| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/MinimalHTMLWriter.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/ListView.html)   [**NEXT CLASS**](http://docs.google.com/javax/swing/text/html/ObjectView.html) | [**FRAMES**](http://docs.google.com/index.html?javax/swing/text/html/MinimalHTMLWriter.html)    [**NO FRAMES**](http://docs.google.com/MinimalHTMLWriter.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | [FIELD](#2et92p0) | [CONSTR](#tyjcwt) | [METHOD](#3dy6vkm) | DETAIL: FIELD | [CONSTR](#2s8eyo1) | [METHOD](#26in1rg) |

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

Class MinimalHTMLWriter

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

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

MinimalHTMLWriter is a fallback writer used by the HTMLEditorKit to write out HTML for a document that is a not produced by the EditorKit. The format for the document is:

<html>  
 <head>  
 <style>  
 <!-- list of named styles  
 p.normal {  
 font-family: SansSerif;  
 margin-height: 0;  
 font-size: 14  
 }  
 -->  
 </style>  
 </head>  
 <body>  
 <p style=normal>  
 **Bold, italic, and underline attributes  
 of the run are emitted as HTML tags.  
 The remaining attributes are emitted as  
 part of the style attribute of a <span> tag.  
 The syntax is similar to inline styles.**  
 </p>  
 </body>  
 </html>

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

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

| **Constructor Summary** | |
| --- | --- |
| [**MinimalHTMLWriter**](http://docs.google.com/javax/swing/text/html/MinimalHTMLWriter.html#MinimalHTMLWriter(java.io.Writer,%20javax.swing.text.StyledDocument))([Writer](http://docs.google.com/java/io/Writer.html) w, [StyledDocument](http://docs.google.com/javax/swing/text/StyledDocument.html) doc)            Creates a new MinimalHTMLWriter. |
| [**MinimalHTMLWriter**](http://docs.google.com/javax/swing/text/html/MinimalHTMLWriter.html#MinimalHTMLWriter(java.io.Writer,%20javax.swing.text.StyledDocument,%20int,%20int))([Writer](http://docs.google.com/java/io/Writer.html) w, [StyledDocument](http://docs.google.com/javax/swing/text/StyledDocument.html) doc, int pos, int len)            Creates a new MinimalHTMLWriter. |

