| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/ImageTypeSpecifier.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/imageio/ImageTranscoder.html)   [**NEXT CLASS**](http://docs.google.com/javax/imageio/ImageWriteParam.html) | [**FRAMES**](http://docs.google.com/index.html?javax/imageio/ImageTypeSpecifier.html)    [**NO FRAMES**](http://docs.google.com/ImageTypeSpecifier.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | [FIELD](#3znysh7) | [CONSTR](#2et92p0) | [METHOD](#tyjcwt) | DETAIL: [FIELD](#1t3h5sf) | [CONSTR](#17dp8vu) | [METHOD](#lnxbz9) |

## **javax.imageio**

Class ImageTypeSpecifier

[java.lang.Object](http://docs.google.com/java/lang/Object.html)  
 **javax.imageio.ImageTypeSpecifier**

public class **ImageTypeSpecifier**extends [Object](http://docs.google.com/java/lang/Object.html)

A class that allows the format of an image (in particular, its SampleModel and ColorModel) to be specified in a convenient manner.

| **Field Summary** | |
| --- | --- |
| protected  [ColorModel](http://docs.google.com/java/awt/image/ColorModel.html) | [**colorModel**](http://docs.google.com/javax/imageio/ImageTypeSpecifier.html#colorModel)            The ColorModel to be used as a prototype. |
| protected  [SampleModel](http://docs.google.com/java/awt/image/SampleModel.html) | [**sampleModel**](http://docs.google.com/javax/imageio/ImageTypeSpecifier.html#sampleModel)            A SampleModel to be used as a prototype. |

| **Constructor Summary** | |
| --- | --- |
| [**ImageTypeSpecifier**](http://docs.google.com/javax/imageio/ImageTypeSpecifier.html#ImageTypeSpecifier(java.awt.image.ColorModel,%20java.awt.image.SampleModel))([ColorModel](http://docs.google.com/java/awt/image/ColorModel.html) colorModel, [SampleModel](http://docs.google.com/java/awt/image/SampleModel.html) sampleModel)            Constructs an ImageTypeSpecifier directly from a ColorModel and a SampleModel. |
| [**ImageTypeSpecifier**](http://docs.google.com/javax/imageio/ImageTypeSpecifier.html#ImageTypeSpecifier(java.awt.image.RenderedImage))([RenderedImage](http://docs.google.com/java/awt/image/RenderedImage.html) image)            Constructs an ImageTypeSpecifier from a RenderedImage. |

