| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/PackedColorModel.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/java/awt/image/MultiPixelPackedSampleModel.html)   [**NEXT CLASS**](http://docs.google.com/java/awt/image/PixelGrabber.html) | [**FRAMES**](http://docs.google.com/index.html?java/awt/image/PackedColorModel.html)    [**NO FRAMES**](http://docs.google.com/PackedColorModel.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | [FIELD](#2et92p0) | [CONSTR](#3dy6vkm) | [METHOD](#1t3h5sf) | DETAIL: FIELD | [CONSTR](#17dp8vu) | [METHOD](#lnxbz9) |

## **java.awt.image**

Class PackedColorModel

[java.lang.Object](http://docs.google.com/java/lang/Object.html)  
 [java.awt.image.ColorModel](http://docs.google.com/java/awt/image/ColorModel.html)  
 **java.awt.image.PackedColorModel**

**All Implemented Interfaces:** [Transparency](http://docs.google.com/java/awt/Transparency.html) **Direct Known Subclasses:** [DirectColorModel](http://docs.google.com/java/awt/image/DirectColorModel.html)

public abstract class **PackedColorModel**extends [ColorModel](http://docs.google.com/java/awt/image/ColorModel.html)

The PackedColorModel class is an abstract [ColorModel](http://docs.google.com/java/awt/image/ColorModel.html) class that works with pixel values which represent color and alpha information as separate samples and which pack all samples for a single pixel into a single int, short, or byte quantity. This class can be used with an arbitrary [ColorSpace](http://docs.google.com/java/awt/color/ColorSpace.html). The number of color samples in the pixel values must be the same as the number of color components in the ColorSpace. There can be a single alpha sample. The array length is always 1 for those methods that use a primitive array pixel representation of type transferType. The transfer types supported are DataBuffer.TYPE\_BYTE, DataBuffer.TYPE\_USHORT, and DataBuffer.TYPE\_INT. Color and alpha samples are stored in the single element of the array in bits indicated by bit masks. Each bit mask must be contiguous and masks must not overlap. The same masks apply to the single int pixel representation used by other methods. The correspondence of masks and color/alpha samples is as follows:

* Masks are identified by indices running from 0 through [getNumComponents](http://docs.google.com/java/awt/image/ColorModel.html#getNumComponents()) - 1.
* The first [getNumColorComponents](http://docs.google.com/java/awt/image/ColorModel.html#getNumColorComponents()) indices refer to color samples.
* If an alpha sample is present, it corresponds the last index.
* The order of the color indices is specified by the ColorSpace. Typically, this reflects the name of the color space type (for example, TYPE\_RGB), index 0 corresponds to red, index 1 to green, and index 2 to blue.

The translation from pixel values to color/alpha components for display or processing purposes is a one-to-one correspondence of samples to components. A PackedColorModel is typically used with image data that uses masks to define packed samples. For example, a PackedColorModel can be used in conjunction with a [SinglePixelPackedSampleModel](http://docs.google.com/java/awt/image/SinglePixelPackedSampleModel.html) to construct a [BufferedImage](http://docs.google.com/java/awt/image/BufferedImage.html). Normally the masks used by the [SampleModel](http://docs.google.com/java/awt/image/SampleModel.html) and the ColorModel would be the same. However, if they are different, the color interpretation of pixel data is done according to the masks of the ColorModel.

A single int pixel representation is valid for all objects of this class since it is always possible to represent pixel values used with this class in a single int. Therefore, methods that use this representation do not throw an IllegalArgumentException due to an invalid pixel value.

A subclass of PackedColorModel is [DirectColorModel](http://docs.google.com/java/awt/image/DirectColorModel.html), which is similar to an X11 TrueColor visual.

