TeXDoclet Java Documentation

Created with Javadoc TeXDoclet Doclet

Greg Wonderly nd S"oren Caspersen nd Stefan Marx

March 26, 2013

Contents

Class Hierarchy **2**

1 Package org.stfm.texdoclet **3**
- 1.1 Interface ClassFilter . . . 4
  - 1.1.1 Declaration . . . 4
  - 1.1.2 All known subinterfaces . . . 4
  - 1.1.3 All classes known to implement interface . . . 4
  - 1.1.4 Method summary . . . 4
  - 1.1.5 Methods . . . 4
- 1.2 Class ClassHierachy . . . 4
  - 1.2.1 Declaration . . . 4
  - 1.2.2 Field summary . . . 4
  - 1.2.3 Constructor summary . . . 4
  - 1.2.4 Method summary . . . 5
  - 1.2.5 Fields . . . 5
  - 1.2.6 Constructors . . . 5
  - 1.2.7 Methods . . . 5
- 1.3 Class HelpOutput . . . 5
  - 1.3.1 Declaration . . . 5
  - 1.3.2 Constructor summary . . . 6
  - 1.3.3 Method summary . . . 6
  - 1.3.4 Constructors . . . 6
  - 1.3.5 Methods . . . 6
- 1.4 Class HTMLtoLaTeXBackEnd . . . 6
  - 1.4.1 See also . . . 6
  - 1.4.2 Declaration . . . 6
  - 1.4.3 Constructor summary . . . 6
  - 1.4.4 Method summary . . . 6
  - 1.4.5 Constructors . . . 7
  - 1.4.6 Methods . . . 7
  - 1.4.7 Members inherited from class HTMLEditorKit.ParserCallback . . . 8
- 1.5 Class InterfaceHierachy . . . 8
  - 1.5.1 Declaration . . . 8
  - 1.5.2 Field summary . . . 8
  - 1.5.3 Constructor summary . . . 8
  - 1.5.4 Method summary . . . 8

1.5.5 Fields . . . 8
1.5.6 Constructors . . . 8
1.5.7 Methods . . . 8
1.6 Class Package . . . 9
1.6.1 See also . . . 9
1.6.2 Declaration . . . 9
1.6.3 Field summary . . . 9
1.6.4 Constructor summary . . . 9
1.6.5 Method summary . . . 10
1.6.6 Fields . . . 10
1.6.7 Constructors . . . 10
1.6.8 Methods . . . 10
1.7 Class TableInfo . . . 11
1.7.1 Declaration . . . 11
1.7.2 Constructor summary . . . 11
1.7.3 Method summary . . . 12
1.7.4 Constructors . . . 12
1.7.5 Methods . . . 12
1.8 Class TestFilter . . . 13
1.8.1 Declaration . . . 13
1.8.2 Constructor summary . . . 13
1.8.3 Method summary . . . 13
1.8.4 Constructors . . . 13
1.8.5 Methods . . . 14
1.9 Class TeXDoclet . . . 14
1.9.1 See also . . . 16
1.9.2 Declaration . . . 16
1.9.3 Field summary . . . 16
1.9.4 Constructor summary . . . 17
1.9.5 Method summary . . . 17
1.9.6 Fields . . . 17
1.9.7 Constructors . . . 17
1.9.8 Methods . . . 17
1.9.9 Members inherited from class Doclet . . . 18

A File appendix_a.html **19**

B File appendix_b.txt **20**

Class Hierarchy

Classes

- java.lang.Object
  - com.sun.javadoc.Doclet
    - org.stfm.texdoclet.TeXDoclet (in 1.9, page 14)
  - javax.swing.text.html.HTMLEditorKit.ParserCallback
    - org.stfm.texdoclet.HTMLtoLaTeXBackEnd (in 1.4, page 6)
  - org.stfm.texdoclet.ClassHierachy (in 1.2, page 4)
  - org.stfm.texdoclet.HelpOutput (in 1.3, page 5)
  - org.stfm.texdoclet.InterfaceHierachy (in 1.5, page 8)
  - org.stfm.texdoclet.Package (in 1.6, page 9)
  - org.stfm.texdoclet.TableInfo (in 1.7, page 11)
  - org.stfm.texdoclet.TestFilter (in 1.8, page 13)

