TeXDoclet Java Documentation

Created with Javadoc TeXDoclet Doclet

Greg Wonderly Soeren Caspersen Stefan Marx

June 16, 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

1 Introduction **1**

2 Package org.stfm.texdoclet **1**

2.1 Interface ClassFilter . . . 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 ClassHierachy . . . 3
2.2.1 Declaration . . . 3
2.2.2 Fields . . . 3
2.2.3 Constructors . . . 3
2.2.4 Methods . . . 3
2.3 Class HelpOutput . . . 4
2.3.1 Declaration . . . 4
2.3.2 Constructors . . . 4
2.3.3 Methods . . . 4
2.4 Class HTMLtoLaTeXBackEnd . . . 4
2.4.1 See also . . . 4
2.4.2 Declaration . . . 4

2.4.3 Constructors . . . 4
2.4.4 Methods . . . 4
2.4.5 Members inherited from class HTMLEditorKit.ParserCallback . . . 5
2.5 Class InterfaceHierachy . . . 5
2.5.1 Declaration . . . 6
2.5.2 Fields . . . 6
2.5.3 Constructors . . . 6
2.5.4 Methods . . . 6
2.6 Class Package . . . 6
2.6.1 See also . . . 6
2.6.2 Declaration . . . 7
2.6.3 Fields . . . 7
2.6.4 Constructors . . . 7
2.6.5 Methods . . . 7
2.7 Class TableInfo . . . 8
2.7.1 Declaration . . . 8
2.7.2 Constructors . . . 9
2.7.3 Methods . . . 9
2.8 Class TestFilter . . . 10
2.8.1 Declaration . . . 10
2.8.2 Constructors . . . 10
2.8.3 Methods . . . 10
2.9 Class TeXDoclet . . . 11
2.9.1 See also . . . 12
2.9.2 Declaration . . . 12
2.9.3 Fields . . . 12
2.9.4 Constructors . . . 12
2.9.5 Methods . . . 12
2.9.6 Members inherited from class Doclet . . . 13

3 Package com.keypoint **13**
3.1 Class PngEncoder . . . 14
3.1.1 Declaration . . . 14
3.1.2 All known subclasses . . . 14
3.1.3 Fields . . . 14
3.1.4 Constructors . . . 15
3.1.5 Methods . . . 16
3.2 Class PngEncoderB . . . 20
3.2.1 Declaration . . . 20
3.2.2 Fields . . . 21
3.2.3 Constructors . . . 21
3.2.4 Methods . . . 22
3.2.5 Members inherited from class PngEncoder . . . 23

4 Finish **24**

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

ClassFilter 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

ClassHierachy 3
Manages and prints a class hierarchy.

HelpOutput 4

HTMLtoLaTeXBackEnd 4
This class implements a `ParserCallback` that translates HTML to the corresponding LaTeX.

InterfaceHierachy 5
Manages and prints a interface hierarchy.

Package 6
This class is used to manage the contents of a Java package.

TableInfo 8
This class provides support for converting HTML tables into LaTeX tables.

TestFilter 10
This class filters out classes beginning with "Test" when applied to the Doclet.

TeXDoclet 11
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.

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 10)

2.1.3 All classes known to implement interface

TestFilter (in 2.8, page 10)

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 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 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 LaTeX. Not all tags a processed but the most common are.

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 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 (eg. <`/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 7)

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 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
(nested table below)		

A nested table example		
Column 1 Heading	**Coludliadfuapofd a fia fopia foipapio dupoau foapoifd pdpfiu apsd oipoioaofiduaopiufopiiiiiiiiiimn 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`	

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 LaTeX 2ε into.

- **endRow**
  `public void` **`endRow`**`()`
  - **Description**
    Ends the current row.
  - **Parameters**
    - `ret` – The output buffer to put LaTeX 2ε into.

- **endTable**
  `public java.lang.StringBuffer` **`endTable`**`()`
  - **Description**
    Ends the table, closing the last row as needed
  - **Parameters**
    - `ret` – The output buffer to put LaTeX 2ε 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 LaTeX 2ε 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 LaTeX 2ε 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 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 $\LaTeX\,2_\varepsilon$ 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

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.

