| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/BandedSampleModel.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/BandCombineOp.html)   [**NEXT CLASS**](http://docs.google.com/java/awt/image/BufferedImage.html) | [**FRAMES**](http://docs.google.com/index.html?java/awt/image/BandedSampleModel.html)    [**NO FRAMES**](http://docs.google.com/BandedSampleModel.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | [FIELD](#2et92p0) | [CONSTR](#3dy6vkm) | [METHOD](#1t3h5sf) | DETAIL: FIELD | [CONSTR](#3rdcrjn) | [METHOD](#35nkun2) |

## **java.awt.image**

Class BandedSampleModel

[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.ComponentSampleModel](http://docs.google.com/java/awt/image/ComponentSampleModel.html)  
 **java.awt.image.BandedSampleModel**

public final class **BandedSampleModel**extends [ComponentSampleModel](http://docs.google.com/java/awt/image/ComponentSampleModel.html)

This class represents image data which is stored in a band interleaved fashion and for which each sample of a pixel occupies one data element of the DataBuffer. It subclasses ComponentSampleModel but provides a more efficent implementation for accessing band interleaved image data than is provided by ComponentSampleModel. This class should typically be used when working with images which store sample data for each band in a different bank of the DataBuffer. Accessor methods are provided so that image data can be manipulated directly. Pixel stride is the number of data array elements between two samples for the same band on the same scanline. The pixel stride for a BandedSampleModel is one. 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. Band offsets denote the number of data array elements from the first data array element of the bank of the DataBuffer holding each band to the first sample of the band. The bands are numbered from 0 to N-1. Bank indices denote the correspondence between a bank of the data buffer and a band of image data. 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\_SHORT](http://docs.google.com/java/awt/image/DataBuffer.html#TYPE_SHORT), [TYPE\_INT](http://docs.google.com/java/awt/image/DataBuffer.html#TYPE_INT), [TYPE\_FLOAT](http://docs.google.com/java/awt/image/DataBuffer.html#TYPE_FLOAT), and [TYPE\_DOUBLE](http://docs.google.com/java/awt/image/DataBuffer.html#TYPE_DOUBLE) datatypes

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

| **Fields inherited from class java.awt.image.**[**ComponentSampleModel**](http://docs.google.com/java/awt/image/ComponentSampleModel.html) |
| --- |
| [bandOffsets](http://docs.google.com/java/awt/image/ComponentSampleModel.html#bandOffsets), [bankIndices](http://docs.google.com/java/awt/image/ComponentSampleModel.html#bankIndices), [numBands](http://docs.google.com/java/awt/image/ComponentSampleModel.html#numBands), [numBanks](http://docs.google.com/java/awt/image/ComponentSampleModel.html#numBanks), [pixelStride](http://docs.google.com/java/awt/image/ComponentSampleModel.html#pixelStride), [scanlineStride](http://docs.google.com/java/awt/image/ComponentSampleModel.html#scanlineStride) |

| **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), [width](http://docs.google.com/java/awt/image/SampleModel.html#width) |

| **Constructor Summary** | |
| --- | --- |
| [**BandedSampleModel**](http://docs.google.com/java/awt/image/BandedSampleModel.html#BandedSampleModel(int,%20int,%20int,%20int))(int dataType, int w, int h, int numBands)            Constructs a BandedSampleModel with the specified parameters. |
| [**BandedSampleModel**](http://docs.google.com/java/awt/image/BandedSampleModel.html#BandedSampleModel(int,%20int,%20int,%20int,%20int%5B%5D,%20int%5B%5D))(int dataType, int w, int h, int scanlineStride, int[] bankIndices, int[] bandOffsets)            Constructs a BandedSampleModel with the specified parameters. |