Interfaces

- org.stfm.texdoclet.ClassFilter (in 1.1, page 4)

Chapter 1

Package org.stfm.texdoclet

Package Contents *Page*

Interfaces

ClassFilter 4

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

ClassHierachy 4

Manages and prints a class hierarchy.

HelpOutput 5

HTMLtoLaTeXBackEnd 6

This class implements a `ParserCallback` that translates HTML to the corresponding LaTeX.

InterfaceHierachy 8

Manages and prints a interface hierarchy.

Package 9

This class is used to manage the contents of a Java package.

TableInfo 11

This class provides support for converting HTML tables into LaTeX tables.

TestFilter 13

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

TeXDoclet 14

This class provides a Java `javadoc` Doclet which generates a LaTeX 2_ε 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. and its revision by XO Software. The project of Greg Wonderly is available here : http://java.net/projects/texdoclet.

1.1 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.

1.1.1 Declaration

public interface ClassFilter

1.1.2 All known subinterfaces

TestFilter (in 1.8, page 13)

1.1.3 All classes known to implement interface

TestFilter (in 1.8, page 13)

1.1.4 Method summary

includeClass(ClassDoc) Filters the ClassDoc passed.

1.1.5 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.

1.2 Class ClassHierachy

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

1.2.1 Declaration

public class ClassHierachy
extends java.lang.Object

1.2.2 Field summary

root

1.2.3 Constructor summary

ClassHierachy() Creates new ClassHierachy

1.2.4 Method summary

add(ClassDoc) Adds another class to the hierachy
printBranch(RootDoc, SortedMap, double, double) Prints a branch of the tree.
printTree(RootDoc, double) Prints the LaTeX corresponding to the tree.

1.2.5 Fields

- public java.util.SortedMap **root**

1.2.6 Constructors

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

1.2.7 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 LaTeX corresponding to the tree. The tree is printed using `TeXDoclet.os`.

1.3 Class HelpOutput

1.3.1 Declaration

public class HelpOutput
extends java.lang.Object

1.3.2 Constructor summary

HelpOutput()

1.3.3 Method summary

printHelp()

1.3.4 Constructors

- **HelpOutput**
  `public` **HelpOutput**`()`

1.3.5 Methods

- **printHelp**
  `protected static void` **printHelp**`()`

1.4 Class HTMLtoLaTeXBackEnd

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

HTML links to files located in the doc-files directory (appendix_a.html (in A, page 19), appendix_b.txt (in B, page 20)) are transformed to references to the appendix, whereby the referenced files itself are included in the appendix.

1.4.1 See also

– `javax.swing.text.html.parser.ParserDelegator`

1.4.2 Declaration

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

1.4.3 Constructor summary

HTMLtoLaTeXBackEnd(StringBuffer) Constructs a new instance.

1.4.4 Method summary

fixText(String) Converts a HTML string into LaTeX using an instance of `HTMLtoLaTeXBackEnd`.
handleEndTag(HTML.Tag, int) This method handles HTML tags that mark an ending (e.g.
handleSimpleTag(HTML.Tag, MutableAttributeSet, int) This method handles simple HTML tags (e.g.

handleStartTag(HTML.Tag, MutableAttributeSet, int) This method handles HTML tags that mark a beginning (e.g.
handleText(char[], int) This method handles all other text.

1.4.5 Constructors

- **HTMLtoLaTeXBackEnd**
  `public` **HTMLtoLaTeXBackEnd**`(java.lang.StringBuffer` **stringBuffer**`)`
  - **Description**
    Constructs a new instance.
  - **Parameters**
    * `stringBuffer` – The `StringBuffer` where the translated HTML is appended.

