| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/SinglePixelPackedSampleModel.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/ShortLookupTable.html)   [**NEXT CLASS**](http://docs.google.com/java/awt/image/TileObserver.html) | [**FRAMES**](http://docs.google.com/index.html?java/awt/image/SinglePixelPackedSampleModel.html)    [**NO FRAMES**](http://docs.google.com/SinglePixelPackedSampleModel.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | [FIELD](#2et92p0) | [CONSTR](#tyjcwt) | [METHOD](#3dy6vkm) | DETAIL: FIELD | [CONSTR](#2s8eyo1) | [METHOD](#26in1rg) |

## **java.awt.image**

Class SinglePixelPackedSampleModel

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

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

This class represents pixel data packed such that the N samples which make up a single pixel are stored in a single data array element, and each data data array element holds samples for only one pixel. This class supports [TYPE\_BYTE](http://docs.google.com/java/awt/image/DataBuffer.html#TYPE_BYTE), [TYPE\_USHORT](http://docs.google.com/java/awt/image/DataBuffer.html#TYPE_USHORT), [TYPE\_INT](http://docs.google.com/java/awt/image/DataBuffer.html#TYPE_INT) data types. All data array elements reside in the first bank of a DataBuffer. Accessor methods are provided so that the image data can be manipulated directly. Scanline stride is the number of data array elements between a given sample and the corresponding sample in the same column of the next scanline. Bit masks are the masks required to extract the samples representing the bands of the pixel. Bit offsets are the offsets in bits into the data array element of the samples representing the bands of the pixel.

The following code illustrates extracting the bits of the sample representing band b for pixel x,y from DataBuffer data:

int sample = data.getElem(y \* scanlineStride + x);  
 sample = (sample & bitMasks[b]) >>> bitOffsets[b];

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

| **Fields inherited from class java.awt.image.**[**SampleModel**](http://docs.google.com/java/awt/image/SampleModel.html) |
| --- |
| [dataType](http://docs.google.com/java/awt/image/SampleModel.html#dataType), [height](http://docs.google.com/java/awt/image/SampleModel.html#height), [numBands](http://docs.google.com/java/awt/image/SampleModel.html#numBands), [width](http://docs.google.com/java/awt/image/SampleModel.html#width) |

| **Constructor Summary** | |
| --- | --- |
| [**SinglePixelPackedSampleModel**](http://docs.google.com/java/awt/image/SinglePixelPackedSampleModel.html#SinglePixelPackedSampleModel(int,%20int,%20int,%20int%5B%5D))(int dataType, int w, int h, int[] bitMasks)            Constructs a SinglePixelPackedSampleModel with bitMasks.length bands. |
| [**SinglePixelPackedSampleModel**](http://docs.google.com/java/awt/image/SinglePixelPackedSampleModel.html#SinglePixelPackedSampleModel(int,%20int,%20int,%20int,%20int%5B%5D))(int dataType, int w, int h, int scanlineStride, int[] bitMasks)            Constructs a SinglePixelPackedSampleModel with bitMasks.length bands and a scanline stride equal to scanlineStride data array elements. |

