

# TeXDoclet Java Documentation

Created with Javadoc TeXDoclet Doclet

Greg Wonderly

Sören Caspersen

Stefan Marx

July 1, 2012

## Abstract

(content from file setup.tex)

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet.

## Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
<b>2</b>	<b>Package <code>org.stfm.texdoclet</code></b>	<b>1</b>
2.1	Interface <code>ClassFilter</code>	2
2.1.1	Declaration	2
2.1.2	All known subinterfaces	2
2.1.3	All classes known to implement interface	2
2.1.4	Methods	2
2.2	Class <code>ClassHierachy</code>	3
2.2.1	Declaration	3
2.2.2	Fields	3
2.2.3	Constructors	3
2.2.4	Methods	3
2.3	Class <code>HelpOutput</code>	4
2.3.1	Declaration	4
2.3.2	Constructors	4
2.3.3	Methods	4
2.4	Class <code>HTMLtoLaTeXBackEnd</code>	4
2.4.1	See also	4
2.4.2	Declaration	4

1	INTRODUCTION	2
2.4.3	Constructors	4
2.4.4	Methods	5
2.4.5	Members inherited from class <code>HTMLToolkit.ParserCallback</code>	5
2.5	Class <code>InterfaceHierachy</code>	6
2.5.1	Declaration	6
2.5.2	Fields	6
2.5.3	Constructors	6
2.5.4	Methods	6
2.6	Class <code>Package</code>	7
2.6.1	See also	7
2.6.2	Declaration	7
2.6.3	Fields	7
2.6.4	Constructors	7
2.6.5	Methods	8
2.7	Class <code>TableInfo</code>	8
2.7.1	Declaration	9
2.7.2	Constructors	9
2.7.3	Methods	9
2.8	Class <code>TestFilter</code>	11
2.8.1	Declaration	11
2.8.2	Constructors	11
2.8.3	Methods	11
2.9	Class <code>TeXDoclet</code>	11
2.9.1	See also	13
2.9.2	Declaration	13
2.9.3	Fields	13
2.9.4	Constructors	13
2.9.5	Methods	13
2.9.6	Members inherited from class <code>Doclet</code>	14
<b>3</b>	<b>Finish</b>	<b>14</b>
<b>A</b>	<b>File <code>appendix_a.html</code></b>	<b>14</b>
<b>B</b>	<b>File <code>appendix_b.txt</code></b>	<b>15</b>

## 1 Introduction

(content from file `intro.tex`)

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet.

## 2 Package org.stfm.texdoclet

*Package Contents*

*Page*

### Interfaces

<b>ClassFilter</b> .....	2
This interface can be implemented and a class name provided to the Doclet to filter which classes are and are not included in the output document.	

### Classes

<b>ClassHierarchy</b> .....	3
Manages and prints a class hierarchy.	
<b>HelpOutput</b> .....	4
<b>HTMLtoLaTeXBackEnd</b> .....	4
This class implements a <code>ParserCallback</code> that translates HTML to the corresponding $\LaTeX$ .	
<b>InterfaceHierarchy</b> .....	6
Manages and prints a interface hierarchy.	
<b>Package</b> .....	7
This class is used to manage the contents of a Java package.	
<b>TableInfo</b> .....	8
This class provides support for converting HTML tables into $\LaTeX$ tables.	
<b>TestFilter</b> .....	11
This class filters out classes beginning with "Test" when applied to the Doclet.	
<b>TeXDoclet</b> .....	11
This class provides a Java javadoc Doclet which generates a $\LaTeX 2_{\epsilon}$ document out of the java classes that it is used on.	

This doclet is based on the doclet originally created by Greg Wonderly of [C2 technologies Inc.](http://www.c2technologies.com) and its revision by [XO Software](http://www.xo.com). The project of Greg Wonderly is available here : <http://java.net/projects/texdoclet>.

### 2.1 Interface ClassFilter

This interface can be implemented and a class name provided to the Doclet to filter which classes are and are not included in the output document.

