| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/Parser.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/parser/Entity.html)   [**NEXT CLASS**](http://docs.google.com/javax/swing/text/html/parser/ParserDelegator.html) | [**FRAMES**](http://docs.google.com/index.html?javax/swing/text/html/parser/Parser.html)    [**NO FRAMES**](http://docs.google.com/Parser.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | [FIELD](#3znysh7) | [CONSTR](#tyjcwt) | [METHOD](#3dy6vkm) | DETAIL: [FIELD](#4d34og8) | [CONSTR](#3rdcrjn) | [METHOD](#lnxbz9) |

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

Class Parser

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

**All Implemented Interfaces:** [DTDConstants](http://docs.google.com/javax/swing/text/html/parser/DTDConstants.html) **Direct Known Subclasses:** [DocumentParser](http://docs.google.com/javax/swing/text/html/parser/DocumentParser.html)

public class **Parser**extends [Object](http://docs.google.com/java/lang/Object.html)implements [DTDConstants](http://docs.google.com/javax/swing/text/html/parser/DTDConstants.html)

A simple DTD-driven HTML parser. The parser reads an HTML file from an InputStream and calls various methods (which should be overridden in a subclass) when tags and data are encountered.

Unfortunately there are many badly implemented HTML parsers out there, and as a result there are many badly formatted HTML files. This parser attempts to parse most HTML files. This means that the implementation sometimes deviates from the SGML specification in favor of HTML.

The parser treats \r and \r\n as \n. Newlines after starttags and before end tags are ignored just as specified in the SGML/HTML specification.

The html spec does not specify how spaces are to be coalesced very well. Specifically, the following scenarios are not discussed (note that a space should be used here, but I am using &nbsp to force the space to be displayed):

'<b>blah <i> <strike> foo' which can be treated as: '<b>blah <i><strike>foo'

as well as: '<p><a href="xx"> <em>Using</em></a></p>' which appears to be treated as: '<p><a href="xx"><em>Using</em></a></p>'

If strict is false, when a tag that breaks flow, (TagElement.breaksFlows) or trailing whitespace is encountered, all whitespace will be ignored until a non whitespace character is encountered. This appears to give behavior closer to the popular browsers.

**See Also:**[DTD](http://docs.google.com/javax/swing/text/html/parser/DTD.html), [TagElement](http://docs.google.com/javax/swing/text/html/parser/TagElement.html), [SimpleAttributeSet](http://docs.google.com/javax/swing/text/SimpleAttributeSet.html)

| **Field Summary** | |
| --- | --- |
| protected  [DTD](http://docs.google.com/javax/swing/text/html/parser/DTD.html) | [**dtd**](http://docs.google.com/javax/swing/text/html/parser/Parser.html#dtd) |
| protected  boolean | [**strict**](http://docs.google.com/javax/swing/text/html/parser/Parser.html#strict)            This flag determines whether or not the Parser will be strict in enforcing SGML compatibility. |