**See Also:**[DirectColorModel](http://docs.google.com/java/awt/image/DirectColorModel.html), [SinglePixelPackedSampleModel](http://docs.google.com/java/awt/image/SinglePixelPackedSampleModel.html), [BufferedImage](http://docs.google.com/java/awt/image/BufferedImage.html)

| **Field Summary** | |
| --- | --- |

| **Fields inherited from class java.awt.image.**[**ColorModel**](http://docs.google.com/java/awt/image/ColorModel.html) |
| --- |
| [pixel\_bits](http://docs.google.com/java/awt/image/ColorModel.html#pixel_bits), [transferType](http://docs.google.com/java/awt/image/ColorModel.html#transferType) |

| **Fields inherited from interface java.awt.**[**Transparency**](http://docs.google.com/java/awt/Transparency.html) |
| --- |
| [BITMASK](http://docs.google.com/java/awt/Transparency.html#BITMASK), [OPAQUE](http://docs.google.com/java/awt/Transparency.html#OPAQUE), [TRANSLUCENT](http://docs.google.com/java/awt/Transparency.html#TRANSLUCENT) |

| **Constructor Summary** | |
| --- | --- |
| [**PackedColorModel**](http://docs.google.com/java/awt/image/PackedColorModel.html#PackedColorModel(java.awt.color.ColorSpace,%20int,%20int%5B%5D,%20int,%20boolean,%20int,%20int))([ColorSpace](http://docs.google.com/java/awt/color/ColorSpace.html) space, int bits, int[] colorMaskArray, int alphaMask, boolean isAlphaPremultiplied, int trans, int transferType)            Constructs a PackedColorModel from a color mask array, which specifies which bits in an int pixel representation contain each of the color samples, and an alpha mask. |
| [**PackedColorModel**](http://docs.google.com/java/awt/image/PackedColorModel.html#PackedColorModel(java.awt.color.ColorSpace,%20int,%20int,%20int,%20int,%20int,%20boolean,%20int,%20int))([ColorSpace](http://docs.google.com/java/awt/color/ColorSpace.html) space, int bits, int rmask, int gmask, int bmask, int amask, boolean isAlphaPremultiplied, int trans, int transferType)            Constructs a PackedColorModel from the specified masks which indicate which bits in an int pixel representation contain the alpha, red, green and blue color samples. |

| **Method Summary** | |
| --- | --- |
| [SampleModel](http://docs.google.com/java/awt/image/SampleModel.html) | [**createCompatibleSampleModel**](http://docs.google.com/java/awt/image/PackedColorModel.html#createCompatibleSampleModel(int,%20int))(int w, int h)            Creates a SampleModel with the specified width and height that has a data layout compatible with this ColorModel. |
| boolean | [**equals**](http://docs.google.com/java/awt/image/PackedColorModel.html#equals(java.lang.Object))([Object](http://docs.google.com/java/lang/Object.html) obj)            Tests if the specified Object is an instance of PackedColorModel and equals this PackedColorModel. |
| [WritableRaster](http://docs.google.com/java/awt/image/WritableRaster.html) | [**getAlphaRaster**](http://docs.google.com/java/awt/image/PackedColorModel.html#getAlphaRaster(java.awt.image.WritableRaster))([WritableRaster](http://docs.google.com/java/awt/image/WritableRaster.html) raster)            Returns a [WritableRaster](http://docs.google.com/java/awt/image/WritableRaster.html) representing the alpha channel of an image, extracted from the input WritableRaster. |
| int | [**getMask**](http://docs.google.com/java/awt/image/PackedColorModel.html#getMask(int))(int index)            Returns the mask indicating which bits in a pixel contain the specified color/alpha sample. |
| int[] | [**getMasks**](http://docs.google.com/java/awt/image/PackedColorModel.html#getMasks())()            Returns a mask array indicating which bits in a pixel contain the color and alpha samples. |
| boolean | [**isCompatibleSampleModel**](http://docs.google.com/java/awt/image/PackedColorModel.html#isCompatibleSampleModel(java.awt.image.SampleModel))([SampleModel](http://docs.google.com/java/awt/image/SampleModel.html) sm)            Checks if the specified SampleModel is compatible with this ColorModel. |