| **Method Summary** | |
| --- | --- |
| [SampleModel](http://docs.google.com/java/awt/image/SampleModel.html) | [**createCompatibleSampleModel**](http://docs.google.com/java/awt/image/SinglePixelPackedSampleModel.html#createCompatibleSampleModel(int,%20int))(int w, int h)            Creates a new SinglePixelPackedSampleModel with the specified width and height. |
| [DataBuffer](http://docs.google.com/java/awt/image/DataBuffer.html) | [**createDataBuffer**](http://docs.google.com/java/awt/image/SinglePixelPackedSampleModel.html#createDataBuffer())()            Creates a DataBuffer that corresponds to this SinglePixelPackedSampleModel. |
| [SampleModel](http://docs.google.com/java/awt/image/SampleModel.html) | [**createSubsetSampleModel**](http://docs.google.com/java/awt/image/SinglePixelPackedSampleModel.html#createSubsetSampleModel(int%5B%5D))(int[] bands)            This creates a new SinglePixelPackedSampleModel with a subset of the bands of this SinglePixelPackedSampleModel. |
| boolean | [**equals**](http://docs.google.com/java/awt/image/SinglePixelPackedSampleModel.html#equals(java.lang.Object))([Object](http://docs.google.com/java/lang/Object.html) o)            Indicates whether some other object is "equal to" this one. |
| int[] | [**getBitMasks**](http://docs.google.com/java/awt/image/SinglePixelPackedSampleModel.html#getBitMasks())()            Returns the bit masks for all bands. |
| int[] | [**getBitOffsets**](http://docs.google.com/java/awt/image/SinglePixelPackedSampleModel.html#getBitOffsets())()            Returns the bit offsets into the data array element representing a pixel for all bands. |
| [Object](http://docs.google.com/java/lang/Object.html) | [**getDataElements**](http://docs.google.com/java/awt/image/SinglePixelPackedSampleModel.html#getDataElements(int,%20int,%20java.lang.Object,%20java.awt.image.DataBuffer))(int x, int y, [Object](http://docs.google.com/java/lang/Object.html) obj, [DataBuffer](http://docs.google.com/java/awt/image/DataBuffer.html) data)            Returns data for a single pixel in a primitive array of type TransferType. |
| int | [**getNumDataElements**](http://docs.google.com/java/awt/image/SinglePixelPackedSampleModel.html#getNumDataElements())()            Returns the number of data elements needed to transfer one pixel via the getDataElements and setDataElements methods. |
| int | [**getOffset**](http://docs.google.com/java/awt/image/SinglePixelPackedSampleModel.html#getOffset(int,%20int))(int x, int y)            Returns the offset (in data array elements) of pixel (x,y). |
| int[] | [**getPixel**](http://docs.google.com/java/awt/image/SinglePixelPackedSampleModel.html#getPixel(int,%20int,%20int%5B%5D,%20java.awt.image.DataBuffer))(int x, int y, int[] iArray, [DataBuffer](http://docs.google.com/java/awt/image/DataBuffer.html) data)            Returns all samples in for the specified pixel in an int array. |
| int[] | [**getPixels**](http://docs.google.com/java/awt/image/SinglePixelPackedSampleModel.html#getPixels(int,%20int,%20int,%20int,%20int%5B%5D,%20java.awt.image.DataBuffer))(int x, int y, int w, int h, int[] iArray, [DataBuffer](http://docs.google.com/java/awt/image/DataBuffer.html) data)            Returns all samples for the specified rectangle of pixels in an int array, one sample per array element. |
| int | [**getSample**](http://docs.google.com/java/awt/image/SinglePixelPackedSampleModel.html#getSample(int,%20int,%20int,%20java.awt.image.DataBuffer))(int x, int y, int b, [DataBuffer](http://docs.google.com/java/awt/image/DataBuffer.html) data)            Returns as int the sample in a specified band for the pixel located at (x,y). |
| int[] | [**getSamples**](http://docs.google.com/java/awt/image/SinglePixelPackedSampleModel.html#getSamples(int,%20int,%20int,%20int,%20int,%20int%5B%5D,%20java.awt.image.DataBuffer))(int x, int y, int w, int h, int b, int[] iArray, [DataBuffer](http://docs.google.com/java/awt/image/DataBuffer.html) data)            Returns the samples for a specified band for the specified rectangle of pixels in an int array, one sample per array element. |
| int[] | [**getSampleSize**](http://docs.google.com/java/awt/image/SinglePixelPackedSampleModel.html#getSampleSize())()            Returns the number of bits per sample for all bands. |
| int | [**getSampleSize**](http://docs.google.com/java/awt/image/SinglePixelPackedSampleModel.html#getSampleSize(int))(int band)            Returns the number of bits per sample for the specified band. |
| int | [**getScanlineStride**](http://docs.google.com/java/awt/image/SinglePixelPackedSampleModel.html#getScanlineStride())()            Returns the scanline stride of this SinglePixelPackedSampleModel. |
| int | [**hashCode**](http://docs.google.com/java/awt/image/SinglePixelPackedSampleModel.html#hashCode())()            Returns a hash code value for the object. |
| void | [**setDataElements**](http://docs.google.com/java/awt/image/SinglePixelPackedSampleModel.html#setDataElements(int,%20int,%20java.lang.Object,%20java.awt.image.DataBuffer))(int x, int y, [Object](http://docs.google.com/java/lang/Object.html) obj, [DataBuffer](http://docs.google.com/java/awt/image/DataBuffer.html) data)            Sets the data for a single pixel in the specified DataBuffer from a primitive array of type TransferType. |
| void | [**setPixel**](http://docs.google.com/java/awt/image/SinglePixelPackedSampleModel.html#setPixel(int,%20int,%20int%5B%5D,%20java.awt.image.DataBuffer))(int x, int y, int[] iArray, [DataBuffer](http://docs.google.com/java/awt/image/DataBuffer.html) data)            Sets a pixel in the DataBuffer using an int array of samples for input. |
| void | [**setPixels**](http://docs.google.com/java/awt/image/SinglePixelPackedSampleModel.html#setPixels(int,%20int,%20int,%20int,%20int%5B%5D,%20java.awt.image.DataBuffer))(int x, int y, int w, int h, int[] iArray, [DataBuffer](http://docs.google.com/java/awt/image/DataBuffer.html) data)            Sets all samples for a rectangle of pixels from an int array containing one sample per array element. |
| void | [**setSample**](http://docs.google.com/java/awt/image/SinglePixelPackedSampleModel.html#setSample(int,%20int,%20int,%20int,%20java.awt.image.DataBuffer))(int x, int y, int b, int s, [DataBuffer](http://docs.google.com/java/awt/image/DataBuffer.html) data)            Sets a sample in the specified band for the pixel located at (x,y) in the DataBuffer using an int for input. |
| void | [**setSamples**](http://docs.google.com/java/awt/image/SinglePixelPackedSampleModel.html#setSamples(int,%20int,%20int,%20int,%20int,%20int%5B%5D,%20java.awt.image.DataBuffer))(int x, int y, int w, int h, int b, int[] iArray, [DataBuffer](http://docs.google.com/java/awt/image/DataBuffer.html) data)            Sets the samples in the specified band for the specified rectangle of pixels from an int array containing one sample per array element. |