| **Method Summary** | |
| --- | --- |
| static [ImageTypeSpecifier](http://docs.google.com/javax/imageio/ImageTypeSpecifier.html) | [**createBanded**](http://docs.google.com/javax/imageio/ImageTypeSpecifier.html#createBanded(java.awt.color.ColorSpace,%20int%5B%5D,%20int%5B%5D,%20int,%20boolean,%20boolean))([ColorSpace](http://docs.google.com/java/awt/color/ColorSpace.html) colorSpace, int[] bankIndices, int[] bandOffsets, int dataType, boolean hasAlpha, boolean isAlphaPremultiplied)            Returns a specifier for a banded image format that will use a ComponentColorModel and a BandedSampleModel to store each channel in a separate array. |
| [BufferedImage](http://docs.google.com/java/awt/image/BufferedImage.html) | [**createBufferedImage**](http://docs.google.com/javax/imageio/ImageTypeSpecifier.html#createBufferedImage(int,%20int))(int width, int height)            Creates a BufferedImage with a given width and height according to the specification embodied in this object. |
| static [ImageTypeSpecifier](http://docs.google.com/javax/imageio/ImageTypeSpecifier.html) | [**createFromBufferedImageType**](http://docs.google.com/javax/imageio/ImageTypeSpecifier.html#createFromBufferedImageType(int))(int bufferedImageType)            Returns an ImageTypeSpecifier that encodes one of the standard BufferedImage types (other than TYPE\_CUSTOM). |
| static [ImageTypeSpecifier](http://docs.google.com/javax/imageio/ImageTypeSpecifier.html) | [**createFromRenderedImage**](http://docs.google.com/javax/imageio/ImageTypeSpecifier.html#createFromRenderedImage(java.awt.image.RenderedImage))([RenderedImage](http://docs.google.com/java/awt/image/RenderedImage.html) image)            Returns an ImageTypeSpecifier that encodes the layout of a RenderedImage (which may be a BufferedImage). |
| static [ImageTypeSpecifier](http://docs.google.com/javax/imageio/ImageTypeSpecifier.html) | [**createGrayscale**](http://docs.google.com/javax/imageio/ImageTypeSpecifier.html#createGrayscale(int,%20int,%20boolean))(int bits, int dataType, boolean isSigned)            Returns a specifier for a grayscale image format that will pack pixels of the given bit depth into array elements of the specified data type. |
| static [ImageTypeSpecifier](http://docs.google.com/javax/imageio/ImageTypeSpecifier.html) | [**createGrayscale**](http://docs.google.com/javax/imageio/ImageTypeSpecifier.html#createGrayscale(int,%20int,%20boolean,%20boolean))(int bits, int dataType, boolean isSigned, boolean isAlphaPremultiplied)            Returns a specifier for a grayscale plus alpha image format that will pack pixels of the given bit depth into array elements of the specified data type. |
| static [ImageTypeSpecifier](http://docs.google.com/javax/imageio/ImageTypeSpecifier.html) | [**createIndexed**](http://docs.google.com/javax/imageio/ImageTypeSpecifier.html#createIndexed(byte%5B%5D,%20byte%5B%5D,%20byte%5B%5D,%20byte%5B%5D,%20int,%20int))(byte[] redLUT, byte[] greenLUT, byte[] blueLUT, byte[] alphaLUT, int bits, int dataType)            Returns a specifier for an indexed-color image format that will pack index values of the given bit depth into array elements of the specified data type. |
| static [ImageTypeSpecifier](http://docs.google.com/javax/imageio/ImageTypeSpecifier.html) | [**createInterleaved**](http://docs.google.com/javax/imageio/ImageTypeSpecifier.html#createInterleaved(java.awt.color.ColorSpace,%20int%5B%5D,%20int,%20boolean,%20boolean))([ColorSpace](http://docs.google.com/java/awt/color/ColorSpace.html) colorSpace, int[] bandOffsets, int dataType, boolean hasAlpha, boolean isAlphaPremultiplied)            Returns a specifier for an interleaved image format that will use a ComponentColorModel and a PixelInterleavedSampleModel to store each pixel component in a separate byte, short, or int. |
| static [ImageTypeSpecifier](http://docs.google.com/javax/imageio/ImageTypeSpecifier.html) | [**createPacked**](http://docs.google.com/javax/imageio/ImageTypeSpecifier.html#createPacked(java.awt.color.ColorSpace,%20int,%20int,%20int,%20int,%20int,%20boolean))([ColorSpace](http://docs.google.com/java/awt/color/ColorSpace.html) colorSpace, int redMask, int greenMask, int blueMask, int alphaMask, int transferType, boolean isAlphaPremultiplied)            Returns a specifier for a packed image format that will use a DirectColorModel and a packed SampleModel to store each pixel packed into in a single byte, short, or int. |
| boolean | [**equals**](http://docs.google.com/javax/imageio/ImageTypeSpecifier.html#equals(java.lang.Object))([Object](http://docs.google.com/java/lang/Object.html) o)            Returns true if the given Object is an ImageTypeSpecifier and has a SampleModel and ColorModel that are equal to those of this object. |
| int | [**getBitsPerBand**](http://docs.google.com/javax/imageio/ImageTypeSpecifier.html#getBitsPerBand(int))(int band)            Return the number of bits used to represent samples of the given band. |
| int | [**getBufferedImageType**](http://docs.google.com/javax/imageio/ImageTypeSpecifier.html#getBufferedImageType())()            Returns an int containing one of the enumerated constant values describing image formats from BufferedImage. |
| [ColorModel](http://docs.google.com/java/awt/image/ColorModel.html) | [**getColorModel**](http://docs.google.com/javax/imageio/ImageTypeSpecifier.html#getColorModel())()            Returns the ColorModel specified by this object. |
| int | [**getNumBands**](http://docs.google.com/javax/imageio/ImageTypeSpecifier.html#getNumBands())()            Return the number of bands specified by this object. |
| int | [**getNumComponents**](http://docs.google.com/javax/imageio/ImageTypeSpecifier.html#getNumComponents())()            Return the number of color components specified by this object. |
| [SampleModel](http://docs.google.com/java/awt/image/SampleModel.html) | [**getSampleModel**](http://docs.google.com/javax/imageio/ImageTypeSpecifier.html#getSampleModel())()            Returns a SampleModel based on the settings encapsulated within this object. |
| [SampleModel](http://docs.google.com/java/awt/image/SampleModel.html) | [**getSampleModel**](http://docs.google.com/javax/imageio/ImageTypeSpecifier.html#getSampleModel(int,%20int))(int width, int height)            Returns a SampleModel based on the settings encapsulated within this object. |
| int | [**hashCode**](http://docs.google.com/javax/imageio/ImageTypeSpecifier.html#hashCode())()            Returns the hash code for this ImageTypeSpecifier. |