| **Methods inherited from class java.awt.image.**[**ColorModel**](http://docs.google.com/java/awt/image/ColorModel.html) |
| --- |
| [coerceData](http://docs.google.com/java/awt/image/ColorModel.html#coerceData(java.awt.image.WritableRaster,%20boolean)), [createCompatibleWritableRaster](http://docs.google.com/java/awt/image/ColorModel.html#createCompatibleWritableRaster(int,%20int)), [finalize](http://docs.google.com/java/awt/image/ColorModel.html#finalize()), [getAlpha](http://docs.google.com/java/awt/image/ColorModel.html#getAlpha(int)), [getAlpha](http://docs.google.com/java/awt/image/ColorModel.html#getAlpha(java.lang.Object)), [getBlue](http://docs.google.com/java/awt/image/ColorModel.html#getBlue(int)), [getBlue](http://docs.google.com/java/awt/image/ColorModel.html#getBlue(java.lang.Object)), [getColorSpace](http://docs.google.com/java/awt/image/ColorModel.html#getColorSpace()), [getComponents](http://docs.google.com/java/awt/image/ColorModel.html#getComponents(int,%20int%5B%5D,%20int)), [getComponents](http://docs.google.com/java/awt/image/ColorModel.html#getComponents(java.lang.Object,%20int%5B%5D,%20int)), [getComponentSize](http://docs.google.com/java/awt/image/ColorModel.html#getComponentSize()), [getComponentSize](http://docs.google.com/java/awt/image/ColorModel.html#getComponentSize(int)), [getDataElement](http://docs.google.com/java/awt/image/ColorModel.html#getDataElement(float%5B%5D,%20int)), [getDataElement](http://docs.google.com/java/awt/image/ColorModel.html#getDataElement(int%5B%5D,%20int)), [getDataElements](http://docs.google.com/java/awt/image/ColorModel.html#getDataElements(float%5B%5D,%20int,%20java.lang.Object)), [getDataElements](http://docs.google.com/java/awt/image/ColorModel.html#getDataElements(int%5B%5D,%20int,%20java.lang.Object)), [getDataElements](http://docs.google.com/java/awt/image/ColorModel.html#getDataElements(int,%20java.lang.Object)), [getGreen](http://docs.google.com/java/awt/image/ColorModel.html#getGreen(int)), [getGreen](http://docs.google.com/java/awt/image/ColorModel.html#getGreen(java.lang.Object)), [getNormalizedComponents](http://docs.google.com/java/awt/image/ColorModel.html#getNormalizedComponents(int%5B%5D,%20int,%20float%5B%5D,%20int)), [getNormalizedComponents](http://docs.google.com/java/awt/image/ColorModel.html#getNormalizedComponents(java.lang.Object,%20float%5B%5D,%20int)), [getNumColorComponents](http://docs.google.com/java/awt/image/ColorModel.html#getNumColorComponents()), [getNumComponents](http://docs.google.com/java/awt/image/ColorModel.html#getNumComponents()), [getPixelSize](http://docs.google.com/java/awt/image/ColorModel.html#getPixelSize()), [getRed](http://docs.google.com/java/awt/image/ColorModel.html#getRed(int)), [getRed](http://docs.google.com/java/awt/image/ColorModel.html#getRed(java.lang.Object)), [getRGB](http://docs.google.com/java/awt/image/ColorModel.html#getRGB(int)), [getRGB](http://docs.google.com/java/awt/image/ColorModel.html#getRGB(java.lang.Object)), [getRGBdefault](http://docs.google.com/java/awt/image/ColorModel.html#getRGBdefault()), [getTransferType](http://docs.google.com/java/awt/image/ColorModel.html#getTransferType()), [getTransparency](http://docs.google.com/java/awt/image/ColorModel.html#getTransparency()), [getUnnormalizedComponents](http://docs.google.com/java/awt/image/ColorModel.html#getUnnormalizedComponents(float%5B%5D,%20int,%20int%5B%5D,%20int)), [hasAlpha](http://docs.google.com/java/awt/image/ColorModel.html#hasAlpha()), [hashCode](http://docs.google.com/java/awt/image/ColorModel.html#hashCode()), [isAlphaPremultiplied](http://docs.google.com/java/awt/image/ColorModel.html#isAlphaPremultiplied()), [isCompatibleRaster](http://docs.google.com/java/awt/image/ColorModel.html#isCompatibleRaster(java.awt.image.Raster)), [toString](http://docs.google.com/java/awt/image/ColorModel.html#toString()) |

| **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()), [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()), [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)) |