| **Methods inherited from class java.awt.image.**[**SampleModel**](http://docs.google.com/java/awt/image/SampleModel.html) |
| --- |
| [getDataElements](http://docs.google.com/java/awt/image/SampleModel.html#getDataElements(int,%20int,%20int,%20int,%20java.lang.Object,%20java.awt.image.DataBuffer)), [getDataType](http://docs.google.com/java/awt/image/SampleModel.html#getDataType()), [getHeight](http://docs.google.com/java/awt/image/SampleModel.html#getHeight()), [getNumBands](http://docs.google.com/java/awt/image/SampleModel.html#getNumBands()), [getPixel](http://docs.google.com/java/awt/image/SampleModel.html#getPixel(int,%20int,%20double%5B%5D,%20java.awt.image.DataBuffer)), [getPixel](http://docs.google.com/java/awt/image/SampleModel.html#getPixel(int,%20int,%20float%5B%5D,%20java.awt.image.DataBuffer)), [getPixels](http://docs.google.com/java/awt/image/SampleModel.html#getPixels(int,%20int,%20int,%20int,%20double%5B%5D,%20java.awt.image.DataBuffer)), [getPixels](http://docs.google.com/java/awt/image/SampleModel.html#getPixels(int,%20int,%20int,%20int,%20float%5B%5D,%20java.awt.image.DataBuffer)), [getSampleDouble](http://docs.google.com/java/awt/image/SampleModel.html#getSampleDouble(int,%20int,%20int,%20java.awt.image.DataBuffer)), [getSampleFloat](http://docs.google.com/java/awt/image/SampleModel.html#getSampleFloat(int,%20int,%20int,%20java.awt.image.DataBuffer)), [getSamples](http://docs.google.com/java/awt/image/SampleModel.html#getSamples(int,%20int,%20int,%20int,%20int,%20double%5B%5D,%20java.awt.image.DataBuffer)), [getSamples](http://docs.google.com/java/awt/image/SampleModel.html#getSamples(int,%20int,%20int,%20int,%20int,%20float%5B%5D,%20java.awt.image.DataBuffer)), [getTransferType](http://docs.google.com/java/awt/image/SampleModel.html#getTransferType()), [getWidth](http://docs.google.com/java/awt/image/SampleModel.html#getWidth()), [setDataElements](http://docs.google.com/java/awt/image/SampleModel.html#setDataElements(int,%20int,%20int,%20int,%20java.lang.Object,%20java.awt.image.DataBuffer)), [setPixel](http://docs.google.com/java/awt/image/SampleModel.html#setPixel(int,%20int,%20double%5B%5D,%20java.awt.image.DataBuffer)), [setPixel](http://docs.google.com/java/awt/image/SampleModel.html#setPixel(int,%20int,%20float%5B%5D,%20java.awt.image.DataBuffer)), [setPixels](http://docs.google.com/java/awt/image/SampleModel.html#setPixels(int,%20int,%20int,%20int,%20double%5B%5D,%20java.awt.image.DataBuffer)), [setPixels](http://docs.google.com/java/awt/image/SampleModel.html#setPixels(int,%20int,%20int,%20int,%20float%5B%5D,%20java.awt.image.DataBuffer)), [setSample](http://docs.google.com/java/awt/image/SampleModel.html#setSample(int,%20int,%20int,%20double,%20java.awt.image.DataBuffer)), [setSample](http://docs.google.com/java/awt/image/SampleModel.html#setSample(int,%20int,%20int,%20float,%20java.awt.image.DataBuffer)), [setSamples](http://docs.google.com/java/awt/image/SampleModel.html#setSamples(int,%20int,%20int,%20int,%20int,%20double%5B%5D,%20java.awt.image.DataBuffer)), [setSamples](http://docs.google.com/java/awt/image/SampleModel.html#setSamples(int,%20int,%20int,%20int,%20int,%20float%5B%5D,%20java.awt.image.DataBuffer)) |

| **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)) |

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

### SinglePixelPackedSampleModel

public **SinglePixelPackedSampleModel**(int dataType,  
 int w,  
 int h,  
 int[] bitMasks)

Constructs a SinglePixelPackedSampleModel with bitMasks.length bands. Each sample is stored in a data array element in the position of its corresponding bit mask. Each bit mask must be contiguous and masks must not overlap.