1.4.6 Methods

- **fixText**
  `public static java.lang.String` **fixText**`(java.lang.String` **str**`)`
  - **Description**
    Converts a HTML string into LaTeX using an instance of `HTMLtoLaTeXBackEnd`.
  - **Returns** – The converted string.
- **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.

1.4.7 Members inherited from class HTMLEditorKit.ParserCallback

`javax.swing.text.html.HTMLEditorKit.ParserCallback`
flush, handleComment, handleEndOfLineString, handleEndTag, handleError, handleSimpleTag, handleStartTag, handleText, IMPLIED

1.5 Class InterfaceHierachy

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

1.5.1 Declaration

public class InterfaceHierachy
extends java.lang.Object

1.5.2 Field summary

root

1.5.3 Constructor summary

InterfaceHierachy() Creates new InterfaceHierachy

1.5.4 Method summary

add(ClassDoc) Adds another interface to the hierachy
printBranch(RootDoc, SortedMap, double, double) Prints a branch of the tree.
printTree(RootDoc, double) Prints the LaTeX corresponding to the tree.

1.5.5 Fields

- public java.util.SortedMap **root**

1.5.6 Constructors

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

1.5.7 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`.

1.6 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.

1.6.1 See also

– `Package.sort()` (in 1.6.8, page 10)

1.6.2 Declaration

public class Package
extends java.lang.Object

1.6.3 Field summary

classes The classes this package has in it
errors The errors this package has in it
exceptions The exceptions this package has in it
interfaces The interfaces this package has in it
pkg The name of the package this object is for
pkgDoc

1.6.4 Constructor summary

Package(String, PackageDoc) Construct a new object corresponding to the passed package name.

1.6.5 Method summary

addElement(ClassDoc) Adds a ClassDoc element to this package.
sort() Sorts the vectors of classes, interfaces exceptions and errors.

1.6.6 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

1.6.7 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

1.6.8 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.

1.7 Class TableInfo

This class provides support for converting HTML tables into LaTeX 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	
more data	right	left

A nested table example		
Column one Heading	**Column two heading**	**Column three heading**
data	Span two columns	
more data	right	left

```
1
 2
3
  4
```
	```
	first line
	second line
	third line
	fourth line
	```

1.7.1 Declaration

public class TableInfo
extends java.lang.Object

1.7.2 Constructor summary

TableInfo()

1.7.3 Method summary

endCol() Ends the current column.
endRow() Ends the current row.
endTable() Ends the table, closing the last row as needed
startCol(MutableAttributeSet) Starts a new column, possibly closing the current column if needed //@param ret The output buffer to put LaTeX 2$_\varepsilon$ into.
startHeadCol(MutableAttributeSet) Starts a new Heading column, possibly closing the current column if needed.
startRow(MutableAttributeSet) Starts a new row, possibly closing the current row if needed //@param ret The output buffer to put LaTeX into.
startTable(StringBuffer, MutableAttributeSet) Constructs a new table object and starts processing of the table by scanning the <`table`> passed to count columns.

1.7.4 Constructors

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

1.7.5 Methods

- **endCol**
  `public void` **endCol**`()`
  - **Description**
    Ends the current column. //@param ret The output buffer to put LaTeX 2$_\varepsilon$ into.
- **endRow**
  `public void` **endRow**`()`
  - **Description**
    Ends the current row. // @param ret The output buffer to put LaTeX 2$_\varepsilon$ into.
- **endTable**
  `public java.lang.StringBuffer` **endTable**`()`
  - **Description**
    Ends the table, closing the last row as needed
- **startCol**
  `public void` **startCol**`(javax.swing.text.MutableAttributeSet` **attrSet**`)`
  - **Description**
    Starts a new column, possibly closing the current column if needed //@param ret The output buffer to put LaTeX 2$_\varepsilon$ into. //@param 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. //@param ret The output buffer to put $\LaTeX\,2_\varepsilon$ into. //@param 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 //@param ret The output buffer to put $\LaTeX$ into. //@param 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. //@param p // properties found on the <`table`> tag //@param ret // the result buffer that will contain the output //@param table // the input string that has the entire table definition in it. //@param off // the offset into <`table`> where scanning // should start

