| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/SampleModel.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/RGBImageFilter.html)   [**NEXT CLASS**](http://docs.google.com/java/awt/image/ShortLookupTable.html) | [**FRAMES**](http://docs.google.com/index.html?java/awt/image/SampleModel.html)    [**NO FRAMES**](http://docs.google.com/SampleModel.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | [FIELD](#3znysh7) | [CONSTR](#2et92p0) | [METHOD](#tyjcwt) | DETAIL: [FIELD](#1t3h5sf) | [CONSTR](#26in1rg) | [METHOD](#35nkun2) |

## **java.awt.image**

Class SampleModel

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

**Direct Known Subclasses:** [ComponentSampleModel](http://docs.google.com/java/awt/image/ComponentSampleModel.html), [MultiPixelPackedSampleModel](http://docs.google.com/java/awt/image/MultiPixelPackedSampleModel.html), [SinglePixelPackedSampleModel](http://docs.google.com/java/awt/image/SinglePixelPackedSampleModel.html)

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

This abstract class defines an interface for extracting samples of pixels in an image. All image data is expressed as a collection of pixels. Each pixel consists of a number of samples. A sample is a datum for one band of an image and a band consists of all samples of a particular type in an image. For example, a pixel might contain three samples representing its red, green and blue components. There are three bands in the image containing this pixel. One band consists of all the red samples from all pixels in the image. The second band consists of all the green samples and the remaining band consists of all of the blue samples. The pixel can be stored in various formats. For example, all samples from a particular band can be stored contiguously or all samples from a single pixel can be stored contiguously.

Subclasses of SampleModel specify the types of samples they can represent (e.g. unsigned 8-bit byte, signed 16-bit short, etc.) and may specify how the samples are organized in memory. In the Java 2D(tm) API, built-in image processing operators may not operate on all possible sample types, but generally will work for unsigned integral samples of 16 bits or less. Some operators support a wider variety of sample types.

A collection of pixels is represented as a Raster, which consists of a DataBuffer and a SampleModel. The SampleModel allows access to samples in the DataBuffer and may provide low-level information that a programmer can use to directly manipulate samples and pixels in the DataBuffer.

This class is generally a fall back method for dealing with images. More efficient code will cast the SampleModel to the appropriate subclass and extract the information needed to directly manipulate pixels in the DataBuffer.

**See Also:**[DataBuffer](http://docs.google.com/java/awt/image/DataBuffer.html), [Raster](http://docs.google.com/java/awt/image/Raster.html), [ComponentSampleModel](http://docs.google.com/java/awt/image/ComponentSampleModel.html), [PixelInterleavedSampleModel](http://docs.google.com/java/awt/image/PixelInterleavedSampleModel.html), [BandedSampleModel](http://docs.google.com/java/awt/image/BandedSampleModel.html), [MultiPixelPackedSampleModel](http://docs.google.com/java/awt/image/MultiPixelPackedSampleModel.html), [SinglePixelPackedSampleModel](http://docs.google.com/java/awt/image/SinglePixelPackedSampleModel.html)

| **Field Summary** | |
| --- | --- |
| protected  int | [**dataType**](http://docs.google.com/java/awt/image/SampleModel.html#dataType)            Data type of the DataBuffer storing the pixel data. |
| protected  int | [**height**](http://docs.google.com/java/awt/image/SampleModel.html#height)            Height in pixels of the region of image data that this SampleModel describes. |
| protected  int | [**numBands**](http://docs.google.com/java/awt/image/SampleModel.html#numBands)            Number of bands of the image data that this SampleModel describes. |
| protected  int | [**width**](http://docs.google.com/java/awt/image/SampleModel.html#width)            Width in pixels of the region of image data that this SampleModel describes. |