| **Method Summary** | |
| --- | --- |
| protected  void | [**endFontTag**](http://docs.google.com/javax/swing/text/html/MinimalHTMLWriter.html#endFontTag())()            This is no longer used, instead <span> will be written out. |
| protected  boolean | [**inFontTag**](http://docs.google.com/javax/swing/text/html/MinimalHTMLWriter.html#inFontTag())()            Returns true if we are currently in a <font> tag. |
| protected  boolean | [**isText**](http://docs.google.com/javax/swing/text/html/MinimalHTMLWriter.html#isText(javax.swing.text.Element))([Element](http://docs.google.com/javax/swing/text/Element.html) elem)            Returns true if the element is a text element. |
| protected  void | [**startFontTag**](http://docs.google.com/javax/swing/text/html/MinimalHTMLWriter.html#startFontTag(java.lang.String))([String](http://docs.google.com/java/lang/String.html) style)            This is no longer used, instead <span> will be written out. |
| protected  void | [**text**](http://docs.google.com/javax/swing/text/html/MinimalHTMLWriter.html#text(javax.swing.text.Element))([Element](http://docs.google.com/javax/swing/text/Element.html) elem)            Writes out text. |
| void | [**write**](http://docs.google.com/javax/swing/text/html/MinimalHTMLWriter.html#write())()            Generates HTML output from a StyledDocument. |
| protected  void | [**writeAttributes**](http://docs.google.com/javax/swing/text/html/MinimalHTMLWriter.html#writeAttributes(javax.swing.text.AttributeSet))([AttributeSet](http://docs.google.com/javax/swing/text/AttributeSet.html) attr)            Writes out all the attributes for the following types: StyleConstants.ParagraphConstants, StyleConstants.CharacterConstants, StyleConstants.FontConstants, StyleConstants.ColorConstants. |
| protected  void | [**writeBody**](http://docs.google.com/javax/swing/text/html/MinimalHTMLWriter.html#writeBody())()            Iterates over the elements in the document and processes elements based on whether they are branch elements or leaf elements. |
| protected  void | [**writeComponent**](http://docs.google.com/javax/swing/text/html/MinimalHTMLWriter.html#writeComponent(javax.swing.text.Element))([Element](http://docs.google.com/javax/swing/text/Element.html) elem)            Responsible for handling Component Elements; deliberately unimplemented. |
| protected  void | [**writeContent**](http://docs.google.com/javax/swing/text/html/MinimalHTMLWriter.html#writeContent(javax.swing.text.Element,%20boolean))([Element](http://docs.google.com/javax/swing/text/Element.html) elem, boolean needsIndenting)            Writes out the attribute set in an HTML-compliant manner. |
| protected  void | [**writeEndParagraph**](http://docs.google.com/javax/swing/text/html/MinimalHTMLWriter.html#writeEndParagraph())()            Emits an end tag for a <p> tag. |
| protected  void | [**writeEndTag**](http://docs.google.com/javax/swing/text/html/MinimalHTMLWriter.html#writeEndTag(java.lang.String))([String](http://docs.google.com/java/lang/String.html) endTag)            Writes out an end tag appropriately indented. |
| protected  void | [**writeHeader**](http://docs.google.com/javax/swing/text/html/MinimalHTMLWriter.html#writeHeader())()            Writes out the <head> and <style> tags, and then invokes writeStyles() to write out all the named styles as the content of the <style> tag. |
| protected  void | [**writeHTMLTags**](http://docs.google.com/javax/swing/text/html/MinimalHTMLWriter.html#writeHTMLTags(javax.swing.text.AttributeSet))([AttributeSet](http://docs.google.com/javax/swing/text/AttributeSet.html) attr)            Generates bold <b>, italic <i>, and <u> tags for the text based on its attribute settings. |
| protected  void | [**writeImage**](http://docs.google.com/javax/swing/text/html/MinimalHTMLWriter.html#writeImage(javax.swing.text.Element))([Element](http://docs.google.com/javax/swing/text/Element.html) elem)            Responsible for handling Icon Elements; deliberately unimplemented. |
| protected  void | [**writeLeaf**](http://docs.google.com/javax/swing/text/html/MinimalHTMLWriter.html#writeLeaf(javax.swing.text.Element))([Element](http://docs.google.com/javax/swing/text/Element.html) elem)            Responsible for writing out other non-text leaf elements. |
| protected  void | [**writeNonHTMLAttributes**](http://docs.google.com/javax/swing/text/html/MinimalHTMLWriter.html#writeNonHTMLAttributes(javax.swing.text.AttributeSet))([AttributeSet](http://docs.google.com/javax/swing/text/AttributeSet.html) attr)            Writes out the remaining character-level attributes (attributes other than bold, italic, and underline) in an HTML-compliant way. |
| protected  void | [**writeStartParagraph**](http://docs.google.com/javax/swing/text/html/MinimalHTMLWriter.html#writeStartParagraph(javax.swing.text.Element))([Element](http://docs.google.com/javax/swing/text/Element.html) elem)            Emits the start tag for a paragraph. |
| protected  void | [**writeStartTag**](http://docs.google.com/javax/swing/text/html/MinimalHTMLWriter.html#writeStartTag(java.lang.String))([String](http://docs.google.com/java/lang/String.html) tag)            Writes out a start tag appropriately indented. |
| protected  void | [**writeStyles**](http://docs.google.com/javax/swing/text/html/MinimalHTMLWriter.html#writeStyles())()            Writes out all the named styles as the content of the <style> tag. |