2.9.1 See also

– `HTMLtoLaTeXBackEnd` (in 2.4, page 4)
– `TeXDoclet.start(RootDoc)` (in 2.9.5, page 13)

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 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 Package com.keypoint

Package Contents *Page*

Classes
PngEncoder 14

PngEncoderB 20

3.1 Class PngEncoder

3.1.1 Declaration

public class PngEncoder
extends java.lang.Object

3.1.2 All known subclasses

PngEncoderB (in 3.2, page 20)

3.1.3 Fields

- public static final boolean **ENCODE_ALPHA**
  - Constant specifying that alpha channel should be encoded.
- public static final boolean **NO_ALPHA**
  - Constant specifying that alpha channel should not be encoded.
- public static final int **FILTER_NONE**
  - Constants for filters
- public static final int **FILTER_SUB**
- public static final int **FILTER_UP**
- public static final int **FILTER_LAST**
- protected byte **pngBytes**
- protected byte **priorRow**
- protected byte **leftBytes**
- protected java.awt.Image **image**
- protected int **width**
- protected int **height**
- protected int **bytePos**
- protected int **maxPos**
- protected int **hdrPos**
- protected int **dataPos**
- protected int **endPos**
- protected java.util.zip.CRC32 **crc**
- protected long **crcValue**
- protected boolean **encodeAlpha**
- protected int **filter**
- protected int **bytesPerPixel**
- protected int **compressionLevel**

3.1.4 Constructors

- **PngEncoder**
  `public` **PngEncoder**`()`
  - **Description**
    Class constructor
- **PngEncoder**
  `public` **PngEncoder**`(java.awt.Image` **image**`)`
  - **Description**
    Class constructor specifying Image to encode, with no alpha channel encoding.
  - **Parameters**
    - `image` – A Java Image object which uses the DirectColorModel
  - **See also**
    - `java.awt.Image`
- **PngEncoder**
  `public` **PngEncoder**`(java.awt.Image` **image**`, boolean` **encodeAlpha**`)`
  - **Description**
    Class constructor specifying Image to encode, and whether to encode alpha.
  - **Parameters**
    - `image` – A Java Image object which uses the DirectColorModel
    - `encodeAlpha` – Encode the alpha channel? false=no; true=yes
  - **See also**
    - `java.awt.Image`
- **PngEncoder**
  `public` **PngEncoder**`(java.awt.Image` **image**`, boolean` **encodeAlpha**`, int` **whichFilter**`)`
  - **Description**
    Class constructor specifying Image to encode, whether to encode alpha, and filter to use.
  - **Parameters**
    - `image` – A Java Image object which uses the DirectColorModel
    - `encodeAlpha` – Encode the alpha channel? false=no; true=yes
    - `whichFilter` – 0=none, 1=sub, 2=up
  - **See also**
    - `java.awt.Image`
- **PngEncoder**
  `public` **PngEncoder**`(java.awt.Image` **image**`, boolean` **encodeAlpha**`, int` **whichFilter**`, int` **compLevel**`)`

- **Description**

  Class constructor specifying Image source to encode, whether to encode alpha, filter to use, and compression level.
- **Parameters**
  - `image` – A Java Image object
  - `encodeAlpha` – Encode the alpha channel? false=no; true=yes
  - `whichFilter` – 0=none, 1=sub, 2=up
  - `compLevel` – 0..9
- **See also**
  - `java.awt.Image`

3.1.5 Methods

- **filterSub**

  `protected void` **filterSub**`(byte[]` **pixels**`, int` **startPos**`, int` **width**`)`
  - **Description**

    Perform "sub" filtering on the given row. Uses temporary array leftBytes to store the original values of the previous pixels. The array is 16 bytes long, which will easily hold two-byte samples plus two-byte alpha.
  - **Parameters**
    - `pixels` – The array holding the scan lines being built
    - `startPos` – Starting position within pixels of bytes to be filtered.
    - `width` – Width of a scanline in pixels.
- **filterUp**

  `protected void` **filterUp**`(byte[]` **pixels**`, int` **startPos**`, int` **width**`)`
  - **Description**