| **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()), [finalize](http://docs.google.com/java/lang/Object.html#finalize()), [getClass](http://docs.google.com/java/lang/Object.html#getClass()), [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** |
| --- |

### colorModel

protected [ColorModel](http://docs.google.com/java/awt/image/ColorModel.html) **colorModel**

The ColorModel to be used as a prototype.

### sampleModel

protected [SampleModel](http://docs.google.com/java/awt/image/SampleModel.html) **sampleModel**

A SampleModel to be used as a prototype.

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

### ImageTypeSpecifier

public **ImageTypeSpecifier**([ColorModel](http://docs.google.com/java/awt/image/ColorModel.html) colorModel,  
 [SampleModel](http://docs.google.com/java/awt/image/SampleModel.html) sampleModel)

Constructs an ImageTypeSpecifier directly from a ColorModel and a SampleModel. It is the caller's responsibility to supply compatible parameters.

**Parameters:**colorModel - a ColorModel.sampleModel - a SampleModel. **Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if either parameter is null. [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if sampleModel is not compatible with colorModel.

### ImageTypeSpecifier

public **ImageTypeSpecifier**([RenderedImage](http://docs.google.com/java/awt/image/RenderedImage.html) image)

Constructs an ImageTypeSpecifier from a RenderedImage. If a BufferedImage is being used, one of the factory methods createFromRenderedImage or createFromBufferedImageType should be used instead in order to get a more accurate result.

**Parameters:**image - a RenderedImage. **Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if the argument is null.

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

### createPacked

public static [ImageTypeSpecifier](http://docs.google.com/javax/imageio/ImageTypeSpecifier.html) **createPacked**([ColorSpace](http://docs.google.com/java/awt/color/ColorSpace.html) colorSpace,  
 int redMask,  
 int greenMask,  
 int blueMask,  
 int alphaMask,  
 int transferType,  
 boolean isAlphaPremultiplied)

Returns a specifier for a packed image format that will use a DirectColorModel and a packed SampleModel to store each pixel packed into in a single byte, short, or int.

**Parameters:**colorSpace - the desired ColorSpace.redMask - a contiguous mask indicated the position of the red channel.greenMask - a contiguous mask indicated the position of the green channel.blueMask - a contiguous mask indicated the position of the blue channel.alphaMask - a contiguous mask indicated the position of the alpha channel.transferType - the desired SampleModel transfer type.isAlphaPremultiplied - true if the color channels will be premultipled by the alpha channel. **Returns:**an ImageTypeSpecifier with the desired characteristics. **Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if colorSpace is null. [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if colorSpace is not of type TYPE\_RGB. [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if no mask has at least 1 bit set. [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if transferType if not one of DataBuffer.TYPE\_BYTE, DataBuffer.TYPE\_USHORT, or DataBuffer.TYPE\_INT.