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

| **Method Summary** | |
| --- | --- |
| abstract  [SampleModel](http://docs.google.com/java/awt/image/SampleModel.html) | [**createCompatibleSampleModel**](http://docs.google.com/java/awt/image/SampleModel.html#createCompatibleSampleModel(int,%20int))(int w, int h)            Creates a SampleModel which describes data in this SampleModel's format, but with a different width and height. |
| abstract  [DataBuffer](http://docs.google.com/java/awt/image/DataBuffer.html) | [**createDataBuffer**](http://docs.google.com/java/awt/image/SampleModel.html#createDataBuffer())()            Creates a DataBuffer that corresponds to this SampleModel. |
| abstract  [SampleModel](http://docs.google.com/java/awt/image/SampleModel.html) | [**createSubsetSampleModel**](http://docs.google.com/java/awt/image/SampleModel.html#createSubsetSampleModel(int%5B%5D))(int[] bands)            Creates a new SampleModel with a subset of the bands of this SampleModel. |
| [Object](http://docs.google.com/java/lang/Object.html) | [**getDataElements**](http://docs.google.com/java/awt/image/SampleModel.html#getDataElements(int,%20int,%20int,%20int,%20java.lang.Object,%20java.awt.image.DataBuffer))(int x, int y, int w, int h, [Object](http://docs.google.com/java/lang/Object.html) obj, [DataBuffer](http://docs.google.com/java/awt/image/DataBuffer.html) data)            Returns the pixel data for the specified rectangle of pixels in a primitive array of type TransferType. |
| abstract  [Object](http://docs.google.com/java/lang/Object.html) | [**getDataElements**](http://docs.google.com/java/awt/image/SampleModel.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 | [**getDataType**](http://docs.google.com/java/awt/image/SampleModel.html#getDataType())()            Returns the data type of the DataBuffer storing the pixel data. |
| int | [**getHeight**](http://docs.google.com/java/awt/image/SampleModel.html#getHeight())()            Returns the height in pixels. |
| int | [**getNumBands**](http://docs.google.com/java/awt/image/SampleModel.html#getNumBands())()            Returns the total number of bands of image data. |
| abstract  int | [**getNumDataElements**](http://docs.google.com/java/awt/image/SampleModel.html#getNumDataElements())()            Returns the number of data elements needed to transfer a pixel via the getDataElements and setDataElements methods. |
| double[] | [**getPixel**](http://docs.google.com/java/awt/image/SampleModel.html#getPixel(int,%20int,%20double%5B%5D,%20java.awt.image.DataBuffer))(int x, int y, double[] dArray, [DataBuffer](http://docs.google.com/java/awt/image/DataBuffer.html) data)            Returns the samples for the specified pixel in an array of double. |
| float[] | [**getPixel**](http://docs.google.com/java/awt/image/SampleModel.html#getPixel(int,%20int,%20float%5B%5D,%20java.awt.image.DataBuffer))(int x, int y, float[] fArray, [DataBuffer](http://docs.google.com/java/awt/image/DataBuffer.html) data)            Returns the samples for the specified pixel in an array of float. |
| int[] | [**getPixel**](http://docs.google.com/java/awt/image/SampleModel.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 the samples for a specified pixel in an int array, one sample per array element. |
| double[] | [**getPixels**](http://docs.google.com/java/awt/image/SampleModel.html#getPixels(int,%20int,%20int,%20int,%20double%5B%5D,%20java.awt.image.DataBuffer))(int x, int y, int w, int h, double[] dArray, [DataBuffer](http://docs.google.com/java/awt/image/DataBuffer.html) data)            Returns all samples for a rectangle of pixels in a double array, one sample per array element. |
| float[] | [**getPixels**](http://docs.google.com/java/awt/image/SampleModel.html#getPixels(int,%20int,%20int,%20int,%20float%5B%5D,%20java.awt.image.DataBuffer))(int x, int y, int w, int h, float[] fArray, [DataBuffer](http://docs.google.com/java/awt/image/DataBuffer.html) data)            Returns all samples for a rectangle of pixels in a float array, one sample per array element. |
| int[] | [**getPixels**](http://docs.google.com/java/awt/image/SampleModel.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 a rectangle of pixels in an int array, one sample per array element. |
| abstract  int | [**getSample**](http://docs.google.com/java/awt/image/SampleModel.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 the sample in a specified band for the pixel located at (x,y) as an int. |
| double | [**getSampleDouble**](http://docs.google.com/java/awt/image/SampleModel.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/SampleModel.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. |
| double[] | [**getSamples**](http://docs.google.com/java/awt/image/SampleModel.html#getSamples(int,%20int,%20int,%20int,%20int,%20double%5B%5D,%20java.awt.image.DataBuffer))(int x, int y, int w, int h, int b, double[] dArray, [DataBuffer](http://docs.google.com/java/awt/image/DataBuffer.html) data)            Returns the samples for a specified band for a specified rectangle of pixels in a double array, one sample per array element. |