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

### PackedColorModel

public **PackedColorModel**([ColorSpace](http://docs.google.com/java/awt/color/ColorSpace.html) space,  
 int bits,  
 int[] colorMaskArray,  
 int alphaMask,  
 boolean isAlphaPremultiplied,  
 int trans,  
 int transferType)

Constructs a PackedColorModel from a color mask array, which specifies which bits in an int pixel representation contain each of the color samples, and an alpha mask. Color components are in the specified ColorSpace. The length of colorMaskArray should be the number of components in the ColorSpace. All of the bits in each mask must be contiguous and fit in the specified number of least significant bits of an int pixel representation. If the alphaMask is 0, there is no alpha. If there is alpha, the boolean isAlphaPremultiplied specifies how to interpret color and alpha samples in pixel values. If the boolean is true, color samples are assumed to have been multiplied by the alpha sample. The transparency, trans, specifies what alpha values can be represented by this color model. The transfer type is the type of primitive array used to represent pixel values.

**Parameters:**space - the specified ColorSpacebits - the number of bits in the pixel valuescolorMaskArray - array that specifies the masks representing the bits of the pixel values that represent the color componentsalphaMask - specifies the mask representing the bits of the pixel values that represent the alpha componentisAlphaPremultiplied - true if color samples are premultiplied by the alpha sample; false otherwisetrans - specifies the alpha value that can be represented by this color modeltransferType - the type of array used to represent pixel values **Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if bits is less than 1 or greater than 32

### PackedColorModel

public **PackedColorModel**([ColorSpace](http://docs.google.com/java/awt/color/ColorSpace.html) space,  
 int bits,  
 int rmask,  
 int gmask,  
 int bmask,  
 int amask,  
 boolean isAlphaPremultiplied,  
 int trans,  
 int transferType)

Constructs a PackedColorModel from the specified masks which indicate which bits in an int pixel representation contain the alpha, red, green and blue color samples. Color components are in the specified ColorSpace, which must be of type ColorSpace.TYPE\_RGB. All of the bits in each mask must be contiguous and fit in the specified number of least significant bits of an int pixel representation. If amask is 0, there is no alpha. If there is alpha, the boolean isAlphaPremultiplied specifies how to interpret color and alpha samples in pixel values. If the boolean is true, color samples are assumed to have been multiplied by the alpha sample. The transparency, trans, specifies what alpha values can be represented by this color model. The transfer type is the type of primitive array used to represent pixel values.

**Parameters:**space - the specified ColorSpacebits - the number of bits in the pixel valuesrmask - specifies the mask representing the bits of the pixel values that represent the red color componentgmask - specifies the mask representing the bits of the pixel values that represent the green color componentbmask - specifies the mask representing the bits of the pixel values that represent the blue color componentamask - specifies the mask representing the bits of the pixel values that represent the alpha componentisAlphaPremultiplied - true if color samples are premultiplied by the alpha sample; false otherwisetrans - specifies the alpha value that can be represented by this color modeltransferType - the type of array used to represent pixel values **Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if space is not a TYPE\_RGB space**See Also:**[ColorSpace](http://docs.google.com/java/awt/color/ColorSpace.html)

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

### getMask

public final int **getMask**(int index)

Returns the mask indicating which bits in a pixel contain the specified color/alpha sample. For color samples, index corresponds to the placement of color sample names in the color space. Thus, an index equal to 0 for a CMYK ColorSpace would correspond to Cyan and an index equal to 1 would correspond to Magenta. If there is alpha, the alpha index would be:

alphaIndex = numComponents() - 1;

**Parameters:**index - the specified color or alpha sample **Returns:**the mask, which indicates which bits of the int pixel representation contain the color or alpha sample specified by index. **Throws:** [ArrayIndexOutOfBoundsException](http://docs.google.com/java/lang/ArrayIndexOutOfBoundsException.html) - if index is greater than the number of components minus 1 in this PackedColorModel or if index is less than zero

### getMasks

public final int[] **getMasks**()

Returns a mask array indicating which bits in a pixel contain the color and alpha samples.