| **Method Summary** | |
| --- | --- |
| [SampleModel](http://docs.google.com/java/awt/image/SampleModel.html) | [**createCompatibleSampleModel**](http://docs.google.com/java/awt/image/BandedSampleModel.html#createCompatibleSampleModel(int,%20int))(int w, int h)            Creates a new BandedSampleModel with the specified width and height. |
| [DataBuffer](http://docs.google.com/java/awt/image/DataBuffer.html) | [**createDataBuffer**](http://docs.google.com/java/awt/image/BandedSampleModel.html#createDataBuffer())()            Creates a DataBuffer that corresponds to this BandedSampleModel, The DataBuffer's data type, number of banks, and size will be consistent with this BandedSampleModel. |
| [SampleModel](http://docs.google.com/java/awt/image/SampleModel.html) | [**createSubsetSampleModel**](http://docs.google.com/java/awt/image/BandedSampleModel.html#createSubsetSampleModel(int%5B%5D))(int[] bands)            Creates a new BandedSampleModel with a subset of the bands of this BandedSampleModel. |
| [Object](http://docs.google.com/java/lang/Object.html) | [**getDataElements**](http://docs.google.com/java/awt/image/BandedSampleModel.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[] | [**getPixel**](http://docs.google.com/java/awt/image/BandedSampleModel.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 for the specified pixel in an int array. |
| int[] | [**getPixels**](http://docs.google.com/java/awt/image/BandedSampleModel.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 data array element. |
| int | [**getSample**](http://docs.google.com/java/awt/image/BandedSampleModel.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). |
| double | [**getSampleDouble**](http://docs.google.com/java/awt/image/BandedSampleModel.html#getSampleDouble(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 the sample in a specified band for a pixel located at (x,y) as a double. |
| float | [**getSampleFloat**](http://docs.google.com/java/awt/image/BandedSampleModel.html#getSampleFloat(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 the sample in a specified band for the pixel located at (x,y) as a float. |
| int[] | [**getSamples**](http://docs.google.com/java/awt/image/BandedSampleModel.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 in a specified band for the specified rectangle of pixels in an int array, one sample per data array element. |
| int | [**hashCode**](http://docs.google.com/java/awt/image/BandedSampleModel.html#hashCode())()            Returns a hash code value for the object. |
| void | [**setDataElements**](http://docs.google.com/java/awt/image/BandedSampleModel.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/BandedSampleModel.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/BandedSampleModel.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/BandedSampleModel.html#setSample(int,%20int,%20int,%20double,%20java.awt.image.DataBuffer))(int x, int y, int b, double 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 a double for input. |
| void | [**setSample**](http://docs.google.com/java/awt/image/BandedSampleModel.html#setSample(int,%20int,%20int,%20float,%20java.awt.image.DataBuffer))(int x, int y, int b, float 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 a float for input. |
| void | [**setSample**](http://docs.google.com/java/awt/image/BandedSampleModel.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/BandedSampleModel.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 data array element. |

| **Methods inherited from class java.awt.image.**[**ComponentSampleModel**](http://docs.google.com/java/awt/image/ComponentSampleModel.html) |
| --- |
| [equals](http://docs.google.com/java/awt/image/ComponentSampleModel.html#equals(java.lang.Object)), [getBandOffsets](http://docs.google.com/java/awt/image/ComponentSampleModel.html#getBandOffsets()), [getBankIndices](http://docs.google.com/java/awt/image/ComponentSampleModel.html#getBankIndices()), [getNumDataElements](http://docs.google.com/java/awt/image/ComponentSampleModel.html#getNumDataElements()), [getOffset](http://docs.google.com/java/awt/image/ComponentSampleModel.html#getOffset(int,%20int)), [getOffset](http://docs.google.com/java/awt/image/ComponentSampleModel.html#getOffset(int,%20int,%20int)), [getPixelStride](http://docs.google.com/java/awt/image/ComponentSampleModel.html#getPixelStride()), [getSampleSize](http://docs.google.com/java/awt/image/ComponentSampleModel.html#getSampleSize()), [getSampleSize](http://docs.google.com/java/awt/image/ComponentSampleModel.html#getSampleSize(int)), [getScanlineStride](http://docs.google.com/java/awt/image/ComponentSampleModel.html#getScanlineStride()) |

| **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)), [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)), [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** |
| --- |

### BandedSampleModel

public **BandedSampleModel**(int dataType,  
 int w,  
 int h,  
 int numBands)

Constructs a BandedSampleModel with the specified parameters. The pixel stride will be one data element. The scanline stride will be the same as the width. Each band will be stored in a separate bank and all band offsets will be zero.

**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.numBands - The number of bands for the image data. **Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if dataType is not one of the supported data types

### BandedSampleModel

public **BandedSampleModel**(int dataType,  
 int w,  
 int h,  
 int scanlineStride,  
 int[] bankIndices,  
 int[] bandOffsets)

Constructs a BandedSampleModel with the specified parameters. The number of bands will be inferred from the lengths of the bandOffsets bankIndices arrays, which must be equal. The pixel stride will be one data element.