**Parameters:**dataType - The data type for storing samples.w - The width (in pixels) of the region of the image data described.h - The height (in pixels) of the region of the image data described.bitMasks - The bit masks for all bands. **Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if dataType is not either DataBuffer.TYPE\_BYTE, DataBuffer.TYPE\_USHORT, or DataBuffer.TYPE\_INT

### SinglePixelPackedSampleModel

public **SinglePixelPackedSampleModel**(int dataType,  
 int w,  
 int h,  
 int scanlineStride,  
 int[] bitMasks)

Constructs a SinglePixelPackedSampleModel with bitMasks.length bands and a scanline stride equal to scanlineStride data array elements. Each sample is stored in a data array element in the position of its corresponding bit mask. Each bit mask must be contiguous and masks must not overlap.

**Parameters:**dataType - The data type for storing samples.w - The width (in pixels) of the region of image data described.h - The height (in pixels) of the region of image data described.scanlineStride - The line stride of the image data.bitMasks - The bit masks for all bands. **Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if w or h is not greater than 0 [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if any mask in bitMask is not contiguous [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if dataType is not either DataBuffer.TYPE\_BYTE, DataBuffer.TYPE\_USHORT, or DataBuffer.TYPE\_INT

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

### getNumDataElements

public int **getNumDataElements**()

Returns the number of data elements needed to transfer one pixel via the getDataElements and setDataElements methods. For a SinglePixelPackedSampleModel, this is one.

**Specified by:**[getNumDataElements](http://docs.google.com/java/awt/image/SampleModel.html#getNumDataElements()) in class [SampleModel](http://docs.google.com/java/awt/image/SampleModel.html) **Returns:**the number of data elements.**See Also:**[SampleModel.getDataElements(int, int, Object, DataBuffer)](http://docs.google.com/java/awt/image/SampleModel.html#getDataElements(int,%20int,%20java.lang.Object,%20java.awt.image.DataBuffer)), [SampleModel.getDataElements(int, int, int, int, Object, DataBuffer)](http://docs.google.com/java/awt/image/SampleModel.html#getDataElements(int,%20int,%20int,%20int,%20java.lang.Object,%20java.awt.image.DataBuffer)), [SampleModel.setDataElements(int, int, Object, DataBuffer)](http://docs.google.com/java/awt/image/SampleModel.html#setDataElements(int,%20int,%20java.lang.Object,%20java.awt.image.DataBuffer)), [SampleModel.setDataElements(int, int, int, int, Object, DataBuffer)](http://docs.google.com/java/awt/image/SampleModel.html#setDataElements(int,%20int,%20int,%20int,%20java.lang.Object,%20java.awt.image.DataBuffer)), [SampleModel.getTransferType()](http://docs.google.com/java/awt/image/SampleModel.html#getTransferType())

### createCompatibleSampleModel

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

Creates a new SinglePixelPackedSampleModel with the specified width and height. The new SinglePixelPackedSampleModel will have the same storage data type and bit masks as this SinglePixelPackedSampleModel.

**Specified by:**[createCompatibleSampleModel](http://docs.google.com/java/awt/image/SampleModel.html#createCompatibleSampleModel(int,%20int)) in class [SampleModel](http://docs.google.com/java/awt/image/SampleModel.html) **Parameters:**w - the width of the resulting SampleModelh - the height of the resulting SampleModel **Returns:**a SinglePixelPackedSampleModel with the specified width and height. **Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if w or h is not greater than 0

### createDataBuffer

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

Creates a DataBuffer that corresponds to this SinglePixelPackedSampleModel. The DataBuffer's data type and size will be consistent with this SinglePixelPackedSampleModel. The DataBuffer will have a single bank.

**Specified by:**[createDataBuffer](http://docs.google.com/java/awt/image/SampleModel.html#createDataBuffer()) in class [SampleModel](http://docs.google.com/java/awt/image/SampleModel.html) **Returns:**a DataBuffer corresponding to this SampleModel.

### getSampleSize

public int[] **getSampleSize**()

Returns the number of bits per sample for all bands.

**Specified by:**[getSampleSize](http://docs.google.com/java/awt/image/SampleModel.html#getSampleSize()) in class [SampleModel](http://docs.google.com/java/awt/image/SampleModel.html) **Returns:**the size of samples for all bands.