### createInterleaved

public static [ImageTypeSpecifier](http://docs.google.com/javax/imageio/ImageTypeSpecifier.html) **createInterleaved**([ColorSpace](http://docs.google.com/java/awt/color/ColorSpace.html) colorSpace,  
 int[] bandOffsets,  
 int dataType,  
 boolean hasAlpha,  
 boolean isAlphaPremultiplied)

Returns a specifier for an interleaved image format that will use a ComponentColorModel and a PixelInterleavedSampleModel to store each pixel component in a separate byte, short, or int.

**Parameters:**colorSpace - the desired ColorSpace.bandOffsets - an array of ints indicating the offsets for each band.dataType - the desired data type, as one of the enumerations from the DataBuffer class.hasAlpha - true if an alpha channel is desired.isAlphaPremultiplied - true if the color channels will be premultipled by the alpha channel. **Returns:**an ImageTypeSpecifier with the desired characteristics. **Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if colorSpace is null. [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if bandOffsets is null. [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if dataType is not one of the legal DataBuffer.TYPE\_\* constants. [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if bandOffsets.length does not equal the number of color space components, plus 1 if hasAlpha is true.

### createBanded

public static [ImageTypeSpecifier](http://docs.google.com/javax/imageio/ImageTypeSpecifier.html) **createBanded**([ColorSpace](http://docs.google.com/java/awt/color/ColorSpace.html) colorSpace,  
 int[] bankIndices,  
 int[] bandOffsets,  
 int dataType,  
 boolean hasAlpha,  
 boolean isAlphaPremultiplied)

Returns a specifier for a banded image format that will use a ComponentColorModel and a BandedSampleModel to store each channel in a separate array.

**Parameters:**colorSpace - the desired ColorSpace.bankIndices - an array of ints indicating the bank in which each band will be stored.bandOffsets - an array of ints indicating the starting offset of each band within its bank.dataType - the desired data type, as one of the enumerations from the DataBuffer class.hasAlpha - true if an alpha channel is desired.isAlphaPremultiplied - true if the color channels will be premultipled by the alpha channel. **Returns:**an ImageTypeSpecifier with the desired characteristics. **Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if colorSpace is null. [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if bankIndices is null. [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if bandOffsets is null. [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if the lengths of bankIndices and bandOffsets differ. [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if bandOffsets.length does not equal the number of color space components, plus 1 if hasAlpha is true. [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if dataType is not one of the legal DataBuffer.TYPE\_\* constants.

### createGrayscale

public static [ImageTypeSpecifier](http://docs.google.com/javax/imageio/ImageTypeSpecifier.html) **createGrayscale**(int bits,  
 int dataType,  
 boolean isSigned)

Returns a specifier for a grayscale image format that will pack pixels of the given bit depth into array elements of the specified data type.

**Parameters:**bits - the number of bits per gray value (1, 2, 4, 8, or 16).dataType - the desired data type, as one of the enumerations from the DataBuffer class.isSigned - true if negative values are to be represented. **Returns:**an ImageTypeSpecifier with the desired characteristics. **Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if bits is not one of 1, 2, 4, 8, or 16. [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if dataType is not one of DataBuffer.TYPE\_BYTE, DataBuffer.TYPE\_SHORT, or DataBuffer.TYPE\_USHORT. [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if bits is larger than the bit size of the given dataType.

### createGrayscale

public static [ImageTypeSpecifier](http://docs.google.com/javax/imageio/ImageTypeSpecifier.html) **createGrayscale**(int bits,  
 int dataType,  
 boolean isSigned,  
 boolean isAlphaPremultiplied)

Returns a specifier for a grayscale plus alpha image format that will pack pixels of the given bit depth into array elements of the specified data type.