#### 2.1.1 Declaration

```
public interface ClassFilter
```

#### 2.1.2 All known subinterfaces

`TestFilter` (in 2.8, page 11)

### 2.1.3 All classes known to implement interface

TestFilter (in 2.8, page 11)

### 2.1.4 Methods

- **includeClass**

`boolean includeClass(com.sun.javadoc.ClassDoc cd)`

- **Description**

Filters the ClassDoc passed. If true is returned, the passed class will be included into the output. If false is returned, this document will not be included.

## 2.2 Class ClassHierachy

Manages and prints a class hierarchy. Use `add` to add another class to the hierarchy. Use `printTree` to print the corresponding  $\text{\LaTeX}$ .

### 2.2.1 Declaration

```
public class ClassHierachy
extends java.lang.Object
```

### 2.2.2 Fields

- `public java.util.SortedMap root`

### 2.2.3 Constructors

- **ClassHierachy**

`public ClassHierachy()`

- **Description**

Creates new ClassHierachy

### 2.2.4 Methods

- **add**

`protected java.util.SortedMap add(com.sun.javadoc.ClassDoc cls)`

- **Description**

Adds another class to the hierachy

- **printBranch**

`protected void printBranch(com.sun.javadoc.RootDoc rootDoc,  
java.util.SortedMap map, double indent, double overviewindent)`

- **Description**

Prints a branch of the tree. The branch is printed using `TeXDoclet.os`.

- **printTree**

```
public void printTree(com.sun.javadoc.RootDoc rootDoc, double  
overviewindent)
```

- **Description**

- Prints the  $\text{\LaTeX}$  corresponding to the tree. The tree is printed using `TeXDoclet.os`.

## 2.3 Class HelpOutput

### 2.3.1 Declaration

```
public class HelpOutput  
extends java.lang.Object
```

### 2.3.2 Constructors

- **HelpOutput**

```
public HelpOutput()
```

### 2.3.3 Methods

- **printHelp**

```
protected static void printHelp()
```

## 2.4 Class HTMLtoLaTeXBackEnd

This class implements a `ParserCallback` that translates HTML to the corresponding  $\text{\LaTeX}$ . Not all tags are processed but the most common are.

HTML links to files located in the `doc-files` directory (`appendix_a.html` (in [A](#), page 14), `appendix_b.txt` (in [B](#), page 15)) are transformed to references to the appendix, whereby the referenced files themselves are included in the appendix.

### 2.4.1 See also

- [javax.swing.text.html.parser.ParserDelegator](#)

### 2.4.2 Declaration

```
public class HTMLtoLaTeXBackEnd  
extends javax.swing.text.html.HTMLEditorKit.ParserCallback
```

### 2.4.3 Constructors

- **HTMLtoLaTeXBackEnd**

```
public HTMLtoLaTeXBackEnd(java.lang.StringBuffer ret)
```

- **Description**

- Constructs a new instance.

- **Parameters**

- \* `StringBuffer` – The `StringBuffer` where the translated HTML is appended.

#### 2.4.4 Methods

- **fixText**

```
public static java.lang.String fixText(java.lang.String str)
```

- **Description**

- Converts a HTML string into  $\text{\LaTeX}$  using an instance of `HTMLtoLaTeXBackEnd`.

- **handleEndTag**

```
public void handleEndTag(javax.swing.text.html.HTML.Tag tag, int pos)
```

- **Description**

- This method handles HTML tags that mark an ending (e.g. `</P>`-tags). It is called by the parser whenever such a tag is encountered.

- **handleSimpleTag**

```
public void handleSimpleTag(javax.swing.text.html.HTML.Tag tag,
    javax.swing.text.MutableAttributeSet attrSet, int pos)
```

- **Description**

- This method handles simple HTML tags (e.g. `<HR>`-tags). It is called by the parser whenever such a tag is encountered.