### getSampleSize

public int **getSampleSize**(int band)

Returns the number of bits per sample for the specified band.

**Specified by:**[getSampleSize](http://docs.google.com/java/awt/image/SampleModel.html#getSampleSize(int)) in class [SampleModel](http://docs.google.com/java/awt/image/SampleModel.html) **Parameters:**band - the specified band **Returns:**the size of the samples of the specified band.

### getOffset

public int **getOffset**(int x,  
 int y)

Returns the offset (in data array elements) of pixel (x,y). The data element containing pixel x,y can be retrieved from a DataBuffer data with a SinglePixelPackedSampleModel sppsm as:

data.getElem(sppsm.getOffset(x, y));

**Parameters:**x - the X coordinate of the specified pixely - the Y coordinate of the specified pixel **Returns:**the offset of the specified pixel.

### getBitOffsets

public int[] **getBitOffsets**()

Returns the bit offsets into the data array element representing a pixel for all bands.

**Returns:**the bit offsets representing a pixel for all bands.

### getBitMasks

public int[] **getBitMasks**()

Returns the bit masks for all bands.

**Returns:**the bit masks for all bands.

### getScanlineStride

public int **getScanlineStride**()

Returns the scanline stride of this SinglePixelPackedSampleModel.

**Returns:**the scanline stride of this SinglePixelPackedSampleModel.

### createSubsetSampleModel

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

This creates a new SinglePixelPackedSampleModel with a subset of the bands of this SinglePixelPackedSampleModel. The new SinglePixelPackedSampleModel can be used with any DataBuffer that the existing SinglePixelPackedSampleModel can be used with. The new SinglePixelPackedSampleModel/DataBuffer combination will represent an image with a subset of the bands of the original SinglePixelPackedSampleModel/DataBuffer combination.

**Specified by:**[createSubsetSampleModel](http://docs.google.com/java/awt/image/SampleModel.html#createSubsetSampleModel(int%5B%5D)) in class [SampleModel](http://docs.google.com/java/awt/image/SampleModel.html) **Parameters:**bands - the subset of bands of this SampleModel **Returns:**a SampleModel with a subset of bands of this SampleModel. **Throws:** [RasterFormatException](http://docs.google.com/java/awt/image/RasterFormatException.html) - if the length of the bands argument is greater than the number of bands in the sample model.

### getDataElements

public [Object](http://docs.google.com/java/lang/Object.html) **getDataElements**(int x,  
 int y,  
 [Object](http://docs.google.com/java/lang/Object.html) obj,  
 [DataBuffer](http://docs.google.com/java/awt/image/DataBuffer.html) data)

Returns data for a single pixel in a primitive array of type TransferType. For a SinglePixelPackedSampleModel, the array will have one element, and the type will be the same as the storage data type. Generally, obj should be passed in as null, so that the Object will be created automatically and will be of the right primitive data type.

The following code illustrates transferring data for one pixel from DataBuffer db1, whose storage layout is described by SinglePixelPackedSampleModel sppsm1, to DataBuffer db2, whose storage layout is described by SinglePixelPackedSampleModel sppsm2. The transfer will generally be more efficient than using getPixel/setPixel.

SinglePixelPackedSampleModel sppsm1, sppsm2;  
 DataBufferInt db1, db2;  
 sppsm2.setDataElements(x, y, sppsm1.getDataElements(x, y, null,  
 db1), db2);

Using getDataElements/setDataElements to transfer between two DataBuffer/SampleModel pairs is legitimate if the SampleModels have the same number of bands, corresponding bands have the same number of bits per sample, and the TransferTypes are the same.

If obj is non-null, it should be a primitive array of type TransferType. Otherwise, a ClassCastException is thrown. An ArrayIndexOutOfBoundsException may be thrown if the coordinates are not in bounds, or if obj is non-null and is not large enough to hold the pixel data.

**Specified by:**[getDataElements](http://docs.google.com/java/awt/image/SampleModel.html#getDataElements(int,%20int,%20java.lang.Object,%20java.awt.image.DataBuffer)) in class [SampleModel](http://docs.google.com/java/awt/image/SampleModel.html) **Parameters:**x - The X coordinate of the pixel location.y - The Y coordinate of the pixel location.obj - If non-null, a primitive array in which to return the pixel data.data - The DataBuffer containing the image data. **Returns:**the data for the specified pixel.**See Also:**[setDataElements(int, int, Object, DataBuffer)](http://docs.google.com/java/awt/image/SinglePixelPackedSampleModel.html#setDataElements(int,%20int,%20java.lang.Object,%20java.awt.image.DataBuffer))

### getPixel

public int[] **getPixel**(int x,  
 int y,  
 int[] iArray,  
 [DataBuffer](http://docs.google.com/java/awt/image/DataBuffer.html) data)

Returns all samples in for the specified pixel in an int array. ArrayIndexOutOfBoundsException may be thrown if the coordinates are not in bounds.