    Perform "up" filtering on the given row. Side effect: refills the prior row with current row
  - **Parameters**
    - `pixels` – The array holding the scan lines being built
    - `startPos` – Starting position within pixels of bytes to be filtered.
    - `width` – Width of a scanline in pixels.
- **getCompressionLevel**

  `public int` **getCompressionLevel**`()`
  - **Description**

    Retrieve compression level
  - **Returns** – int in range 0-9
- **getEncodeAlpha**

  `public boolean` **getEncodeAlpha**`()`

- **Description**
  Retrieve alpha encoding status.
- **Returns** – boolean false=no, true=yes

• **getFilter**
`public int getFilter()`

- **Description**
  Retrieve filtering scheme
- **Returns** – int (see constant list)

• **pngEncode**
`public byte[] pngEncode()`

- **Description**
  Creates an array of bytes that is the PNG equivalent of the current image. Alpha encoding is determined by its setting in the constructor.
- **Returns** – an array of bytes, or null if there was a problem

• **pngEncode**
`public byte[] pngEncode(boolean encodeAlpha)`

- **Description**
  Creates an array of bytes that is the PNG equivalent of the current image, specifying whether to encode alpha or not.
- **Parameters**
  - `encodeAlpha` – boolean false=no alpha, true=encode alpha
- **Returns** – an array of bytes, or null if there was a problem

• **resizeByteArray**
`protected byte[] resizeByteArray(byte[] array, int newLength)`

- **Description**
  Increase or decrease the length of a byte array.
- **Parameters**
  - `array` – The original array.
  - `newLength` – The length you wish the new array to have.
- **Returns** – Array of newly desired length. If shorter than the original, the trailing elements are truncated.

• **setCompressionLevel**
`public void setCompressionLevel(int level)`

- **Description**
  Set the compression level to use
- **Parameters**

    * `level` – 0 through 9

- **setEncodeAlpha**
  `public void` **setEncodeAlpha**`(boolean` **encodeAlpha**`)`
  - **Description**
    Set the alpha encoding on or off.
  - **Parameters**
    * `encodeAlpha` – false=no, true=yes
- **setFilter**
  `public void` **setFilter**`(int` **whichFilter**`)`
  - **Description**
    Set the filter to use
  - **Parameters**
    * `whichFilter` – from constant list
- **setImage**
  `public void` **setImage**`(java.awt.Image` **image**`)`
  - **Description**
    Set the image to be encoded
  - **Parameters**
    * `image` – A Java Image object which uses the DirectColorModel
  - **See also**
    * `java.awt.Image`
    * `java.awt.image.DirectColorModel`
- **writeByte**
  `protected int` **writeByte**`(int` **b**`, int` **offset**`)`
  - **Description**
    Write a single byte into the pngBytes array at a given position.
  - **Parameters**
    * `n` – The integer to be written into pngBytes.
    * `offset` – The starting point to write to.
  - **Returns** – The next place to be written to in the pngBytes array.
- **writeBytes**
  `protected int` **writeBytes**`(byte[]` **data**`, int` **offset**`)`
  - **Description**
    Write an array of bytes into the pngBytes array. Note: This routine has the side effect of updating maxPos, the largest element written in the array. The array is resized by 1000 bytes or the length of the data to be written, whichever is larger.

  - **Parameters**
    - `data` – The data to be written into pngBytes.
    - `offset` – The starting point to write to.
  - **Returns** – The next place to be written to in the pngBytes array.

- **writeBytes**
  `protected int` **`writeBytes`**`(byte[]` **`data`**`, int` **`nBytes`**`, int` **`offset`**`)`
  - **Description**
    Write an array of bytes into the pngBytes array, specifying number of bytes to write. Note: This routine has the side effect of updating maxPos, the largest element written in the array. The array is resized by 1000 bytes or the length of the data to be written, whichever is larger.
  - **Parameters**
    - `data` – The data to be written into pngBytes.
    - `nBytes` – The number of bytes to be written.
    - `offset` – The starting point to write to.
  - **Returns** – The next place to be written to in the pngBytes array.

- **writeEnd**
  `protected void` **`writeEnd`**`()`
  - **Description**
    Write a PNG ”IEND” chunk into the pngBytes array.