- **handleStartTag**

```
public void handleStartTag(javax.swing.text.html.HTML.Tag tag,
    javax.swing.text.MutableAttributeSet attrSet, int pos)
```

- **Description**

- This method handles HTML tags that mark a beginning (e.g. `<P>`-tags). It is called by the parser whenever such a tag is encountered.

- **handleText**

```
public void handleText(char[] data, int pos)
```

- **Description**

- This method handles all other text.

#### 2.4.5 Members inherited from class `HTMLEditorKit.ParserCallback`

```
javax.swing.text.html.HTMLEditorKit.ParserCallback
```

- `public void flush()` throws `javax.swing.text.BadLocationException`
- `public void handleComment(char[] arg0, int arg1)`
- `public void handleEndOfLineString(java.lang.String arg0)`
- `public void handleEndTag(HTML.Tag arg0, int arg1)`
- `public void handleError(java.lang.String arg0, int arg1)`
- `public void handleSimpleTag(HTML.Tag arg0, javax.swing.text.MutableAttributeSet arg1, int arg2)`
- `public void handleStartTag(HTML.Tag arg0, javax.swing.text.MutableAttributeSet arg1, int arg2)`
- `public void handleText(char[] arg0, int arg1)`
- `public static final IMPLIED`

## 2.5 Class InterfaceHierachy

Manages and prints a interface hierarchy. Use `add` to add another interface to the hierarchy. Use `printTree` to print the corresponding  $\LaTeX$ .

### 2.5.1 Declaration

```
public class InterfaceHierachy
extends java.lang.Object
```

### 2.5.2 Fields

- `public java.util.SortedMap root`

### 2.5.3 Constructors

- **InterfaceHierachy**  
`public InterfaceHierachy()`
  - **Description**  
Creates new InterfaceHierachy

### 2.5.4 Methods

- **add**  
`protected java.util.SortedMap add(com.sun.javadoc.ClassDoc cls)`
  - **Description**  
Adds another interface to the hierachy
- **printBranch**  
`protected void printBranch(com.sun.javadoc.RootDoc rootDoc, java.util.SortedMap map, double indent, double overviewindent)`
  - **Description**  
Prints a branch of the tree. The branch is printed using `TeXDoclet.os`.
- **printTree**  
`public void printTree(com.sun.javadoc.RootDoc rootDoc, double overviewindent)`
  - **Description**  
Prints the  $\LaTeX$  corresponding to the tree. The tree is printed using `TeXDoclet.os`.

## 2.6 Class Package

This class is used to manage the contents of a Java package. It accepts ClassDoc objects and examines them and groups them according to whether they are classes, interfaces, exceptions or errors. The accumulated Vectors can then be processed to get to all of the elements of the package that fall into each category. If needed the classes, interfaces, exceptions and errors can be sorted using the `sort` method.

### 2.6.1 See also

- [Package.sort\(\)](#) (in 2.6.5, page 8)

### 2.6.2 Declaration

```
public class Package
extends java.lang.Object
```

### 2.6.3 Fields

- protected `com.sun.javadoc.PackageDoc` **pkgDoc**
- protected `java.lang.String` **pkg**
  - The name of the package this object is for
- protected `java.util.Vector` **classes**
  - The classes this package has in it
- protected `java.util.Vector` **interfaces**
  - The interfaces this package has in it
- protected `java.util.Vector` **exceptions**
  - The exceptions this package has in it
- protected `java.util.Vector` **errors**
  - The errors this package has in it

### 2.6.4 Constructors

- **Package**

```
public Package(java.lang.String pkg, com.sun.javadoc.PackageDoc doc)
```

  - **Description**

Construct a new object corresponding to the passed package name.
  - **Parameters**
    - \* **pkg** – the package name to use



### 2.6.5 Methods

- **addElement**

```
public void addElement(com.sun.javadoc.ClassDoc cd)
```

- **Description**

Adds a ClassDoc element to this package.