**Parameters:**bits - the number of bits per gray value (1, 2, 4, 8, or 16).dataType - the desired data type, as one of the enumerations from the DataBuffer class.isSigned - true if negative values are to be represented.isAlphaPremultiplied - true if the luminance channel will be premultipled by the alpha channel. **Returns:**an ImageTypeSpecifier with the desired characteristics. **Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if bits is not one of 1, 2, 4, 8, or 16. [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if dataType is not one of DataBuffer.TYPE\_BYTE, DataBuffer.TYPE\_SHORT, or DataBuffer.TYPE\_USHORT. [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if bits is larger than the bit size of the given dataType.

### createIndexed

public static [ImageTypeSpecifier](http://docs.google.com/javax/imageio/ImageTypeSpecifier.html) **createIndexed**(byte[] redLUT,  
 byte[] greenLUT,  
 byte[] blueLUT,  
 byte[] alphaLUT,  
 int bits,  
 int dataType)

Returns a specifier for an indexed-color image format that will pack index values of the given bit depth into array elements of the specified data type.

**Parameters:**redLUT - an array of bytes containing the red values for each index.greenLUT - an array of bytes containing \* the green values for each index.blueLUT - an array of bytes containing the blue values for each index.alphaLUT - an array of bytes containing the alpha values for each index, or null to create a fully opaque LUT.bits - the number of bits in each index.dataType - the desired output type, as one of the enumerations from the DataBuffer class. **Returns:**an ImageTypeSpecifier with the desired characteristics. **Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if redLUT is null. [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if greenLUT is null. [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if blueLUT is null. [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if bits is not one of 1, 2, 4, 8, or 16. [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if the non-null LUT parameters do not have lengths of exactly 1 << bits. [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if dataType is not one of DataBuffer.TYPE\_BYTE, DataBuffer.TYPE\_SHORT, DataBuffer.TYPE\_USHORT, or DataBuffer.TYPE\_INT. [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if bits is larger than the bit size of the given dataType.

### createFromBufferedImageType

public static [ImageTypeSpecifier](http://docs.google.com/javax/imageio/ImageTypeSpecifier.html) **createFromBufferedImageType**(int bufferedImageType)

Returns an ImageTypeSpecifier that encodes one of the standard BufferedImage types (other than TYPE\_CUSTOM).

**Parameters:**bufferedImageType - an int representing one of the standard BufferedImage types. **Returns:**an ImageTypeSpecifier with the desired characteristics. **Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if bufferedImageType is not one of the standard types, or is equal to TYPE\_CUSTOM.**See Also:**[BufferedImage](http://docs.google.com/java/awt/image/BufferedImage.html), [BufferedImage.TYPE\_INT\_RGB](http://docs.google.com/java/awt/image/BufferedImage.html#TYPE_INT_RGB), [BufferedImage.TYPE\_INT\_ARGB](http://docs.google.com/java/awt/image/BufferedImage.html#TYPE_INT_ARGB), [BufferedImage.TYPE\_INT\_ARGB\_PRE](http://docs.google.com/java/awt/image/BufferedImage.html#TYPE_INT_ARGB_PRE), [BufferedImage.TYPE\_INT\_BGR](http://docs.google.com/java/awt/image/BufferedImage.html#TYPE_INT_BGR), [BufferedImage.TYPE\_3BYTE\_BGR](http://docs.google.com/java/awt/image/BufferedImage.html#TYPE_3BYTE_BGR), [BufferedImage.TYPE\_4BYTE\_ABGR](http://docs.google.com/java/awt/image/BufferedImage.html#TYPE_4BYTE_ABGR), [BufferedImage.TYPE\_4BYTE\_ABGR\_PRE](http://docs.google.com/java/awt/image/BufferedImage.html#TYPE_4BYTE_ABGR_PRE), [BufferedImage.TYPE\_USHORT\_565\_RGB](http://docs.google.com/java/awt/image/BufferedImage.html#TYPE_USHORT_565_RGB), [BufferedImage.TYPE\_USHORT\_555\_RGB](http://docs.google.com/java/awt/image/BufferedImage.html#TYPE_USHORT_555_RGB), [BufferedImage.TYPE\_BYTE\_GRAY](http://docs.google.com/java/awt/image/BufferedImage.html#TYPE_BYTE_GRAY), [BufferedImage.TYPE\_USHORT\_GRAY](http://docs.google.com/java/awt/image/BufferedImage.html#TYPE_USHORT_GRAY), [BufferedImage.TYPE\_BYTE\_BINARY](http://docs.google.com/java/awt/image/BufferedImage.html#TYPE_BYTE_BINARY), [BufferedImage.TYPE\_BYTE\_INDEXED](http://docs.google.com/java/awt/image/BufferedImage.html#TYPE_BYTE_INDEXED)