| **Fields inherited from interface javax.swing.text.html.parser.**[**DTDConstants**](http://docs.google.com/javax/swing/text/html/parser/DTDConstants.html) |
| --- |
| [ANY](http://docs.google.com/javax/swing/text/html/parser/DTDConstants.html#ANY), [CDATA](http://docs.google.com/javax/swing/text/html/parser/DTDConstants.html#CDATA), [CONREF](http://docs.google.com/javax/swing/text/html/parser/DTDConstants.html#CONREF), [CURRENT](http://docs.google.com/javax/swing/text/html/parser/DTDConstants.html#CURRENT), [DEFAULT](http://docs.google.com/javax/swing/text/html/parser/DTDConstants.html#DEFAULT), [EMPTY](http://docs.google.com/javax/swing/text/html/parser/DTDConstants.html#EMPTY), [ENDTAG](http://docs.google.com/javax/swing/text/html/parser/DTDConstants.html#ENDTAG), [ENTITIES](http://docs.google.com/javax/swing/text/html/parser/DTDConstants.html#ENTITIES), [ENTITY](http://docs.google.com/javax/swing/text/html/parser/DTDConstants.html#ENTITY), [FIXED](http://docs.google.com/javax/swing/text/html/parser/DTDConstants.html#FIXED), [GENERAL](http://docs.google.com/javax/swing/text/html/parser/DTDConstants.html#GENERAL), [ID](http://docs.google.com/javax/swing/text/html/parser/DTDConstants.html#ID), [IDREF](http://docs.google.com/javax/swing/text/html/parser/DTDConstants.html#IDREF), [IDREFS](http://docs.google.com/javax/swing/text/html/parser/DTDConstants.html#IDREFS), [IMPLIED](http://docs.google.com/javax/swing/text/html/parser/DTDConstants.html#IMPLIED), [MD](http://docs.google.com/javax/swing/text/html/parser/DTDConstants.html#MD), [MODEL](http://docs.google.com/javax/swing/text/html/parser/DTDConstants.html#MODEL), [MS](http://docs.google.com/javax/swing/text/html/parser/DTDConstants.html#MS), [NAME](http://docs.google.com/javax/swing/text/html/parser/DTDConstants.html#NAME), [NAMES](http://docs.google.com/javax/swing/text/html/parser/DTDConstants.html#NAMES), [NMTOKEN](http://docs.google.com/javax/swing/text/html/parser/DTDConstants.html#NMTOKEN), [NMTOKENS](http://docs.google.com/javax/swing/text/html/parser/DTDConstants.html#NMTOKENS), [NOTATION](http://docs.google.com/javax/swing/text/html/parser/DTDConstants.html#NOTATION), [NUMBER](http://docs.google.com/javax/swing/text/html/parser/DTDConstants.html#NUMBER), [NUMBERS](http://docs.google.com/javax/swing/text/html/parser/DTDConstants.html#NUMBERS), [NUTOKEN](http://docs.google.com/javax/swing/text/html/parser/DTDConstants.html#NUTOKEN), [NUTOKENS](http://docs.google.com/javax/swing/text/html/parser/DTDConstants.html#NUTOKENS), [PARAMETER](http://docs.google.com/javax/swing/text/html/parser/DTDConstants.html#PARAMETER), [PI](http://docs.google.com/javax/swing/text/html/parser/DTDConstants.html#PI), [PUBLIC](http://docs.google.com/javax/swing/text/html/parser/DTDConstants.html#PUBLIC), [RCDATA](http://docs.google.com/javax/swing/text/html/parser/DTDConstants.html#RCDATA), [REQUIRED](http://docs.google.com/javax/swing/text/html/parser/DTDConstants.html#REQUIRED), [SDATA](http://docs.google.com/javax/swing/text/html/parser/DTDConstants.html#SDATA), [STARTTAG](http://docs.google.com/javax/swing/text/html/parser/DTDConstants.html#STARTTAG), [SYSTEM](http://docs.google.com/javax/swing/text/html/parser/DTDConstants.html#SYSTEM) |

| **Constructor Summary** | |
| --- | --- |
| [**Parser**](http://docs.google.com/javax/swing/text/html/parser/Parser.html#Parser(javax.swing.text.html.parser.DTD))([DTD](http://docs.google.com/javax/swing/text/html/parser/DTD.html) dtd) |