**Overrides:**[getPixel](http://docs.google.com/java/awt/image/SampleModel.html#getPixel(int,%20int,%20int%5B%5D,%20java.awt.image.DataBuffer)) in class [SampleModel](http://docs.google.com/java/awt/image/SampleModel.html) **Parameters:**x - The X coordinate of the pixel location.y - The Y coordinate of the pixel location.iArray - If non-null, returns the samples in this arraydata - The DataBuffer containing the image data. **Returns:**all samples for the specified pixel.**See Also:**[setPixel(int, int, int[], DataBuffer)](http://docs.google.com/java/awt/image/SinglePixelPackedSampleModel.html#setPixel(int,%20int,%20int%5B%5D,%20java.awt.image.DataBuffer))

### getPixels

public int[] **getPixels**(int x,  
 int y,  
 int w,  
 int h,  
 int[] iArray,  
 [DataBuffer](http://docs.google.com/java/awt/image/DataBuffer.html) data)

Returns all samples for the specified rectangle of pixels in an int array, one sample per array element. ArrayIndexOutOfBoundsException may be thrown if the coordinates are not in bounds.

**Overrides:**[getPixels](http://docs.google.com/java/awt/image/SampleModel.html#getPixels(int,%20int,%20int,%20int,%20int%5B%5D,%20java.awt.image.DataBuffer)) in class [SampleModel](http://docs.google.com/java/awt/image/SampleModel.html) **Parameters:**x - The X coordinate of the upper left pixel location.y - The Y coordinate of the upper left pixel location.w - The width of the pixel rectangle.h - The height of the pixel rectangle.iArray - If non-null, returns the samples in this array.data - The DataBuffer containing the image data. **Returns:**all samples for the specified region of pixels.**See Also:**[setPixels(int, int, int, int, int[], DataBuffer)](http://docs.google.com/java/awt/image/SinglePixelPackedSampleModel.html#setPixels(int,%20int,%20int,%20int,%20int%5B%5D,%20java.awt.image.DataBuffer))

### getSample

public int **getSample**(int x,  
 int y,  
 int b,  
 [DataBuffer](http://docs.google.com/java/awt/image/DataBuffer.html) data)

Returns as int the sample in a specified band for the pixel located at (x,y). ArrayIndexOutOfBoundsException may be thrown if the coordinates are not in bounds.

**Specified by:**[getSample](http://docs.google.com/java/awt/image/SampleModel.html#getSample(int,%20int,%20int,%20java.awt.image.DataBuffer)) in class [SampleModel](http://docs.google.com/java/awt/image/SampleModel.html) **Parameters:**x - The X coordinate of the pixel location.y - The Y coordinate of the pixel location.b - The band to return.data - The DataBuffer containing the image data. **Returns:**the sample in a specified band for the specified pixel.**See Also:**[setSample(int, int, int, int, DataBuffer)](http://docs.google.com/java/awt/image/SinglePixelPackedSampleModel.html#setSample(int,%20int,%20int,%20int,%20java.awt.image.DataBuffer))

### getSamples

public int[] **getSamples**(int x,  
 int y,  
 int w,  
 int h,  
 int b,  
 int[] iArray,  
 [DataBuffer](http://docs.google.com/java/awt/image/DataBuffer.html) data)

Returns the samples for a specified band for the specified rectangle of pixels in an int array, one sample per array element. ArrayIndexOutOfBoundsException may be thrown if the coordinates are not in bounds.

**Overrides:**[getSamples](http://docs.google.com/java/awt/image/SampleModel.html#getSamples(int,%20int,%20int,%20int,%20int,%20int%5B%5D,%20java.awt.image.DataBuffer)) in class [SampleModel](http://docs.google.com/java/awt/image/SampleModel.html) **Parameters:**x - The X coordinate of the upper left pixel location.y - The Y coordinate of the upper left pixel location.w - The width of the pixel rectangle.h - The height of the pixel rectangle.b - The band to return.iArray - If non-null, returns the samples in this array.data - The DataBuffer containing the image data. **Returns:**the samples for the specified band for the specified region of pixels.**See Also:**[setSamples(int, int, int, int, int, int[], DataBuffer)](http://docs.google.com/java/awt/image/SinglePixelPackedSampleModel.html#setSamples(int,%20int,%20int,%20int,%20int,%20int%5B%5D,%20java.awt.image.DataBuffer))

### setDataElements

public void **setDataElements**(int x,  
 int y,  
 [Object](http://docs.google.com/java/lang/Object.html) obj,  
 [DataBuffer](http://docs.google.com/java/awt/image/DataBuffer.html) data)

Sets the data for a single pixel in the specified DataBuffer from a primitive array of type TransferType. For a SinglePixelPackedSampleModel, only the first element of the array will hold valid data, and the type of the array must be the same as the storage data type of the SinglePixelPackedSampleModel.