**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 of the image data.bankIndices - The bank index for each band.bandOffsets - The band offset for each band. **Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if dataType is not one of the supported data types

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

### createCompatibleSampleModel

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

Creates a new BandedSampleModel with the specified width and height. The new BandedSampleModel will have the same number of bands, storage data type, and bank indices as this BandedSampleModel. The band offsets will be compressed such that the offset between bands will be w\*pixelStride and the minimum of all of the band offsets is zero.

**Overrides:**[createCompatibleSampleModel](http://docs.google.com/java/awt/image/ComponentSampleModel.html#createCompatibleSampleModel(int,%20int)) in class [ComponentSampleModel](http://docs.google.com/java/awt/image/ComponentSampleModel.html) **Parameters:**w - the width of the resulting BandedSampleModelh - the height of the resulting BandedSampleModel **Returns:**a new BandedSampleModel with the specified width and height. **Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if w or h equals either Integer.MAX\_VALUE or Integer.MIN\_VALUE [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if dataType is not one of the supported data types

### createSubsetSampleModel

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

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

**Overrides:**[createSubsetSampleModel](http://docs.google.com/java/awt/image/ComponentSampleModel.html#createSubsetSampleModel(int%5B%5D)) in class [ComponentSampleModel](http://docs.google.com/java/awt/image/ComponentSampleModel.html) **Parameters:**bands - a subset of bands from this ComponentSampleModel **Returns:**a ComponentSampleModel created with a subset of bands from this ComponentSampleModel. **Throws:** [RasterFormatException](http://docs.google.com/java/awt/image/RasterFormatException.html) - if the number of bands is greater than the number of banks in this sample model. [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if dataType is not one of the supported data types

### createDataBuffer

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

Creates a DataBuffer that corresponds to this BandedSampleModel, The DataBuffer's data type, number of banks, and size will be consistent with this BandedSampleModel.

**Overrides:**[createDataBuffer](http://docs.google.com/java/awt/image/ComponentSampleModel.html#createDataBuffer()) in class [ComponentSampleModel](http://docs.google.com/java/awt/image/ComponentSampleModel.html) **Returns:**a DataBuffer whose data type, number of banks and size are consistent with this ComponentSampleModel. **Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if dataType is not one of the supported types.

### 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 BandedSampleModel, this will be the same as the data type, and samples will be returned one per array element. 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 BandedSampleModel bsm1, to DataBuffer db2, whose storage layout is described by BandedSampleModel bsm2. The transfer will generally be more efficient than using getPixel/setPixel.

BandedSampleModel bsm1, bsm2;  
 DataBufferInt db1, db2;  
 bsm2.setDataElements(x, y, bsm1.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.

**Overrides:**[getDataElements](http://docs.google.com/java/awt/image/ComponentSampleModel.html#getDataElements(int,%20int,%20java.lang.Object,%20java.awt.image.DataBuffer)) in class [ComponentSampleModel](http://docs.google.com/java/awt/image/ComponentSampleModel.html) **Parameters:**x - The X coordinate of the pixel locationy - The Y coordinate of the pixel locationobj - 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/BandedSampleModel.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 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/ComponentSampleModel.html#getPixel(int,%20int,%20int%5B%5D,%20java.awt.image.DataBuffer)) in class [ComponentSampleModel](http://docs.google.com/java/awt/image/ComponentSampleModel.html) **Parameters:**x - The X coordinate of the pixel locationy - The Y coordinate of the pixel locationiArray - If non-null, returns the samples in this arraydata - The DataBuffer containing the image data **Returns:**the samples for the specified pixel.**See Also:**[setPixel(int, int, int[], DataBuffer)](http://docs.google.com/java/awt/image/BandedSampleModel.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 data array element. ArrayIndexOutOfBoundsException may be thrown if the coordinates are not in bounds.

**Overrides:**[getPixels](http://docs.google.com/java/awt/image/ComponentSampleModel.html#getPixels(int,%20int,%20int,%20int,%20int%5B%5D,%20java.awt.image.DataBuffer)) in class [ComponentSampleModel](http://docs.google.com/java/awt/image/ComponentSampleModel.html) **Parameters:**x - The X coordinate of the upper left pixel locationy - The Y coordinate of the upper left pixel locationw - The width of the pixel rectangleh - The height of the pixel rectangleiArray - If non-null, returns the samples in this arraydata - The DataBuffer containing the image data **Returns:**the samples for the pixels within the specified region.**See Also:**[setPixels(int, int, int, int, int[], DataBuffer)](http://docs.google.com/java/awt/image/BandedSampleModel.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.