### createFromRenderedImage

public static [ImageTypeSpecifier](http://docs.google.com/javax/imageio/ImageTypeSpecifier.html) **createFromRenderedImage**([RenderedImage](http://docs.google.com/java/awt/image/RenderedImage.html) image)

Returns an ImageTypeSpecifier that encodes the layout of a RenderedImage (which may be a BufferedImage).

**Parameters:**image - a RenderedImage. **Returns:**an ImageTypeSpecifier with the desired characteristics. **Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if image is null.

### getBufferedImageType

public int **getBufferedImageType**()

Returns an int containing one of the enumerated constant values describing image formats from BufferedImage.

**Returns:**an int representing a BufferedImage type.**See Also:**[BufferedImage](http://docs.google.com/java/awt/image/BufferedImage.html), [BufferedImage.TYPE\_CUSTOM](http://docs.google.com/java/awt/image/BufferedImage.html#TYPE_CUSTOM), [BufferedImage.TYPE\_INT\_RGB](http://docs.google.com/java/awt/image/BufferedImage.html#TYPE_INT_RGB), [BufferedImage.TYPE\_INT\_ARGB](http://docs.google.com/java/awt/image/BufferedImage.html#TYPE_INT_ARGB), [BufferedImage.TYPE\_INT\_ARGB\_PRE](http://docs.google.com/java/awt/image/BufferedImage.html#TYPE_INT_ARGB_PRE), [BufferedImage.TYPE\_INT\_BGR](http://docs.google.com/java/awt/image/BufferedImage.html#TYPE_INT_BGR), [BufferedImage.TYPE\_3BYTE\_BGR](http://docs.google.com/java/awt/image/BufferedImage.html#TYPE_3BYTE_BGR), [BufferedImage.TYPE\_4BYTE\_ABGR](http://docs.google.com/java/awt/image/BufferedImage.html#TYPE_4BYTE_ABGR), [BufferedImage.TYPE\_4BYTE\_ABGR\_PRE](http://docs.google.com/java/awt/image/BufferedImage.html#TYPE_4BYTE_ABGR_PRE), [BufferedImage.TYPE\_USHORT\_565\_RGB](http://docs.google.com/java/awt/image/BufferedImage.html#TYPE_USHORT_565_RGB), [BufferedImage.TYPE\_USHORT\_555\_RGB](http://docs.google.com/java/awt/image/BufferedImage.html#TYPE_USHORT_555_RGB), [BufferedImage.TYPE\_BYTE\_GRAY](http://docs.google.com/java/awt/image/BufferedImage.html#TYPE_BYTE_GRAY), [BufferedImage.TYPE\_USHORT\_GRAY](http://docs.google.com/java/awt/image/BufferedImage.html#TYPE_USHORT_GRAY), [BufferedImage.TYPE\_BYTE\_BINARY](http://docs.google.com/java/awt/image/BufferedImage.html#TYPE_BYTE_BINARY), [BufferedImage.TYPE\_BYTE\_INDEXED](http://docs.google.com/java/awt/image/BufferedImage.html#TYPE_BYTE_INDEXED)

### getNumComponents

public int **getNumComponents**()

Return the number of color components specified by this object. This is the same value as returned by ColorModel.getNumComponents

**Returns:**the number of components in the image.

### getNumBands

public int **getNumBands**()

Return the number of bands specified by this object. This is the same value as returned by SampleModel.getNumBands

**Returns:**the number of bands in the image.

### getBitsPerBand

public int **getBitsPerBand**(int band)

Return the number of bits used to represent samples of the given band.