| **Methods inherited from class javax.swing.text.**[**AbstractWriter**](http://docs.google.com/javax/swing/text/AbstractWriter.html) |
| --- |
| [decrIndent](http://docs.google.com/javax/swing/text/AbstractWriter.html#decrIndent()), [getCanWrapLines](http://docs.google.com/javax/swing/text/AbstractWriter.html#getCanWrapLines()), [getCurrentLineLength](http://docs.google.com/javax/swing/text/AbstractWriter.html#getCurrentLineLength()), [getDocument](http://docs.google.com/javax/swing/text/AbstractWriter.html#getDocument()), [getElementIterator](http://docs.google.com/javax/swing/text/AbstractWriter.html#getElementIterator()), [getEndOffset](http://docs.google.com/javax/swing/text/AbstractWriter.html#getEndOffset()), [getIndentLevel](http://docs.google.com/javax/swing/text/AbstractWriter.html#getIndentLevel()), [getIndentSpace](http://docs.google.com/javax/swing/text/AbstractWriter.html#getIndentSpace()), [getLineLength](http://docs.google.com/javax/swing/text/AbstractWriter.html#getLineLength()), [getLineSeparator](http://docs.google.com/javax/swing/text/AbstractWriter.html#getLineSeparator()), [getStartOffset](http://docs.google.com/javax/swing/text/AbstractWriter.html#getStartOffset()), [getText](http://docs.google.com/javax/swing/text/AbstractWriter.html#getText(javax.swing.text.Element)), [getWriter](http://docs.google.com/javax/swing/text/AbstractWriter.html#getWriter()), [incrIndent](http://docs.google.com/javax/swing/text/AbstractWriter.html#incrIndent()), [indent](http://docs.google.com/javax/swing/text/AbstractWriter.html#indent()), [inRange](http://docs.google.com/javax/swing/text/AbstractWriter.html#inRange(javax.swing.text.Element)), [isLineEmpty](http://docs.google.com/javax/swing/text/AbstractWriter.html#isLineEmpty()), [output](http://docs.google.com/javax/swing/text/AbstractWriter.html#output(char%5B%5D,%20int,%20int)), [setCanWrapLines](http://docs.google.com/javax/swing/text/AbstractWriter.html#setCanWrapLines(boolean)), [setCurrentLineLength](http://docs.google.com/javax/swing/text/AbstractWriter.html#setCurrentLineLength(int)), [setIndentSpace](http://docs.google.com/javax/swing/text/AbstractWriter.html#setIndentSpace(int)), [setLineLength](http://docs.google.com/javax/swing/text/AbstractWriter.html#setLineLength(int)), [setLineSeparator](http://docs.google.com/javax/swing/text/AbstractWriter.html#setLineSeparator(java.lang.String)), [write](http://docs.google.com/javax/swing/text/AbstractWriter.html#write(char)), [write](http://docs.google.com/javax/swing/text/AbstractWriter.html#write(char%5B%5D,%20int,%20int)), [write](http://docs.google.com/javax/swing/text/AbstractWriter.html#write(java.lang.String)), [writeLineSeparator](http://docs.google.com/javax/swing/text/AbstractWriter.html#writeLineSeparator()) |

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

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

### MinimalHTMLWriter

public **MinimalHTMLWriter**([Writer](http://docs.google.com/java/io/Writer.html) w,  
 [StyledDocument](http://docs.google.com/javax/swing/text/StyledDocument.html) doc)

Creates a new MinimalHTMLWriter.

**Parameters:**w - Writerdoc - StyledDocument

### MinimalHTMLWriter

public **MinimalHTMLWriter**([Writer](http://docs.google.com/java/io/Writer.html) w,  
 [StyledDocument](http://docs.google.com/javax/swing/text/StyledDocument.html) doc,  
 int pos,  
 int len)

Creates a new MinimalHTMLWriter.

**Parameters:**w - Writerdoc - StyledDocumentpos - The location in the document to fetch the content.len - The amount to write out.

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

### write

public void **write**()  
 throws [IOException](http://docs.google.com/java/io/IOException.html),  
 [BadLocationException](http://docs.google.com/javax/swing/text/BadLocationException.html)

Generates HTML output from a StyledDocument.

**Specified by:**[write](http://docs.google.com/javax/swing/text/AbstractWriter.html#write()) in class [AbstractWriter](http://docs.google.com/javax/swing/text/AbstractWriter.html) **Throws:** [IOException](http://docs.google.com/java/io/IOException.html) - on any I/O error [BadLocationException](http://docs.google.com/javax/swing/text/BadLocationException.html) - if pos represents an invalid location within the document.