| float[] | [**getSamples**](http://docs.google.com/java/awt/image/SampleModel.html#getSamples(int,%20int,%20int,%20int,%20int,%20float%5B%5D,%20java.awt.image.DataBuffer))(int x, int y, int w, int h, int b, float[] fArray, [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 a float array, one sample per array element. |
| int[] | [**getSamples**](http://docs.google.com/java/awt/image/SampleModel.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. |
| abstract  int[] | [**getSampleSize**](http://docs.google.com/java/awt/image/SampleModel.html#getSampleSize())()            Returns the size in bits of samples for all bands. |
| abstract  int | [**getSampleSize**](http://docs.google.com/java/awt/image/SampleModel.html#getSampleSize(int))(int band)            Returns the size in bits of samples for the specified band. |
| int | [**getTransferType**](http://docs.google.com/java/awt/image/SampleModel.html#getTransferType())()            Returns the TransferType used to transfer pixels via the getDataElements and setDataElements methods. |
| int | [**getWidth**](http://docs.google.com/java/awt/image/SampleModel.html#getWidth())()            Returns the width in pixels. |
| void | [**setDataElements**](http://docs.google.com/java/awt/image/SampleModel.html#setDataElements(int,%20int,%20int,%20int,%20java.lang.Object,%20java.awt.image.DataBuffer))(int x, int y, int w, int h, [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 rectangle of pixels in the specified DataBuffer from a primitive array of type TransferType. |
| abstract  void | [**setDataElements**](http://docs.google.com/java/awt/image/SampleModel.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/SampleModel.html#setPixel(int,%20int,%20double%5B%5D,%20java.awt.image.DataBuffer))(int x, int y, double[] dArray, [DataBuffer](http://docs.google.com/java/awt/image/DataBuffer.html) data)            Sets a pixel in the DataBuffer using a double array of samples for input. |
| void | [**setPixel**](http://docs.google.com/java/awt/image/SampleModel.html#setPixel(int,%20int,%20float%5B%5D,%20java.awt.image.DataBuffer))(int x, int y, float[] fArray, [DataBuffer](http://docs.google.com/java/awt/image/DataBuffer.html) data)            Sets a pixel in the DataBuffer using a float array of samples for input. |
| void | [**setPixel**](http://docs.google.com/java/awt/image/SampleModel.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/SampleModel.html#setPixels(int,%20int,%20int,%20int,%20double%5B%5D,%20java.awt.image.DataBuffer))(int x, int y, int w, int h, double[] dArray, [DataBuffer](http://docs.google.com/java/awt/image/DataBuffer.html) data)            Sets all samples for a rectangle of pixels from a double array containing one sample per array element. |
| void | [**setPixels**](http://docs.google.com/java/awt/image/SampleModel.html#setPixels(int,%20int,%20int,%20int,%20float%5B%5D,%20java.awt.image.DataBuffer))(int x, int y, int w, int h, float[] fArray, [DataBuffer](http://docs.google.com/java/awt/image/DataBuffer.html) data)            Sets all samples for a rectangle of pixels from a float array containing one sample per array element. |
| void | [**setPixels**](http://docs.google.com/java/awt/image/SampleModel.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/SampleModel.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/SampleModel.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. |
| abstract  void | [**setSample**](http://docs.google.com/java/awt/image/SampleModel.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/SampleModel.html#setSamples(int,%20int,%20int,%20int,%20int,%20double%5B%5D,%20java.awt.image.DataBuffer))(int x, int y, int w, int h, int b, double[] dArray, [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 a double array containing one sample per array element. |
| void | [**setSamples**](http://docs.google.com/java/awt/image/SampleModel.html#setSamples(int,%20int,%20int,%20int,%20int,%20float%5B%5D,%20java.awt.image.DataBuffer))(int x, int y, int w, int h, int b, float[] fArray, [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 a float array containing one sample per array element. |
| void | [**setSamples**](http://docs.google.com/java/awt/image/SampleModel.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.lang.**[**Object**](http://docs.google.com/java/lang/Object.html) |
| --- |
| [clone](http://docs.google.com/java/lang/Object.html#clone()), [equals](http://docs.google.com/java/lang/Object.html#equals(java.lang.Object)), [finalize](http://docs.google.com/java/lang/Object.html#finalize()), [getClass](http://docs.google.com/java/lang/Object.html#getClass()), [hashCode](http://docs.google.com/java/lang/Object.html#hashCode()), [notify](http://docs.google.com/java/lang/Object.html#notify()), [notifyAll](http://docs.google.com/java/lang/Object.html#notifyAll()), [toString](http://docs.google.com/java/lang/Object.html#toString()), [wait](http://docs.google.com/java/lang/Object.html#wait()), [wait](http://docs.google.com/java/lang/Object.html#wait(long)), [wait](http://docs.google.com/java/lang/Object.html#wait(long,%20int)) |