| **Method Summary** | |
| --- | --- |
| protected  void | [**endTag**](http://docs.google.com/javax/swing/text/html/parser/Parser.html#endTag(boolean))(boolean omitted)            Handle an end tag. |
| protected  void | [**error**](http://docs.google.com/javax/swing/text/html/parser/Parser.html#error(java.lang.String))([String](http://docs.google.com/java/lang/String.html) err) |
| protected  void | [**error**](http://docs.google.com/javax/swing/text/html/parser/Parser.html#error(java.lang.String,%20java.lang.String))([String](http://docs.google.com/java/lang/String.html) err, [String](http://docs.google.com/java/lang/String.html) arg1) |
| protected  void | [**error**](http://docs.google.com/javax/swing/text/html/parser/Parser.html#error(java.lang.String,%20java.lang.String,%20java.lang.String))([String](http://docs.google.com/java/lang/String.html) err, [String](http://docs.google.com/java/lang/String.html) arg1, [String](http://docs.google.com/java/lang/String.html) arg2) |
| protected  void | [**error**](http://docs.google.com/javax/swing/text/html/parser/Parser.html#error(java.lang.String,%20java.lang.String,%20java.lang.String,%20java.lang.String))([String](http://docs.google.com/java/lang/String.html) err, [String](http://docs.google.com/java/lang/String.html) arg1, [String](http://docs.google.com/java/lang/String.html) arg2, [String](http://docs.google.com/java/lang/String.html) arg3)            Invoke the error handler. |
| protected  void | [**flushAttributes**](http://docs.google.com/javax/swing/text/html/parser/Parser.html#flushAttributes())() |
| protected  [SimpleAttributeSet](http://docs.google.com/javax/swing/text/SimpleAttributeSet.html) | [**getAttributes**](http://docs.google.com/javax/swing/text/html/parser/Parser.html#getAttributes())() |
| protected  int | [**getCurrentLine**](http://docs.google.com/javax/swing/text/html/parser/Parser.html#getCurrentLine())() |
| protected  int | [**getCurrentPos**](http://docs.google.com/javax/swing/text/html/parser/Parser.html#getCurrentPos())() |
| protected  void | [**handleComment**](http://docs.google.com/javax/swing/text/html/parser/Parser.html#handleComment(char%5B%5D))(char[] text)            Called when an HTML comment is encountered. |
| protected  void | [**handleEmptyTag**](http://docs.google.com/javax/swing/text/html/parser/Parser.html#handleEmptyTag(javax.swing.text.html.parser.TagElement))([TagElement](http://docs.google.com/javax/swing/text/html/parser/TagElement.html) tag)            Called when an empty tag is encountered. |
| protected  void | [**handleEndTag**](http://docs.google.com/javax/swing/text/html/parser/Parser.html#handleEndTag(javax.swing.text.html.parser.TagElement))([TagElement](http://docs.google.com/javax/swing/text/html/parser/TagElement.html) tag)            Called when an end tag is encountered. |
| protected  void | [**handleEOFInComment**](http://docs.google.com/javax/swing/text/html/parser/Parser.html#handleEOFInComment())() |
| protected  void | [**handleError**](http://docs.google.com/javax/swing/text/html/parser/Parser.html#handleError(int,%20java.lang.String))(int ln, [String](http://docs.google.com/java/lang/String.html) msg)            An error has occurred. |
| protected  void | [**handleStartTag**](http://docs.google.com/javax/swing/text/html/parser/Parser.html#handleStartTag(javax.swing.text.html.parser.TagElement))([TagElement](http://docs.google.com/javax/swing/text/html/parser/TagElement.html) tag)            Called when a start tag is encountered. |
| protected  void | [**handleText**](http://docs.google.com/javax/swing/text/html/parser/Parser.html#handleText(char%5B%5D))(char[] text)            Called when PCDATA is encountered. |
| protected  void | [**handleTitle**](http://docs.google.com/javax/swing/text/html/parser/Parser.html#handleTitle(char%5B%5D))(char[] text)            Called when an HTML title tag is encountered. |
| protected  [TagElement](http://docs.google.com/javax/swing/text/html/parser/TagElement.html) | [**makeTag**](http://docs.google.com/javax/swing/text/html/parser/Parser.html#makeTag(javax.swing.text.html.parser.Element))([Element](http://docs.google.com/javax/swing/text/html/parser/Element.html) elem) |
| protected  [TagElement](http://docs.google.com/javax/swing/text/html/parser/TagElement.html) | [**makeTag**](http://docs.google.com/javax/swing/text/html/parser/Parser.html#makeTag(javax.swing.text.html.parser.Element,%20boolean))([Element](http://docs.google.com/javax/swing/text/html/parser/Element.html) elem, boolean fictional)            Makes a TagElement. |
| protected  void | [**markFirstTime**](http://docs.google.com/javax/swing/text/html/parser/Parser.html#markFirstTime(javax.swing.text.html.parser.Element))([Element](http://docs.google.com/javax/swing/text/html/parser/Element.html) elem)            Marks the first time a tag has been seen in a document |
| void | [**parse**](http://docs.google.com/javax/swing/text/html/parser/Parser.html#parse(java.io.Reader))([Reader](http://docs.google.com/java/io/Reader.html) in)            Parse an HTML stream, given a DTD. |
| [String](http://docs.google.com/java/lang/String.html) | [**parseDTDMarkup**](http://docs.google.com/javax/swing/text/html/parser/Parser.html#parseDTDMarkup())()            Parses th Document Declaration Type markup declaration. |
| protected  boolean | [**parseMarkupDeclarations**](http://docs.google.com/javax/swing/text/html/parser/Parser.html#parseMarkupDeclarations(java.lang.StringBuffer))([StringBuffer](http://docs.google.com/java/lang/StringBuffer.html) strBuff)            Parse markup declarations. |
| protected  void | [**startTag**](http://docs.google.com/javax/swing/text/html/parser/Parser.html#startTag(javax.swing.text.html.parser.TagElement))([TagElement](http://docs.google.com/javax/swing/text/html/parser/TagElement.html) tag)            Handle a start tag. |

| **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** |
| --- |

### dtd

protected [DTD](http://docs.google.com/javax/swing/text/html/parser/DTD.html) **dtd**

### strict

protected boolean **strict**

This flag determines whether or not the Parser will be strict in enforcing SGML compatibility. If false, it will be lenient with certain common classes of erroneous HTML constructs. Strict or not, in either case an error will be recorded.

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

### Parser

public **Parser**([DTD](http://docs.google.com/javax/swing/text/html/parser/DTD.html) dtd)

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

### getCurrentLine

protected int **getCurrentLine**()

**Returns:**the line number of the line currently being parsed

### makeTag

protected [TagElement](http://docs.google.com/javax/swing/text/html/parser/TagElement.html) **makeTag**([Element](http://docs.google.com/javax/swing/text/html/parser/Element.html) elem,  
 boolean fictional)

Makes a TagElement.

