via the createDefaultDocument method.

**Parameters:**elem - the branch element whose children will be replacedhtmlText - the string to be parsed and assigned to elem **Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if elem is a leaf [IllegalStateException](http://docs.google.com/java/lang/IllegalStateException.html) - if an HTMLEditorKit.Parser has not been defined [BadLocationException](http://docs.google.com/javax/swing/text/BadLocationException.html) [IOException](http://docs.google.com/java/io/IOException.html)**Since:** 1.3

### setOuterHTML

public void **setOuterHTML**([Element](http://docs.google.com/javax/swing/text/Element.html) elem,  
 [String](http://docs.google.com/java/lang/String.html) htmlText)  
 throws [BadLocationException](http://docs.google.com/javax/swing/text/BadLocationException.html),  
 [IOException](http://docs.google.com/java/io/IOException.html)

Replaces the given element in the parent with the contents specified as an HTML string.

This will be seen as at least two events, n inserts followed by a remove.

When replacing a leaf this will attempt to make sure there is a newline present if one is needed. This may result in an additional element being inserted. Consider, if you were to replace a character element that contained a newline with <img> this would create two elements, one for the image, ane one for the newline.

If you try to replace the element at length you will most likely end up with two elements, eg setOuterHTML(getCharacterElement (getLength()), "blah") will result in two leaf elements at the end, one representing 'blah', and the other representing the end element.

For this to work correcty, the document must have an HTMLEditorKit.Parser set. This will be the case if the document was created from an HTMLEditorKit via the createDefaultDocument method.

**Parameters:**elem - the branch element whose children will be replacedhtmlText - the string to be parsed and assigned to elem **Throws:** [IllegalStateException](http://docs.google.com/java/lang/IllegalStateException.html) - if an HTMLEditorKit.Parser has not been set [BadLocationException](http://docs.google.com/javax/swing/text/BadLocationException.html) [IOException](http://docs.google.com/java/io/IOException.html)**Since:** 1.3

### insertAfterStart

public void **insertAfterStart**([Element](http://docs.google.com/javax/swing/text/Element.html) elem,  
 [String](http://docs.google.com/java/lang/String.html) htmlText)  
 throws [BadLocationException](http://docs.google.com/javax/swing/text/BadLocationException.html),  
 [IOException](http://docs.google.com/java/io/IOException.html)

Inserts the HTML specified as a string at the start of the element.

For this to work correcty, the document must have an HTMLEditorKit.Parser set. This will be the case if the document was created from an HTMLEditorKit via the createDefaultDocument method.

**Parameters:**elem - the branch element to be the root for the new texthtmlText - the string to be parsed and assigned to elem **Throws:** [IllegalStateException](http://docs.google.com/java/lang/IllegalStateException.html) - if an HTMLEditorKit.Parser has not been set on the document [BadLocationException](http://docs.google.com/javax/swing/text/BadLocationException.html) [IOException](http://docs.google.com/java/io/IOException.html)**Since:** 1.3

### insertBeforeEnd

public void **insertBeforeEnd**([Element](http://docs.google.com/javax/swing/text/Element.html) elem,  
 [String](http://docs.google.com/java/lang/String.html) htmlText)  
 throws [BadLocationException](http://docs.google.com/javax/swing/text/BadLocationException.html),  
 [IOException](http://docs.google.com/java/io/IOException.html)

Inserts the HTML specified as a string at the end of the element.

If elem's children are leaves, and the character at a elem.getEndOffset() - 1 is a newline, this will insert before the newline so that there isn't text after the newline.

For this to work correcty, the document must have an HTMLEditorKit.Parser set. This will be the case if the document was created from an HTMLEditorKit via the createDefaultDocument method.

**Parameters:**elem - the element to be the root for the new texthtmlText - the string to be parsed and assigned to elem **Throws:** [IllegalStateException](http://docs.google.com/java/lang/IllegalStateException.html) - if an HTMLEditorKit.Parser has not been set on the document [BadLocationException](http://docs.google.com/javax/swing/text/BadLocationException.html) [IOException](http://docs.google.com/java/io/IOException.html)**Since:** 1.3

### insertBeforeStart

public void **insertBeforeStart**([Element](http://docs.google.com/javax/swing/text/Element.html) elem,  
 [String](http://docs.google.com/java/lang/String.html) htmlText)  
 throws [BadLocationException](http://docs.google.com/javax/swing/text/BadLocationException.html),  
 [IOException](http://docs.google.com/java/io/IOException.html)

Inserts the HTML specified as a string before the start of the given element.