- **Parameters**

\* cd – the object to add to this package

- **sort**

```
public void sort()
```

- **Description**

Sorts the vectors of classes, interfaces exceptions and errors.

## 2.7 Class TableInfo

This class provides support for converting HTML tables into L<sup>A</sup>T<sub>E</sub>X tables. Some of the things **NOT** implemented include the following:

- valign attributes are not processed, but align= is.
- rowspan attributes are not processed, but colspan= is.
- the argument to border= in the table tag is not used to control line size

Here is an example table.

Column 1 Heading	Column two heading	Column three heading
data	Span two columns	
<i>more data</i>	right	left

  

A nested table example		
Column one Heading	Column two heading	Column three heading
data	Span two columns	
<i>more data</i>	right	left
1	first line	
2	second line	
3	third line	
4	fourth line	

### 2.7.1 Declaration

```
public class TableInfo
extends java.lang.Object
```

### 2.7.2 Constructors

- **TableInfo**  
public **TableInfo**()

### 2.7.3 Methods

- **endCol**  
public void **endCol**()

- **Description**

Ends the current column.

- **Parameters**

- \* **ret** – The output buffer to put  $\text{\LaTeX} 2_{\epsilon}$  into.

- **endRow**

```
public void endRow()
```

- **Description**

Ends the current row.

- **Parameters**

- \* **ret** – The output buffer to put  $\text{\LaTeX} 2_{\epsilon}$  into.

- **endTable**

```
public java.lang.StringBuffer endTable()
```

- **Description**

Ends the table, closing the last row as needed

- **Parameters**

- \* **ret** – The output buffer to put  $\text{\LaTeX} 2_{\epsilon}$  into.

- **startCol**

```
public void startCol(javax.swing.text.MutableAttributeSet attrSet)
```

- **Description**

Starts a new column, possibly closing the current column if needed

- **Parameters**

- \* **ret** – The output buffer to put  $\text{\LaTeX} 2_{\epsilon}$  into.
- \* **p** – the properties from the `<td>` tag

- **startHeadCol**

```
public void startHeadCol(javax.swing.text.MutableAttributeSet attrSet)
```

- **Description**

Starts a new Heading column, possibly closing the current column if needed. A Heading column has a Bold Face font directive around it.

- **Parameters**

- \* **ret** – The output buffer to put  $\text{\LaTeX} 2_{\epsilon}$  into.
- \* **p** – The properties from the `<th>` tag

- **startRow**

```
public void startRow(javax.swing.text.MutableAttributeSet attrSet)
```

- **Description**

Starts a new row, possibly closing the current row if needed

- **Parameters**

- \* **ret** – The output buffer to put  $\text{\LaTeX}$  into.
- \* **p** – The properties from the `<tr>` tag

- **startTable**

```
public java.lang.StringBuffer startTable(java.lang.StringBuffer org,
    javax.swing.text.MutableAttributeSet attrSet)
```

- **Description**

Constructs a new table object and starts processing of the table by scanning the `<table>` passed to count columns.

- **Parameters**

- \* `p` – properties found on the `<table>` tag
- \* `ret` – the result buffer that will contain the output
- \* `table` – the input string that has the entire table definition in it.
- \* `off` – the offset into `<table>` where scanning should start

## 2.8 Class TestFilter

This class filters out classes beginning with "Test" when applied to the Doclet.

### 2.8.1 Declaration

```
public class TestFilter
extends java.lang.Object
implements ClassFilter
```

### 2.8.2 Constructors

- **TestFilter**  

```
public TestFilter()
```

### 2.8.3 Methods

- **includeClass**  

```
public boolean includeClass(com.sun.javadoc.ClassDoc cd)
```

- **Description**

Returns false if class name starts with "Test".

## 2.9 Class TeXDoclet

This class provides a Java javadoc Doclet which generates a L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub> document out of the java classes that it is used on. This is convenient for creating printable documentation complete with cross reference information.