**Overrides:**[getSample](http://docs.google.com/java/awt/image/ComponentSampleModel.html#getSample(int,%20int,%20int,%20java.awt.image.DataBuffer)) in class [ComponentSampleModel](http://docs.google.com/java/awt/image/ComponentSampleModel.html) **Parameters:**x - The X coordinate of the pixel locationy - The Y coordinate of the pixel locationb - The band to returndata - The DataBuffer containing the image data **Returns:**the sample in the specified band for the specified pixel.**See Also:**[setSample(int, int, int, int, DataBuffer)](http://docs.google.com/java/awt/image/BandedSampleModel.html#setSample(int,%20int,%20int,%20int,%20java.awt.image.DataBuffer))

### getSampleFloat

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

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

**Overrides:**[getSampleFloat](http://docs.google.com/java/awt/image/ComponentSampleModel.html#getSampleFloat(int,%20int,%20int,%20java.awt.image.DataBuffer)) in class [ComponentSampleModel](http://docs.google.com/java/awt/image/ComponentSampleModel.html) **Parameters:**x - The X coordinate of the pixel locationy - The Y coordinate of the pixel locationb - The band to returndata - The DataBuffer containing the image data **Returns:**a float value that represents the sample in the specified band for the specified pixel.

### getSampleDouble

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

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

**Overrides:**[getSampleDouble](http://docs.google.com/java/awt/image/ComponentSampleModel.html#getSampleDouble(int,%20int,%20int,%20java.awt.image.DataBuffer)) in class [ComponentSampleModel](http://docs.google.com/java/awt/image/ComponentSampleModel.html) **Parameters:**x - The X coordinate of the pixel locationy - The Y coordinate of the pixel locationb - The band to returndata - The DataBuffer containing the image data **Returns:**a double value that represents the sample in the specified band for the specified pixel.

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

**Overrides:**[getSamples](http://docs.google.com/java/awt/image/ComponentSampleModel.html#getSamples(int,%20int,%20int,%20int,%20int,%20int%5B%5D,%20java.awt.image.DataBuffer)) in class [ComponentSampleModel](http://docs.google.com/java/awt/image/ComponentSampleModel.html) **Parameters:**x - The X coordinate of the upper left pixel locationy - The Y coordinate of the upper left pixel locationw - The width of the pixel rectangleh - The height of the pixel rectangleb - The band to returniArray - If non-null, returns the samples in this arraydata - The DataBuffer containing the image data **Returns:**the samples in the specified band for the pixels within the specified region.**See Also:**[setSamples(int, int, int, int, int, int[], DataBuffer)](http://docs.google.com/java/awt/image/BandedSampleModel.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 BandedSampleModel, this will be the same as the data type, and samples are transferred one per array element.

The following code illustrates transferring data for one pixel from DataBuffer db1, whose storage layout is described by BandedSampleModel bsm1, to DataBuffer db2, whose storage layout is described by BandedSampleModel bsm2. The transfer will generally be more efficient than using getPixel/setPixel.

BandedSampleModel bsm1, bsm2;  
 DataBufferInt db1, db2;  
 bsm2.setDataElements(x, y, bsm1.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.

**Overrides:**[setDataElements](http://docs.google.com/java/awt/image/ComponentSampleModel.html#setDataElements(int,%20int,%20java.lang.Object,%20java.awt.image.DataBuffer)) in class [ComponentSampleModel](http://docs.google.com/java/awt/image/ComponentSampleModel.html) **Parameters:**x - The X coordinate of the pixel locationy - The Y coordinate of the pixel locationobj - If non-null, returns the primitive array in this objectdata - The DataBuffer containing the image data**See Also:**[getDataElements(int, int, Object, DataBuffer)](http://docs.google.com/java/awt/image/BandedSampleModel.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/ComponentSampleModel.html#setPixel(int,%20int,%20int%5B%5D,%20java.awt.image.DataBuffer)) in class [ComponentSampleModel](http://docs.google.com/java/awt/image/ComponentSampleModel.html) **Parameters:**x - The X coordinate of the pixel locationy - The Y coordinate of the pixel locationiArray - The input samples in an int arraydata - The DataBuffer containing the image data**See Also:**[getPixel(int, int, int[], DataBuffer)](http://docs.google.com/java/awt/image/BandedSampleModel.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/ComponentSampleModel.html#setPixels(int,%20int,%20int,%20int,%20int%5B%5D,%20java.awt.image.DataBuffer)) in class [ComponentSampleModel](http://docs.google.com/java/awt/image/ComponentSampleModel.html) **Parameters:**x - The X coordinate of the upper left pixel locationy - The Y coordinate of the upper left pixel locationw - The width of the pixel rectangleh - The height of the pixel rectangleiArray - The input samples in an int arraydata - The DataBuffer containing the image data**See Also:**[getPixels(int, int, int, int, int[], DataBuffer)](http://docs.google.com/java/awt/image/BandedSampleModel.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.