For this to work correcty, the document must have an HTMLEditorKit.Parser set. This will be the case if the document was created from an HTMLEditorKit via the createDefaultDocument method.

**Parameters:**elem - the element to be the root for the new texthtmlText - the string to be parsed and assigned to elem **Throws:** [IllegalStateException](http://docs.google.com/java/lang/IllegalStateException.html) - if an HTMLEditorKit.Parser has not been set on the document [BadLocationException](http://docs.google.com/javax/swing/text/BadLocationException.html) [IOException](http://docs.google.com/java/io/IOException.html)**Since:** 1.3

### insertAfterEnd

public void **insertAfterEnd**([Element](http://docs.google.com/javax/swing/text/Element.html) elem,  
 [String](http://docs.google.com/java/lang/String.html) htmlText)  
 throws [BadLocationException](http://docs.google.com/javax/swing/text/BadLocationException.html),  
 [IOException](http://docs.google.com/java/io/IOException.html)

Inserts the HTML specified as a string after the the end of the given element.

For this to work correcty, the document must have an HTMLEditorKit.Parser set. This will be the case if the document was created from an HTMLEditorKit via the createDefaultDocument method.

**Parameters:**elem - the element to be the root for the new texthtmlText - the string to be parsed and assigned to elem **Throws:** [IllegalStateException](http://docs.google.com/java/lang/IllegalStateException.html) - if an HTMLEditorKit.Parser has not been set on the document [BadLocationException](http://docs.google.com/javax/swing/text/BadLocationException.html) [IOException](http://docs.google.com/java/io/IOException.html)**Since:** 1.3

### getElement

public [Element](http://docs.google.com/javax/swing/text/Element.html) **getElement**([String](http://docs.google.com/java/lang/String.html) id)

Returns the element that has the given id Attribute. If the element can't be found, null is returned. Note that this method works on an Attribute, *not* a character tag. In the following HTML snippet: <a id="HelloThere"> the attribute is 'id' and the character tag is 'a'. This is a convenience method for getElement(RootElement, HTML.Attribute.id, id). This is not thread-safe.

**Parameters:**id - the string representing the desired Attribute **Returns:**the element with the specified Attribute or null if it can't be found, or null if id is null**Since:** 1.3 **See Also:**[HTML.Attribute](http://docs.google.com/javax/swing/text/html/HTML.Attribute.html)

### getElement

public [Element](http://docs.google.com/javax/swing/text/Element.html) **getElement**([Element](http://docs.google.com/javax/swing/text/Element.html) e,  
 [Object](http://docs.google.com/java/lang/Object.html) attribute,  
 [Object](http://docs.google.com/java/lang/Object.html) value)

Returns the child element of e that contains the attribute, attribute with value value, or null if one isn't found. This is not thread-safe.

**Parameters:**e - the root element where the search beginsattribute - the desired Attributevalue - the values for the specified Attribute **Returns:**the element with the specified Attribute and the specified value, or null if it can't be found**Since:** 1.3 **See Also:**[HTML.Attribute](http://docs.google.com/javax/swing/text/html/HTML.Attribute.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.

**Overrides:**[fireChangedUpdate](http://docs.google.com/javax/swing/text/AbstractDocument.html#fireChangedUpdate(javax.swing.event.DocumentEvent)) in class [AbstractDocument](http://docs.google.com/javax/swing/text/AbstractDocument.html) **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.

**Overrides:**[fireUndoableEditUpdate](http://docs.google.com/javax/swing/text/AbstractDocument.html#fireUndoableEditUpdate(javax.swing.event.UndoableEditEvent)) in class [AbstractDocument](http://docs.google.com/javax/swing/text/AbstractDocument.html) **Parameters:**e - the event**See Also:**[EventListenerList](http://docs.google.com/javax/swing/event/EventListenerList.html)

| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/HTMLDocument.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/html/HTML.UnknownTag.html)   [**NEXT CLASS**](http://docs.google.com/javax/swing/text/html/HTMLDocument.BlockElement.html) | [**FRAMES**](http://docs.google.com/index.html?javax/swing/text/html/HTMLDocument.html)    [**NO FRAMES**](http://docs.google.com/HTMLDocument.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: [NESTED](#3znysh7) | [FIELD](#3dy6vkm) | [CONSTR](#17dp8vu) | [METHOD](#3rdcrjn) | DETAIL: [FIELD](#44sinio) | [CONSTR](#z337ya) | [METHOD](#2xcytpi) |

[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).