### writeAttributes

protected void **writeAttributes**([AttributeSet](http://docs.google.com/javax/swing/text/AttributeSet.html) attr)  
 throws [IOException](http://docs.google.com/java/io/IOException.html)

Writes out all the attributes for the following types: StyleConstants.ParagraphConstants, StyleConstants.CharacterConstants, StyleConstants.FontConstants, StyleConstants.ColorConstants. The attribute name and value are separated by a colon. Each pair is separated by a semicolon.

**Overrides:**[writeAttributes](http://docs.google.com/javax/swing/text/AbstractWriter.html#writeAttributes(javax.swing.text.AttributeSet)) in class [AbstractWriter](http://docs.google.com/javax/swing/text/AbstractWriter.html) **Parameters:**attr - an AttributeSet. **Throws:** [IOException](http://docs.google.com/java/io/IOException.html) - on any I/O error

### text

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

Writes out text.

**Overrides:**[text](http://docs.google.com/javax/swing/text/AbstractWriter.html#text(javax.swing.text.Element)) in class [AbstractWriter](http://docs.google.com/javax/swing/text/AbstractWriter.html) **Parameters:**elem - an Element. **Throws:** [IOException](http://docs.google.com/java/io/IOException.html) - on any I/O error [BadLocationException](http://docs.google.com/javax/swing/text/BadLocationException.html) - if pos represents an invalid location within the document.

### writeStartTag

protected void **writeStartTag**([String](http://docs.google.com/java/lang/String.html) tag)  
 throws [IOException](http://docs.google.com/java/io/IOException.html)

Writes out a start tag appropriately indented. Also increments the indent level.

**Throws:** [IOException](http://docs.google.com/java/io/IOException.html) - on any I/O error

### writeEndTag

protected void **writeEndTag**([String](http://docs.google.com/java/lang/String.html) endTag)  
 throws [IOException](http://docs.google.com/java/io/IOException.html)

Writes out an end tag appropriately indented. Also decrements the indent level.

**Throws:** [IOException](http://docs.google.com/java/io/IOException.html) - on any I/O error

### writeHeader

protected void **writeHeader**()  
 throws [IOException](http://docs.google.com/java/io/IOException.html)

Writes out the <head> and <style> tags, and then invokes writeStyles() to write out all the named styles as the content of the <style> tag. The content is surrounded by valid HTML comment markers to ensure that the document is viewable in applications/browsers that do not support the tag.

**Throws:** [IOException](http://docs.google.com/java/io/IOException.html) - on any I/O error

### writeStyles

protected void **writeStyles**()  
 throws [IOException](http://docs.google.com/java/io/IOException.html)

Writes out all the named styles as the content of the <style> tag.

**Throws:** [IOException](http://docs.google.com/java/io/IOException.html) - on any I/O error

### writeBody

protected void **writeBody**()  
 throws [IOException](http://docs.google.com/java/io/IOException.html),  
 [BadLocationException](http://docs.google.com/javax/swing/text/BadLocationException.html)

Iterates over the elements in the document and processes elements based on whether they are branch elements or leaf elements. This method specially handles leaf elements that are text.

**Throws:** [IOException](http://docs.google.com/java/io/IOException.html) - on any I/O error [BadLocationException](http://docs.google.com/javax/swing/text/BadLocationException.html)

### writeEndParagraph

protected void **writeEndParagraph**()  
 throws [IOException](http://docs.google.com/java/io/IOException.html)

Emits an end tag for a <p> tag. Before writing out the tag, this method ensures that all other tags that have been opened are appropriately closed off.

**Throws:** [IOException](http://docs.google.com/java/io/IOException.html) - on any I/O error

### writeStartParagraph

protected void **writeStartParagraph**([Element](http://docs.google.com/javax/swing/text/Element.html) elem)  
 throws [IOException](http://docs.google.com/java/io/IOException.html)

Emits the start tag for a paragraph. If the paragraph has a named style associated with it, then this method also generates a class attribute for the <p> tag and sets its value to be the name of the style.

**Throws:** [IOException](http://docs.google.com/java/io/IOException.html) - on any I/O error

### writeLeaf

protected void **writeLeaf**([Element](http://docs.google.com/javax/swing/text/Element.html) elem)  
 throws [IOException](http://docs.google.com/java/io/IOException.html)

Responsible for writing out other non-text leaf elements.