**Parameters:**band - the index of the band to be queried, as an int. **Returns:**an int specifying a number of bits. **Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if band is negative or greater than the largest band index.

### getSampleModel

public [SampleModel](http://docs.google.com/java/awt/image/SampleModel.html) **getSampleModel**()

Returns a SampleModel based on the settings encapsulated within this object. The width and height of the SampleModel will be set to arbitrary values.

**Returns:**a SampleModel with arbitrary dimensions.

### getSampleModel

public [SampleModel](http://docs.google.com/java/awt/image/SampleModel.html) **getSampleModel**(int width,  
 int height)

Returns a SampleModel based on the settings encapsulated within this object. The width and height of the SampleModel will be set to the supplied values.

**Parameters:**width - the desired width of the returned SampleModel.height - the desired height of the returned SampleModel. **Returns:**a SampleModel with the given dimensions. **Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if either width or height are negative or zero. [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if the product of width and height is greater than Integer.MAX\_VALUE

### getColorModel

public [ColorModel](http://docs.google.com/java/awt/image/ColorModel.html) **getColorModel**()

Returns the ColorModel specified by this object.

**Returns:**a ColorModel.

### createBufferedImage

public [BufferedImage](http://docs.google.com/java/awt/image/BufferedImage.html) **createBufferedImage**(int width,  
 int height)

Creates a BufferedImage with a given width and height according to the specification embodied in this object.

**Parameters:**width - the desired width of the returned BufferedImage.height - the desired height of the returned BufferedImage. **Returns:**a new BufferedImage **Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if either width or height are negative or zero. [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if the product of width and height is greater than Integer.MAX\_VALUE, or if the number of array elements needed to store the image is greater than Integer.MAX\_VALUE.

### equals

public boolean **equals**([Object](http://docs.google.com/java/lang/Object.html) o)

Returns true if the given Object is an ImageTypeSpecifier and has a SampleModel and ColorModel that are equal to those of this object.

**Overrides:**[equals](http://docs.google.com/java/lang/Object.html#equals(java.lang.Object)) in class [Object](http://docs.google.com/java/lang/Object.html) **Parameters:**o - the Object to be compared for equality. **Returns:**true if the given object is an equivalent ImageTypeSpecifier.**See Also:**[Object.hashCode()](http://docs.google.com/java/lang/Object.html#hashCode()), [Hashtable](http://docs.google.com/java/util/Hashtable.html)

### hashCode

public int **hashCode**()

Returns the hash code for this ImageTypeSpecifier.

**Overrides:**[hashCode](http://docs.google.com/java/lang/Object.html#hashCode()) in class [Object](http://docs.google.com/java/lang/Object.html) **Returns:**a hash code for this ImageTypeSpecifier**See Also:**[Object.equals(java.lang.Object)](http://docs.google.com/java/lang/Object.html#equals(java.lang.Object)), [Hashtable](http://docs.google.com/java/util/Hashtable.html)

| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/ImageTypeSpecifier.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/imageio/ImageTranscoder.html)   [**NEXT CLASS**](http://docs.google.com/javax/imageio/ImageWriteParam.html) | [**FRAMES**](http://docs.google.com/index.html?javax/imageio/ImageTypeSpecifier.html)    [**NO FRAMES**](http://docs.google.com/ImageTypeSpecifier.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | [FIELD](#3znysh7) | [CONSTR](#2et92p0) | [METHOD](#tyjcwt) | DETAIL: [FIELD](#1t3h5sf) | [CONSTR](#17dp8vu) | [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).