| **Field Detail** |
| --- |

### width

protected int **width**

Width in pixels of the region of image data that this SampleModel describes.

### height

protected int **height**

Height in pixels of the region of image data that this SampleModel describes.

### numBands

protected int **numBands**

Number of bands of the image data that this SampleModel describes.

### dataType

protected int **dataType**

Data type of the DataBuffer storing the pixel data.

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

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

### SampleModel

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

Constructs a SampleModel with the specified parameters.

**Parameters:**dataType - The data type of the DataBuffer storing the pixel data.w - The width (in pixels) of the region of image data.h - The height (in pixels) of the region of image data.numBands - The number of bands of the image data. **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 the product of w and h is greater than Integer.MAX\_VALUE [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if dataType is not one of the supported data types

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

### getWidth

public final int **getWidth**()

Returns the width in pixels.

**Returns:**the width in pixels of the region of image data that this SampleModel describes.

### getHeight

public final int **getHeight**()

Returns the height in pixels.

**Returns:**the height in pixels of the region of image data that this SampleModel describes.

### getNumBands

public final int **getNumBands**()

Returns the total number of bands of image data.

**Returns:**the number of bands of image data that this SampleModel describes.

### getNumDataElements

public abstract int **getNumDataElements**()

Returns the number of data elements needed to transfer a pixel via the getDataElements and setDataElements methods. When pixels are transferred via these methods, they may be transferred in a packed or unpacked format, depending on the implementation of the SampleModel. Using these methods, pixels are transferred as an array of getNumDataElements() elements of a primitive type given by getTransferType(). The TransferType may or may not be the same as the storage DataType.

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

### getDataType

public final int **getDataType**()

Returns the data type of the DataBuffer storing the pixel data.

**Returns:**the data type.

### getTransferType

public int **getTransferType**()

Returns the TransferType used to transfer pixels via the getDataElements and setDataElements methods. When pixels are transferred via these methods, they may be transferred in a packed or unpacked format, depending on the implementation of the SampleModel. Using these methods, pixels are transferred as an array of getNumDataElements() elements of a primitive type given by getTransferType(). The TransferType may or may not be the same as the storage DataType. The TransferType will be one of the types defined in DataBuffer.

**Returns:**the transfer type.**See Also:**[getDataElements(int, int, Object, DataBuffer)](http://docs.google.com/java/awt/image/SampleModel.html#getDataElements(int,%20int,%20java.lang.Object,%20java.awt.image.DataBuffer)), [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)), [setDataElements(int, int, Object, DataBuffer)](http://docs.google.com/java/awt/image/SampleModel.html#setDataElements(int,%20int,%20java.lang.Object,%20java.awt.image.DataBuffer)), [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)), [getNumDataElements()](http://docs.google.com/java/awt/image/SampleModel.html#getNumDataElements()), [DataBuffer](http://docs.google.com/java/awt/image/DataBuffer.html)

### getPixel

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

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

**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. **Throws:** [NullPointerException](http://docs.google.com/java/lang/NullPointerException.html) - if data is null. [ArrayIndexOutOfBoundsException](http://docs.google.com/java/lang/ArrayIndexOutOfBoundsException.html) - if the coordinates are not in bounds, or if iArray is too small to hold the output.**See Also:**[setPixel(int, int, int[], DataBuffer)](http://docs.google.com/java/awt/image/SampleModel.html#setPixel(int,%20int,%20int%5B%5D,%20java.awt.image.DataBuffer))

### getDataElements

public abstract [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 image data supported by the Java 2D API, this will be one of DataBuffer.TYPE\_BYTE, DataBuffer.TYPE\_USHORT, DataBuffer.TYPE\_INT, DataBuffer.TYPE\_SHORT, DataBuffer.TYPE\_FLOAT, or DataBuffer.TYPE\_DOUBLE. Data may be returned in a packed format, thus increasing efficiency for data transfers. 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 SampleModel sm1, to DataBuffer db2, whose storage layout is described by SampleModel sm2. The transfer will generally be more efficient than using getPixel/setPixel.

SampleModel sm1, sm2;  
 DataBuffer db1, db2;  
 sm2.setDataElements(x, y, sm1.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.