**Overrides:**[setSample](http://docs.google.com/java/awt/image/ComponentSampleModel.html#setSample(int,%20int,%20int,%20int,%20java.awt.image.DataBuffer)) in class [ComponentSampleModel](http://docs.google.com/java/awt/image/ComponentSampleModel.html) **Parameters:**x - The X coordinate of the pixel locationy - The Y coordinate of the pixel locationb - The band to sets - The input sample as an intdata - The DataBuffer containing the image data**See Also:**[getSample(int, int, int, DataBuffer)](http://docs.google.com/java/awt/image/BandedSampleModel.html#getSample(int,%20int,%20int,%20java.awt.image.DataBuffer))

### setSample

public void **setSample**(int x,  
 int y,  
 int b,  
 float 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 a float for input. ArrayIndexOutOfBoundsException may be thrown if the coordinates are not in bounds.

**Overrides:**[setSample](http://docs.google.com/java/awt/image/ComponentSampleModel.html#setSample(int,%20int,%20int,%20float,%20java.awt.image.DataBuffer)) in class [ComponentSampleModel](http://docs.google.com/java/awt/image/ComponentSampleModel.html) **Parameters:**x - The X coordinate of the pixel locationy - The Y coordinate of the pixel locationb - The band to sets - The input sample as a floatdata - The DataBuffer containing the image data**See Also:**[getSample(int, int, int, DataBuffer)](http://docs.google.com/java/awt/image/BandedSampleModel.html#getSample(int,%20int,%20int,%20java.awt.image.DataBuffer))

### setSample

public void **setSample**(int x,  
 int y,  
 int b,  
 double 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 a double for input. ArrayIndexOutOfBoundsException may be thrown if the coordinates are not in bounds.

**Overrides:**[setSample](http://docs.google.com/java/awt/image/ComponentSampleModel.html#setSample(int,%20int,%20int,%20double,%20java.awt.image.DataBuffer)) in class [ComponentSampleModel](http://docs.google.com/java/awt/image/ComponentSampleModel.html) **Parameters:**x - The X coordinate of the pixel locationy - The Y coordinate of the pixel locationb - The band to sets - The input sample as a doubledata - The DataBuffer containing the image data**See Also:**[getSample(int, int, int, DataBuffer)](http://docs.google.com/java/awt/image/BandedSampleModel.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 data array element. ArrayIndexOutOfBoundsException may be thrown if the coordinates are not in bounds.

**Overrides:**[setSamples](http://docs.google.com/java/awt/image/ComponentSampleModel.html#setSamples(int,%20int,%20int,%20int,%20int,%20int%5B%5D,%20java.awt.image.DataBuffer)) in class [ComponentSampleModel](http://docs.google.com/java/awt/image/ComponentSampleModel.html) **Parameters:**x - The X coordinate of the upper left pixel locationy - The Y coordinate of the upper left pixel locationw - The width of the pixel rectangleh - The height of the pixel rectangleb - The band to setiArray - The input sample arraydata - The DataBuffer containing the image data**See Also:**[getSamples(int, int, int, int, int, int[], DataBuffer)](http://docs.google.com/java/awt/image/BandedSampleModel.html#getSamples(int,%20int,%20int,%20int,%20int,%20int%5B%5D,%20java.awt.image.DataBuffer))

### 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/awt/image/ComponentSampleModel.html#hashCode()) in class [ComponentSampleModel](http://docs.google.com/java/awt/image/ComponentSampleModel.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/BandedSampleModel.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/BandCombineOp.html)   [**NEXT CLASS**](http://docs.google.com/java/awt/image/BufferedImage.html) | [**FRAMES**](http://docs.google.com/index.html?java/awt/image/BandedSampleModel.html)    [**NO FRAMES**](http://docs.google.com/BandedSampleModel.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | [FIELD](#2et92p0) | [CONSTR](#3dy6vkm) | [METHOD](#1t3h5sf) | DETAIL: FIELD | [CONSTR](#3rdcrjn) | [METHOD](#35nkun2) |

[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).