- **writeHeader**
  `protected void` **`writeHeader`**`()`
  - **Description**
    Write a PNG ”IHDR” chunk into the pngBytes array.

- **writeImageData**
  `protected boolean` **`writeImageData`**`()`
  - **Description**
    Write the image data into the pngBytes array. This will write one or more PNG ”IDAT” chunks. In order to conserve memory, this method grabs as many rows as will fit into 32K bytes, or the whole image; whichever is less.
  - **Returns** – true if no errors; false if error grabbing pixels

- **writeInt2**
  `protected int` **`writeInt2`**`(int` **`n`**`, int` **`offset`**`)`
  - **Description**
    Write a two-byte integer into the pngBytes array at a given position.
  - **Parameters**
    - `n` – The integer to be written into pngBytes.

    - `offset` – The starting point to write to.
  - **Returns** – The next place to be written to in the pngBytes array.

- **writeInt4**
  `protected int` **`writeInt4`**`(int` **`n`**`, int` **`offset`**`)`
  - **Description**
    Write a four-byte integer into the pngBytes array at a given position.
  - **Parameters**
    - `n` – The integer to be written into pngBytes.
    - `offset` – The starting point to write to.
  - **Returns** – The next place to be written to in the pngBytes array.

- **writeString**
  `protected int` **`writeString`**`(java.lang.String s, int` **`offset`**`)`
  - **Description**
    Write a string into the pngBytes array at a given position. This uses the getBytes method, so the encoding used will be its default.
  - **Parameters**
    - `n` – The integer to be written into pngBytes.
    - `offset` – The starting point to write to.
  - **Returns** – The next place to be written to in the pngBytes array.
  - **See also**
    - `java.lang.String.getBytes()`

3.2 Class PngEncoderB

3.2.1 Declaration

public class PngEncoderB
extends com.keypoint.PngEncoder (in 3.1, page 14)

3.2.2 Fields

- protected java.awt.image.BufferedImage **image**
- protected java.awt.image.WritableRaster **wRaster**
- protected int **tType**

3.2.3 Constructors

- **PngEncoderB**
  `public` **PngEncoderB**`()`
  - **Description**
    Class constructor
- **PngEncoderB**
  `public` **PngEncoderB**`(java.awt.image.BufferedImage` **image**`)`
  - **Description**
    Class constructor specifying BufferedImage to encode, with no alpha channel encoding.
  - **Parameters**
    - `image` – A Java BufferedImage object
- **PngEncoderB**
  `public` **PngEncoderB**`(java.awt.image.BufferedImage` **image**`, boolean` **encodeAlpha**`)`
  - **Description**
    Class constructor specifying BufferedImage to encode, and whether to encode alpha.
  - **Parameters**
    - `image` – A Java BufferedImage object
    - `encodeAlpha` – Encode the alpha channel? false=no; true=yes
- **PngEncoderB**
  `public` **PngEncoderB**`(java.awt.image.BufferedImage` **image**`, boolean` **encodeAlpha**`, int` **whichFilter**`)`
  - **Description**
    Class constructor specifying BufferedImage to encode, whether to encode alpha, and filter to use.
  - **Parameters**
    - `image` – A Java BufferedImage object
    - `encodeAlpha` – Encode the alpha channel? false=no; true=yes
    - `whichFilter` – 0=none, 1=sub, 2=up
- **PngEncoderB**
  `public` **PngEncoderB**`(java.awt.image.BufferedImage` **image**`, boolean` **encodeAlpha**`, int` **whichFilter**`, int` **compLevel**`)`
  - **Description**
    Class constructor specifying BufferedImage source to encode, whether to encode alpha, filter to use, and compression level
  - **Parameters**

    * `image` – A Java BufferedImage object
    * `encodeAlpha` – Encode the alpha channel? false=no; true=yes
    * `whichFilter` – 0=none, 1=sub, 2=up
    * `compLevel` – 0..9