### makeTag

protected [TagElement](http://docs.google.com/javax/swing/text/html/parser/TagElement.html) **makeTag**([Element](http://docs.google.com/javax/swing/text/html/parser/Element.html) elem)

### getAttributes

protected [SimpleAttributeSet](http://docs.google.com/javax/swing/text/SimpleAttributeSet.html) **getAttributes**()

### flushAttributes

protected void **flushAttributes**()

### handleText

protected void **handleText**(char[] text)

Called when PCDATA is encountered.

### handleTitle

protected void **handleTitle**(char[] text)

Called when an HTML title tag is encountered.

### handleComment

protected void **handleComment**(char[] text)

Called when an HTML comment is encountered.

### handleEOFInComment

protected void **handleEOFInComment**()

### handleEmptyTag

protected void **handleEmptyTag**([TagElement](http://docs.google.com/javax/swing/text/html/parser/TagElement.html) tag)  
 throws [ChangedCharSetException](http://docs.google.com/javax/swing/text/ChangedCharSetException.html)

Called when an empty tag is encountered.

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

### handleStartTag

protected void **handleStartTag**([TagElement](http://docs.google.com/javax/swing/text/html/parser/TagElement.html) tag)

Called when a start tag is encountered.

### handleEndTag

protected void **handleEndTag**([TagElement](http://docs.google.com/javax/swing/text/html/parser/TagElement.html) tag)

Called when an end tag is encountered.

### handleError

protected void **handleError**(int ln,  
 [String](http://docs.google.com/java/lang/String.html) msg)

An error has occurred.

### error

protected void **error**([String](http://docs.google.com/java/lang/String.html) err,  
 [String](http://docs.google.com/java/lang/String.html) arg1,  
 [String](http://docs.google.com/java/lang/String.html) arg2,  
 [String](http://docs.google.com/java/lang/String.html) arg3)

Invoke the error handler.

### error

protected void **error**([String](http://docs.google.com/java/lang/String.html) err,  
 [String](http://docs.google.com/java/lang/String.html) arg1,  
 [String](http://docs.google.com/java/lang/String.html) arg2)

### error

protected void **error**([String](http://docs.google.com/java/lang/String.html) err,  
 [String](http://docs.google.com/java/lang/String.html) arg1)

### error

protected void **error**([String](http://docs.google.com/java/lang/String.html) err)

### startTag

protected void **startTag**([TagElement](http://docs.google.com/javax/swing/text/html/parser/TagElement.html) tag)  
 throws [ChangedCharSetException](http://docs.google.com/javax/swing/text/ChangedCharSetException.html)

Handle a start tag. The new tag is pushed onto the tag stack. The attribute list is checked for required attributes.

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

### endTag

protected void **endTag**(boolean omitted)

Handle an end tag. The end tag is popped from the tag stack.

### markFirstTime

protected void **markFirstTime**([Element](http://docs.google.com/javax/swing/text/html/parser/Element.html) elem)

Marks the first time a tag has been seen in a document

### parseDTDMarkup

public [String](http://docs.google.com/java/lang/String.html) **parseDTDMarkup**()  
 throws [IOException](http://docs.google.com/java/io/IOException.html)

Parses th Document Declaration Type markup declaration. Currently ignores it.

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

### parseMarkupDeclarations

protected boolean **parseMarkupDeclarations**([StringBuffer](http://docs.google.com/java/lang/StringBuffer.html) strBuff)  
 throws [IOException](http://docs.google.com/java/io/IOException.html)

Parse markup declarations. Currently only handles the Document Type Declaration markup. Returns true if it is a markup declaration false otherwise.

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

### parse

public void **parse**([Reader](http://docs.google.com/java/io/Reader.html) in)  
 throws [IOException](http://docs.google.com/java/io/IOException.html)

Parse an HTML stream, given a DTD.

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

### getCurrentPos

protected int **getCurrentPos**()

| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/Parser.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/parser/Entity.html)   [**NEXT CLASS**](http://docs.google.com/javax/swing/text/html/parser/ParserDelegator.html) | [**FRAMES**](http://docs.google.com/index.html?javax/swing/text/html/parser/Parser.html)    [**NO FRAMES**](http://docs.google.com/Parser.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | [FIELD](#3znysh7) | [CONSTR](#tyjcwt) | [METHOD](#3dy6vkm) | DETAIL: [FIELD](#4d34og8) | [CONSTR](#3rdcrjn) | [METHOD](#lnxbz9) |

[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).