**Throws:** [IOException](http://docs.google.com/java/io/IOException.html) - on any I/O error

### writeImage

protected void **writeImage**([Element](http://docs.google.com/javax/swing/text/Element.html) elem)  
 throws [IOException](http://docs.google.com/java/io/IOException.html)

Responsible for handling Icon Elements; deliberately unimplemented. How to implement this method is an issue of policy. For example, if you're generating an <img> tag, how should you represent the src attribute (the location of the image)? In certain cases it could be a URL, in others it could be read from a stream.

**Parameters:**elem - element of type StyleConstants.IconElementName **Throws:** [IOException](http://docs.google.com/java/io/IOException.html)

### writeComponent

protected void **writeComponent**([Element](http://docs.google.com/javax/swing/text/Element.html) elem)  
 throws [IOException](http://docs.google.com/java/io/IOException.html)

Responsible for handling Component Elements; deliberately unimplemented. How this method is implemented is a matter of policy.

**Throws:** [IOException](http://docs.google.com/java/io/IOException.html)

### isText

protected boolean **isText**([Element](http://docs.google.com/javax/swing/text/Element.html) elem)

Returns true if the element is a text element.

### writeContent

protected void **writeContent**([Element](http://docs.google.com/javax/swing/text/Element.html) elem,  
 boolean needsIndenting)  
 throws [IOException](http://docs.google.com/java/io/IOException.html),  
 [BadLocationException](http://docs.google.com/javax/swing/text/BadLocationException.html)

Writes out the attribute set in an HTML-compliant manner.

**Throws:** [IOException](http://docs.google.com/java/io/IOException.html) - on any I/O error [BadLocationException](http://docs.google.com/javax/swing/text/BadLocationException.html) - if pos represents an invalid location within the document.

### writeHTMLTags

protected void **writeHTMLTags**([AttributeSet](http://docs.google.com/javax/swing/text/AttributeSet.html) attr)  
 throws [IOException](http://docs.google.com/java/io/IOException.html)

Generates bold <b>, italic <i>, and <u> tags for the text based on its attribute settings.

**Throws:** [IOException](http://docs.google.com/java/io/IOException.html) - on any I/O error

### writeNonHTMLAttributes

protected void **writeNonHTMLAttributes**([AttributeSet](http://docs.google.com/javax/swing/text/AttributeSet.html) attr)  
 throws [IOException](http://docs.google.com/java/io/IOException.html)

Writes out the remaining character-level attributes (attributes other than bold, italic, and underline) in an HTML-compliant way. Given that attributes such as font family and font size have no direct mapping to HTML tags, a <span> tag is generated and its style attribute is set to contain the list of remaining attributes just like inline styles.

**Throws:** [IOException](http://docs.google.com/java/io/IOException.html) - on any I/O error

### inFontTag

protected boolean **inFontTag**()

Returns true if we are currently in a <font> tag.

### endFontTag

protected void **endFontTag**()  
 throws [IOException](http://docs.google.com/java/io/IOException.html)

This is no longer used, instead <span> will be written out.

Writes out an end tag for the <font> tag.

**Throws:** [IOException](http://docs.google.com/java/io/IOException.html) - on any I/O error

### startFontTag

protected void **startFontTag**([String](http://docs.google.com/java/lang/String.html) style)  
 throws [IOException](http://docs.google.com/java/io/IOException.html)

This is no longer used, instead <span> will be written out.

Writes out a start tag for the <font> tag. Because font tags cannot be nested, this method closes out any enclosing font tag before writing out a new start tag.

**Throws:** [IOException](http://docs.google.com/java/io/IOException.html) - on any I/O error

| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/MinimalHTMLWriter.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/ListView.html)   [**NEXT CLASS**](http://docs.google.com/javax/swing/text/html/ObjectView.html) | [**FRAMES**](http://docs.google.com/index.html?javax/swing/text/html/MinimalHTMLWriter.html)    [**NO FRAMES**](http://docs.google.com/MinimalHTMLWriter.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | [FIELD](#2et92p0) | [CONSTR](#tyjcwt) | [METHOD](#3dy6vkm) | DETAIL: FIELD | [CONSTR](#2s8eyo1) | [METHOD](#26in1rg) |

[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).