**Returns:**the mask array , which indicates which bits of the int pixel representation contain the color or alpha samples.

### createCompatibleSampleModel

public [SampleModel](http://docs.google.com/java/awt/image/SampleModel.html) **createCompatibleSampleModel**(int w,  
 int h)

Creates a SampleModel with the specified width and height that has a data layout compatible with this ColorModel.

**Overrides:**[createCompatibleSampleModel](http://docs.google.com/java/awt/image/ColorModel.html#createCompatibleSampleModel(int,%20int)) in class [ColorModel](http://docs.google.com/java/awt/image/ColorModel.html) **Parameters:**w - the width (in pixels) of the region of the image data describedh - the height (in pixels) of the region of the image data described **Returns:**the newly created SampleModel. **Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if w or h is not greater than 0**See Also:**[SampleModel](http://docs.google.com/java/awt/image/SampleModel.html)

### isCompatibleSampleModel

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

Checks if the specified SampleModel is compatible with this ColorModel. If sm is null, this method returns false.

**Overrides:**[isCompatibleSampleModel](http://docs.google.com/java/awt/image/ColorModel.html#isCompatibleSampleModel(java.awt.image.SampleModel)) in class [ColorModel](http://docs.google.com/java/awt/image/ColorModel.html) **Parameters:**sm - the specified SampleModel, or null **Returns:**true if the specified SampleModel is compatible with this ColorModel; false otherwise.**See Also:**[SampleModel](http://docs.google.com/java/awt/image/SampleModel.html)

### getAlphaRaster

public [WritableRaster](http://docs.google.com/java/awt/image/WritableRaster.html) **getAlphaRaster**([WritableRaster](http://docs.google.com/java/awt/image/WritableRaster.html) raster)

Returns a [WritableRaster](http://docs.google.com/java/awt/image/WritableRaster.html) representing the alpha channel of an image, extracted from the input WritableRaster. This method assumes that WritableRaster objects associated with this ColorModel store the alpha band, if present, as the last band of image data. Returns null if there is no separate spatial alpha channel associated with this ColorModel. This method creates a new WritableRaster, but shares the data array.

**Overrides:**[getAlphaRaster](http://docs.google.com/java/awt/image/ColorModel.html#getAlphaRaster(java.awt.image.WritableRaster)) in class [ColorModel](http://docs.google.com/java/awt/image/ColorModel.html) **Parameters:**raster - a WritableRaster containing an image **Returns:**a WritableRaster that represents the alpha channel of the image contained in raster.

### equals

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

Tests if the specified Object is an instance of PackedColorModel and equals this PackedColorModel.

**Overrides:**[equals](http://docs.google.com/java/awt/image/ColorModel.html#equals(java.lang.Object)) in class [ColorModel](http://docs.google.com/java/awt/image/ColorModel.html) **Parameters:**obj - the Object to test for equality **Returns:**true if the specified Object is an instance of PackedColorModel and equals this PackedColorModel; false otherwise.**See Also:**[Object.hashCode()](http://docs.google.com/java/lang/Object.html#hashCode()), [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/PackedColorModel.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/java/awt/image/MultiPixelPackedSampleModel.html)   [**NEXT CLASS**](http://docs.google.com/java/awt/image/PixelGrabber.html) | [**FRAMES**](http://docs.google.com/index.html?java/awt/image/PackedColorModel.html)    [**NO FRAMES**](http://docs.google.com/PackedColorModel.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | [FIELD](#2et92p0) | [CONSTR](#3dy6vkm) | [METHOD](#1t3h5sf) | DETAIL: FIELD | [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).