The following code illustrates transferring data for one pixel from DataBuffer db1, whose storage layout is described by SinglePixelPackedSampleModel sppsm1, to DataBuffer db2, whose storage layout is described by SinglePixelPackedSampleModel sppsm2. The transfer will generally be more efficient than using getPixel/setPixel.

SinglePixelPackedSampleModel sppsm1, sppsm2;  
 DataBufferInt db1, db2;  
 sppsm2.setDataElements(x, y, sppsm1.getDataElements(x, y, null,  
 db1), db2);

Using getDataElements/setDataElements to transfer between two DataBuffer/SampleModel pairs is legitimate if the SampleModels have the same number of bands, corresponding bands have the same number of bits per sample, and the TransferTypes are the same.

obj must be a primitive array of type TransferType. Otherwise, a ClassCastException is thrown. An ArrayIndexOutOfBoundsException may be thrown if the coordinates are not in bounds, or if obj is not large enough to hold the pixel data.

**Specified by:**[setDataElements](http://docs.google.com/java/awt/image/SampleModel.html#setDataElements(int,%20int,%20java.lang.Object,%20java.awt.image.DataBuffer)) in class [SampleModel](http://docs.google.com/java/awt/image/SampleModel.html) **Parameters:**x - The X coordinate of the pixel location.y - The Y coordinate of the pixel location.obj - A primitive array containing pixel data.data - The DataBuffer containing the image data.**See Also:**[getDataElements(int, int, Object, DataBuffer)](http://docs.google.com/java/awt/image/SinglePixelPackedSampleModel.html#getDataElements(int,%20int,%20java.lang.Object,%20java.awt.image.DataBuffer))

### setPixel

public void **setPixel**(int x,  
 int y,  
 int[] iArray,  
 [DataBuffer](http://docs.google.com/java/awt/image/DataBuffer.html) data)

Sets a pixel in the DataBuffer using an int array of samples for input. ArrayIndexOutOfBoundsException may be thrown if the coordinates are not in bounds.

**Overrides:**[setPixel](http://docs.google.com/java/awt/image/SampleModel.html#setPixel(int,%20int,%20int%5B%5D,%20java.awt.image.DataBuffer)) in class [SampleModel](http://docs.google.com/java/awt/image/SampleModel.html) **Parameters:**x - The X coordinate of the pixel location.y - The Y coordinate of the pixel location.iArray - The input samples in an int array.data - The DataBuffer containing the image data.**See Also:**[getPixel(int, int, int[], DataBuffer)](http://docs.google.com/java/awt/image/SinglePixelPackedSampleModel.html#getPixel(int,%20int,%20int%5B%5D,%20java.awt.image.DataBuffer))

### setPixels

public void **setPixels**(int x,  
 int y,  
 int w,  
 int h,  
 int[] iArray,  
 [DataBuffer](http://docs.google.com/java/awt/image/DataBuffer.html) data)

Sets all samples for a rectangle of pixels from an int array containing one sample per array element. ArrayIndexOutOfBoundsException may be thrown if the coordinates are not in bounds.

**Overrides:**[setPixels](http://docs.google.com/java/awt/image/SampleModel.html#setPixels(int,%20int,%20int,%20int,%20int%5B%5D,%20java.awt.image.DataBuffer)) in class [SampleModel](http://docs.google.com/java/awt/image/SampleModel.html) **Parameters:**x - The X coordinate of the upper left pixel location.y - The Y coordinate of the upper left pixel location.w - The width of the pixel rectangle.h - The height of the pixel rectangle.iArray - The input samples in an int array.data - The DataBuffer containing the image data.**See Also:**[getPixels(int, int, int, int, int[], DataBuffer)](http://docs.google.com/java/awt/image/SinglePixelPackedSampleModel.html#getPixels(int,%20int,%20int,%20int,%20int%5B%5D,%20java.awt.image.DataBuffer))

### setSample

public void **setSample**(int x,  
 int y,  
 int b,  
 int s,  
 [DataBuffer](http://docs.google.com/java/awt/image/DataBuffer.html) data)

Sets a sample in the specified band for the pixel located at (x,y) in the DataBuffer using an int for input. ArrayIndexOutOfBoundsException may be thrown if the coordinates are not in bounds.