3.2.4 Methods

- **establishStorageInfo**
  `protected boolean` **establishStorageInfo**`()`
  - **Description**
    Get and set variables that determine how picture is stored. Retrieves the writable raster of the buffered image, as well its transfer type. Sets number of output bytes per pixel, and, if only eight-bit bytes, turns off alpha encoding.
  - **Returns** – true if 1-byte or 4-byte data, false otherwise
- **pngEncode**
  `public byte[]` **pngEncode**`()`
  - **Description**
    Creates an array of bytes that is the PNG equivalent of the current image. Alpha encoding is determined by its setting in the constructor.
  - **Returns** – an array of bytes, or null if there was a problem
- **pngEncode**
  `public byte[]` **pngEncode**`(boolean` **encodeAlpha**`)`
  - **Description**
    Creates an array of bytes that is the PNG equivalent of the current image, specifying whether to encode alpha or not.
  - **Parameters**
    * `encodeAlpha` – boolean false=no alpha, true=encode alpha
  - **Returns** – an array of bytes, or null if there was a problem
- **setImage**
  `public void` **setImage**`(java.awt.image.BufferedImage` **image**`)`
  - **Description**
    Set the BufferedImage to be encoded
  - **Parameters**
    * `BufferedImage` – A Java BufferedImage object
- **writeHeader**
  `protected void` **writeHeader**`()`
  - **Description**
    Write a PNG ”IHDR” chunk into the pngBytes array.

- **writeImageData**
  `protected boolean` **writeImageData**`()`
  - **Description**
    Write the image data into the pngBytes array. This will write one or more PNG "IDAT" chunks. In order to conserve memory, this method grabs as many rows as will fit into 32K bytes, or the whole image; whichever is less.
  - **Returns** – true if no errors; false if error grabbing pixels

- **writePalette**
  `protected void` **writePalette**`(java.awt.image.IndexColorModel` **icm**`)`

3.2.5 Members inherited from class PngEncoder

`com.keypoint.PngEncoder` (in 3.1, page 14)

- `protected` **bytePos**
- `protected` **bytesPerPixel**
- `protected` **compressionLevel**
- `protected` **crc**
- `protected` **crcValue**
- `protected` **dataPos**
- `public static final` **ENCODE_ALPHA**
- `protected` **encodeAlpha**
- `protected` **endPos**
- `protected` **filter**
- `public static final` **FILTER_LAST**
- `public static final` **FILTER_NONE**
- `public static final` **FILTER_SUB**
- `public static final` **FILTER_UP**
- `protected void` **filterSub**`(byte[]` **pixels**`, int` **startPos**`, int` **width**`)`
- `protected void` **filterUp**`(byte[]` **pixels**`, int` **startPos**`, int` **width**`)`
- `public int` **getCompressionLevel**`()`
- `public boolean` **getEncodeAlpha**`()`
- `public int` **getFilter**`()`
- `protected` **hdrPos**
- `protected` **height**
- `protected` **image**
- `protected` **leftBytes**
- `protected` **maxPos**
- `public static final` **NO_ALPHA**
- `protected` **pngBytes**
- `public byte` **pngEncode**`()`
- `public byte` **pngEncode**`(boolean` **encodeAlpha**`)`
- `protected` **priorRow**
- `protected byte` **resizeByteArray**`(byte[]` **array**`, int` **newLength**`)`
- `public void` **setCompressionLevel**`(int` **level**`)`
- `public void` **setEncodeAlpha**`(boolean` **encodeAlpha**`)`
- `public void` **setFilter**`(int` **whichFilter**`)`
- `public void` **setImage**`(java.awt.Image` **image**`)`
- `protected` **width**
- `protected int` **writeByte**`(int` **b**`, int` **offset**`)`
- `protected int` **writeBytes**`(byte[]` **data**`, int` **offset**`)`
- `protected int` **writeBytes**`(byte[]` **data**`, int` **nBytes**`, int` **offset**`)`
- `protected void` **writeEnd**`()`
- `protected void` **writeHeader**`()`
- `protected boolean` **writeImageData**`()`
- `protected int` **writeInt2**`(int` **n**`, int` **offset**`)`
- `protected int` **writeInt4**`(int` **n**`, int` **offset**`)`
- `protected int` **writeString**`(java.lang.String` **s**`, int` **offset**`)`

4 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.