### Supported HTML tags

`<a>` including an additional attribut "doprinturl". Since the output of the doclet should be printable, the href attribut of tags is printed in parentheses following the link if attribut "doprinturl" is set. Sometimes this is undesirable, and omitting "doprinturl" attribut will prevent this.

`<dl>` with the associated `<dt><dd></dl>` tags

<p> but not align=center...yet

<br> but not clear=xxx

<table> including all the associated <td><th><tr></td></th></tr>

<ol> ordered lists

<ul> unordered lists

<font> font coloring

<pre> preformatted text

<code> fixed point fonts

<i> italicized fonts

<b> bold fonts

<sub> subscript

<sup> superscript

<center> center

<img> image located in java sources ()

1. example converted from JPG:



2. example converted from GIF:



<img> image located in the www: (see image at <http://upload.wikimedia.org/wikipedia/commons/9/92/LaTeX...>)

### Extra tags

A new tag is defined: <TEX>. This tag is useful for passing T<sub>E</sub>X code directly to the T<sub>E</sub>X compiler. The following code:

```
<TEX txt="\[ F\left( x \right) = \int_{-\infty}^x \frac{1}{\sqrt{2\pi}} e^{-\frac{z^2}{2}} dz \]">
<BR><BR><B>This alternative text will appear if the javadoc/HTML is parsed
by any other doclet/browser</B><BR><BR></TEX>
```

will produce the following result:

$$F(x) = \int_{-\infty}^x \frac{1}{\sqrt{2\pi}} e^{-\frac{z^2}{2}} dz$$

The "alternative" text is ignored by the TeXDoclet, but useful if you want to use both the TeXDoclet and a regular HTML based doclet.

### 2.9.1 See also

- [HTMLtoLaTeXBackEnd](#) (in 2.4, page 4)
- [TeXDoclet.start\(RootDoc\)](#) (in 2.9.5, page 14)

### 2.9.2 Declaration

```
public class TeXDoclet
extends com.sun.javadoc.Doclet
```

### 2.9.3 Fields

- public static final java.lang.String **SECTION\_LEVEL**
- public static final java.lang.String **CHAPTER\_LEVEL**
- public static final java.lang.String **SUBSECTION\_LEVEL**
- public static final java.lang.String **BOLD**
- public static final java.lang.String **TRUETYPE**
- public static final java.lang.String **ITALIC**
- public static java.io.PrintWriter **os**
  - Writer for writing to output file

### 2.9.4 Constructors

- **TeXDoclet**  
public **TeXDoclet()**

### 2.9.5 Methods

- **finish**  
public static void **finish()**
- **init**  
public static void **init()**
- **initSections**  
public static void **initSections()**
- **main**  
public static void **main**(java.lang.String[] args)
- **optionLength**  
public static int **optionLength**(java.lang.String option)
  - **Description**  
Returns how many arguments would be consumed if option is a recognized option.

- **Parameters**

- \* `option` – the option to check

- **start**

```
public static boolean start(com.sun.javadoc.RootDoc root)
```

- **Description**

- Called by the framework to format the entire document

- **Parameters**

- \* `root` – the root of the starting document

- **validOptions**

```
public static boolean validOptions(java.lang.String[] [] args,
com.sun.javadoc.DocErrorReporter err)
```

- **Description**

- Checks the passed options and their arguments for validity.

- **Parameters**

- \* `args` – the arguments to check

- \* `err` – the interface to use for reporting errors

### 2.9.6 Members inherited from class Doclet

`com.sun.javadoc.Doclet`

- `public static LanguageVersion languageVersion()`
- `public static int optionLength(java.lang.String arg0)`
- `public static boolean start(RootDoc arg0)`
- `public static boolean validOptions(java.lang.String[] [] arg0, DocErrorReporter arg1)`

## 3 Finish

(content from file `finish.tex`)

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet.

## A File appendix\_a.html

### Appendix A content

content of file `doc-files/appendix_a.html`

## **B File appendix\_b.txt**

content of file doc-files/appendix\_b.txt