Contents

Chapter 1

Package org.wonderly.doclets

Package Contents *Page*

Interfaces

Classes

1.1 Interfaces

1.1.1 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.

Declaration

public interface ClassFilter

Methods

- *includeClass*
  `public boolean` **includeClass**`( com.sun.javadoc.ClassDoc` **cd** `)`
  - **Usage**
    - 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 Classes

1.2.1 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 catagory.

Declaration

public class Package
extends java.lang.Object

Constructors

- *Package*
  `public` **Package**`( java.lang.String` **pkg** `)`
  - **Usage**

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

METHODS

- *addElement*
  `public void` **addElement**`( com.sun.javadoc.ClassDoc` **cd** `)`
  - **Usage**
    - Adds a ClassDoc element to this package.
  - **Parameters**
    - `cd` - the object to add to this package

1.2.2 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 procesed, 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.

<table>
<tr><th>Column 1 Heading</th><th>Column two heading</th><th>Column three heading</th></tr>
<tr><td>data</td><td colspan="2">Span two columns</td></tr>
<tr><td><i>more data</i></td><td align="right">right</td><td>left</td></tr>
<tr><td colspan="3">
<table>
<tr><th colspan="3">A nested table example</th></tr>
<tr><th>Column 1 Heading</th><th>Column two heading</th><th>Column three heading</th></tr>
<tr><td>data</td><td colspan="2">Span two columns</td></tr>
<tr><td><i>more data</i></td><td align="right">right</td><td>left</td></tr>
<tr><td><code>1
  2
  3
  4</code></td><td><code>first line
  second line
  third line
  fourth line</code></td><td></td></tr>
</table>
</td></tr>
</table>

Declaration

public class TableInfo
extends java.lang.Object

Constructors

- *TableInfo*
  `public` **TableInfo**`( java.util.Properties` **p**`, java.lang.StringBuffer` **ret**`, java.lang.String` **table**`, int` **off** `)`
  - **Usage**
    - 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

Methods

- *endCol*
  `public void` **endCol**`( java.lang.StringBuffer` **ret** `)`
  - **Usage**
    - Ends the current column.
  - **Parameters**
    - `ret` - the output buffer to put LaTeX into
- *endRow*
  `public void` **endRow**`( java.lang.StringBuffer` **ret** `)`
  - **Usage**
    - Ends the current row.
  - **Parameters**
    - `ret` - the output buffer to put LaTeX into
- *endTable*
  `public void` **endTable**`( java.lang.StringBuffer` **ret** `)`
  - **Usage**
    - Ends the table, closing the last row as needed
  - **Parameters**

        - `ret` - the output buffer to put LaTeX into

- *startCol*
  `public void` **startCol**`( java.lang.StringBuffer` **ret**`, java.util.Properties` **p** `)`
  - **Usage**
    - Starts a new column, possibly closing the current column if needed
  - **Parameters**
    - `ret` - the output buffer to put LaTeX into
    - `p` - the properties from the <`td`> tag

- *startHeadCol*
  `public void` **startHeadCol**`( java.lang.StringBuffer` **ret**`, java.util.Properties` **p** `)`
  - **Usage**
    - 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 LaTeX into
    - `p` - the properties from the <`th`> tag

- *startRow*
  `public void` **startRow**`( java.lang.StringBuffer` **ret**`, java.util.Properties` **p** `)`
  - **Usage**
    - Starts a new row, possibly closing the current row if needed
  - **Parameters**
    - `ret` - the output buffer to put LaTeX into
    - `p` - the properties from the <`tr`> tag

1.2.3 Class **TexDoclet**

This class provides a Java 2, `javadoc` Doclet which generates a LaTeX2e document out of the java classes that it is used on. This is convienient for creating printable documentation complete with cross reference information.

Supported HTML tags within comments include the following

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

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

<br> but not clear=xxx

<table> including all the assocated <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

TexDoclet start

Declaration

public class TexDoclet
extends com.sun.javadoc.Doclet

Fields

- public static PrintWriter os
  - Writer for writing to output file

Constructors

- *TexDoclet*
  `public` **TexDoclet**`( )`

Methods

- *optionLength*
  `public static int` **optionLength**`( java.lang.String` **option** `)`
  - **Usage**
    - 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** `)`
  - **Usage**
    - 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** `)`
  - **Usage**
    - Checks the passed options and their arguments for validity.
  - **Parameters**
    - `args` - the arguments to check
    - `err` - the interface to use for reporting errors

Methods inherited from class `com.sun.javadoc.Doclet`

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