**Specified by:**[setSample](http://docs.google.com/java/awt/image/SampleModel.html#setSample(int,%20int,%20int,%20int,%20java.awt.image.DataBuffer)) in class [SampleModel](http://docs.google.com/java/awt/image/SampleModel.html) **Parameters:**x - The X coordinate of the pixel location.y - The Y coordinate of the pixel location.b - The band to set.s - The input sample as an int.data - The DataBuffer containing the image data.**See Also:**[getSample(int, int, int, DataBuffer)](http://docs.google.com/java/awt/image/SinglePixelPackedSampleModel.html#getSample(int,%20int,%20int,%20java.awt.image.DataBuffer))

### setSamples

public void **setSamples**(int x,  
 int y,  
 int w,  
 int h,  
 int b,  
 int[] iArray,  
 [DataBuffer](http://docs.google.com/java/awt/image/DataBuffer.html) data)

Sets the samples in the specified band for the specified rectangle of pixels from an int array containing one sample per array element. ArrayIndexOutOfBoundsException may be thrown if the coordinates are not in bounds.

**Overrides:**[setSamples](http://docs.google.com/java/awt/image/SampleModel.html#setSamples(int,%20int,%20int,%20int,%20int,%20int%5B%5D,%20java.awt.image.DataBuffer)) in class [SampleModel](http://docs.google.com/java/awt/image/SampleModel.html) **Parameters:**x - The X coordinate of the upper left pixel location.y - The Y coordinate of the upper left pixel location.w - The width of the pixel rectangle.h - The height of the pixel rectangle.b - The band to set.iArray - The input samples in an int array.data - The DataBuffer containing the image data.**See Also:**[getSamples(int, int, int, int, int, int[], DataBuffer)](http://docs.google.com/java/awt/image/SinglePixelPackedSampleModel.html#getSamples(int,%20int,%20int,%20int,%20int,%20int%5B%5D,%20java.awt.image.DataBuffer))

### equals

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

**Description copied from class:** [**Object**](http://docs.google.com/java/lang/Object.html#equals(java.lang.Object)) Indicates whether some other object is "equal to" this one.

The equals method implements an equivalence relation on non-null object references:

* It is *reflexive*: for any non-null reference value x, x.equals(x) should return true.
* It is *symmetric*: for any non-null reference values x and y, x.equals(y) should return true if and only if y.equals(x) returns true.
* It is *transitive*: for any non-null reference values x, y, and z, if x.equals(y) returns true and y.equals(z) returns true, then x.equals(z) should return true.
* It is *consistent*: for any non-null reference values x and y, multiple invocations of x.equals(y) consistently return true or consistently return false, provided no information used in equals comparisons on the objects is modified.
* For any non-null reference value x, x.equals(null) should return false.

The equals method for class Object implements the most discriminating possible equivalence relation on objects; that is, for any non-null reference values x and y, this method returns true if and only if x and y refer to the same object (x == y has the value true).

Note that it is generally necessary to override the hashCode method whenever this method is overridden, so as to maintain the general contract for the hashCode method, which states that equal objects must have equal hash codes.

**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 reference object with which to compare. **Returns:**true if this object is the same as the obj argument; false otherwise.**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**()

**Description copied from class:** [**Object**](http://docs.google.com/java/lang/Object.html#hashCode()) Returns a hash code value for the object. This method is supported for the benefit of hashtables such as those provided by java.util.Hashtable.

The general contract of hashCode is:

* Whenever it is invoked on the same object more than once during an execution of a Java application, the hashCode method must consistently return the same integer, provided no information used in equals comparisons on the object is modified. This integer need not remain consistent from one execution of an application to another execution of the same application.
* If two objects are equal according to the equals(Object) method, then calling the hashCode method on each of the two objects must produce the same integer result.
* It is *not* required that if two objects are unequal according to the [Object.equals(java.lang.Object)](http://docs.google.com/java/lang/Object.html#equals(java.lang.Object)) method, then calling the hashCode method on each of the two objects must produce distinct integer results. However, the programmer should be aware that producing distinct integer results for unequal objects may improve the performance of hashtables.

As much as is reasonably practical, the hashCode method defined by class Object does return distinct integers for distinct objects. (This is typically implemented by converting the internal address of the object into an integer, but this implementation technique is not required by the JavaTM programming language.)

**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 value for this object.**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/SinglePixelPackedSampleModel.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/ShortLookupTable.html)   [**NEXT CLASS**](http://docs.google.com/java/awt/image/TileObserver.html) | [**FRAMES**](http://docs.google.com/index.html?java/awt/image/SinglePixelPackedSampleModel.html)    [**NO FRAMES**](http://docs.google.com/SinglePixelPackedSampleModel.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | [FIELD](#2et92p0) | [CONSTR](#tyjcwt) | [METHOD](#3dy6vkm) | DETAIL: FIELD | [CONSTR](#2s8eyo1) | [METHOD](#26in1rg) |

[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).