1.8 Class TestFilter

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

1.8.1 Declaration

public class TestFilter
extends java.lang.Object
implements ClassFilter

1.8.2 Constructor summary

TestFilter()

1.8.3 Method summary

includeClass(ClassDoc) Returns false if class name starts with "Test".

1.8.4 Constructors

- **TestFilter**
  `public` **TestFilter**`()`

1.8.5 Methods

- **includeClass**
  `public boolean` **`includeClass`**`(com.sun.javadoc.ClassDoc` **`cd`**`)`
  - **Description**
    Returns false if class name starts with "Test".

1.9 Class TeXDoclet

This class provides a Java `javadoc` Doclet which generates a LaTeX 2_ε 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>	italized fonts
<b>	bold fonts
<sub>	subscript
<sup>	superscript
<center>	center
<img>	image located in java sources (<img src="package path/image name">)

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

<TEX>

A new tag is defined: <TEX>. This tag is useful for passing TEX code directly to the TEX 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:

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

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

<PRE format="markdown">

Instead of writing your java documentation in often hard to read HTML code you can make use of Markdown syntax. The HTML <PRE> tag is used therefore to prevent your IDE from automatically reordering your Markdown documentation text. Markdown parsing is based on the Pegdown implementation. The following code :

```
<PRE format="markdown">

some text some text some text some text some text some text some text

##### Lists

- item1
    1. item11
    2. item12
- item1

##### Text formatting
```

```
_emphasis_ and __strong__ and some ‘code‘ :

    code line 1
    code line 2

some text some text some text some text some text some text some text
```

<PRE>

will produce the following :

some text some text some text some text some text some text some text

Lists

- item1
    1. item11
    2. item12
- item1

Text formatting

emphasis and **strong** and some `code` :

```
code line 1
code line 2
```

some text some text some text some text some text some text some text

1.9.1 See also

- `HTMLtoLaTeXBackEnd` (in 1.4, page 6)
- `TeXDoclet.start(RootDoc)` (in 1.9.8, page 18)

1.9.2 Declaration

public class TeXDoclet
extends com.sun.javadoc.Doclet

1.9.3 Field summary

BOLD
CHAPTER_LEVEL
ITALIC
os Writer for writing to output file
SECTION_LEVEL
SUBSECTION_LEVEL
TRUETYPE

1.9.4 Constructor summary

TeXDoclet()

1.9.5 Method summary

main(String[])
optionLength(String) Doclet class method that returns how many arguments would be consumed if `option` is a recognized option.
start(RootDoc) Doclet class method that is called by the framework to format the entire document
validOptions(String[][], DocErrorReporter) Doclet class method that checks the passed options and their arguments for validity.

1.9.6 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

1.9.7 Constructors

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

1.9.8 Methods

- **main**
  `public static void` **main**`(java.lang.String[]` **args**`)`
- **optionLength**
  `public static int` **optionLength**`(java.lang.String` **option**`)`
  – **Description**
    Doclet class method that 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**
    Doclet class method that is 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**
    Doclet class method that checks the passed options and their arguments for validity.
  - **Parameters**
    * `args` – the arguments to check
    * `err` – the interface to use for reporting errors

1.9.9 Members inherited from class Doclet

`com.sun.javadoc.Doclet`
languageVersion, optionLength, start, validOptions

Appendix A

File appendix_a.html

Appendix A content
content of file doc-files/appendix_a.html

Appendix B

File appendix_b.txt

content of file doc-files/appendix_b.txt