**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 elements for the specified pixel. **Throws:** [NullPointerException](http://docs.google.com/java/lang/NullPointerException.html) - if data is null. [ArrayIndexOutOfBoundsException](http://docs.google.com/java/lang/ArrayIndexOutOfBoundsException.html) - if the coordinates are not in bounds, or if obj is too small to hold the output.**See Also:**[getNumDataElements()](http://docs.google.com/java/awt/image/SampleModel.html#getNumDataElements()), [getTransferType()](http://docs.google.com/java/awt/image/SampleModel.html#getTransferType()), [DataBuffer](http://docs.google.com/java/awt/image/DataBuffer.html), [setDataElements(int, int, Object, DataBuffer)](http://docs.google.com/java/awt/image/SampleModel.html#setDataElements(int,%20int,%20java.lang.Object,%20java.awt.image.DataBuffer))

### getDataElements

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

Returns the pixel data for the specified rectangle of pixels in a primitive array of type TransferType. For image data supported by the Java 2D API, this will be one of DataBuffer.TYPE\_BYTE, DataBuffer.TYPE\_USHORT, DataBuffer.TYPE\_INT, DataBuffer.TYPE\_SHORT, DataBuffer.TYPE\_FLOAT, or DataBuffer.TYPE\_DOUBLE. Data may be returned in a packed format, thus increasing efficiency for data transfers. 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 a rectangular region of pixels from DataBuffer db1, whose storage layout is described by SampleModel sm1, to DataBuffer db2, whose storage layout is described by SampleModel sm2. The transfer will generally be more efficient than using getPixels/setPixels.

SampleModel sm1, sm2;  
 DataBuffer db1, db2;  
 sm2.setDataElements(x, y, w, h, sm1.getDataElements(x, y, w,  
 h, 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.

**Parameters:**x - The minimum X coordinate of the pixel rectangle.y - The minimum Y coordinate of the pixel rectangle.w - The width of the pixel rectangle.h - The height of the pixel rectangle.obj - If non-null, a primitive array in which to return the pixel data.data - The DataBuffer containing the image data. **Returns:**the data elements for the specified region of pixels. **Throws:** [NullPointerException](http://docs.google.com/java/lang/NullPointerException.html) - if data is null. [ArrayIndexOutOfBoundsException](http://docs.google.com/java/lang/ArrayIndexOutOfBoundsException.html) - if the coordinates are not in bounds, or if obj is too small to hold the output.**See Also:**[getNumDataElements()](http://docs.google.com/java/awt/image/SampleModel.html#getNumDataElements()), [getTransferType()](http://docs.google.com/java/awt/image/SampleModel.html#getTransferType()), [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)), [DataBuffer](http://docs.google.com/java/awt/image/DataBuffer.html)

### setDataElements

public abstract 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 image data supported by the Java 2D API, this will be one of DataBuffer.TYPE\_BYTE, DataBuffer.TYPE\_USHORT, DataBuffer.TYPE\_INT, DataBuffer.TYPE\_SHORT, DataBuffer.TYPE\_FLOAT, or DataBuffer.TYPE\_DOUBLE. Data in the array may be in a packed format, thus increasing efficiency for data transfers.

The following code illustrates transferring data for one pixel from DataBuffer db1, whose storage layout is described by SampleModel sm1, to DataBuffer db2, whose storage layout is described by SampleModel sm2. The transfer will generally be more efficient than using getPixel/setPixel.

SampleModel sm1, sm2;  
 DataBuffer db1, db2;  
 sm2.setDataElements(x, y, sm1.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.

**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. **Throws:** [NullPointerException](http://docs.google.com/java/lang/NullPointerException.html) - if data is null. [ArrayIndexOutOfBoundsException](http://docs.google.com/java/lang/ArrayIndexOutOfBoundsException.html) - if the coordinates are not in bounds, or if obj is too small to hold the input.**See Also:**[getNumDataElements()](http://docs.google.com/java/awt/image/SampleModel.html#getNumDataElements()), [getTransferType()](http://docs.google.com/java/awt/image/SampleModel.html#getTransferType()), [getDataElements(int, int, Object, DataBuffer)](http://docs.google.com/java/awt/image/SampleModel.html#getDataElements(int,%20int,%20java.lang.Object,%20java.awt.image.DataBuffer)), [DataBuffer](http://docs.google.com/java/awt/image/DataBuffer.html)

### setDataElements

public void **setDataElements**(int x,  
 int y,  
 int w,  
 int h,  
 [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 rectangle of pixels in the specified DataBuffer from a primitive array of type TransferType. For image data supported by the Java 2D API, this will be one of DataBuffer.TYPE\_BYTE, DataBuffer.TYPE\_USHORT, DataBuffer.TYPE\_INT, DataBuffer.TYPE\_SHORT, DataBuffer.TYPE\_FLOAT, or DataBuffer.TYPE\_DOUBLE. Data in the array may be in a packed format, thus increasing efficiency for data transfers.

The following code illustrates transferring data for a rectangular region of pixels from DataBuffer db1, whose storage layout is described by SampleModel sm1, to DataBuffer db2, whose storage layout is described by SampleModel sm2. The transfer will generally be more efficient than using getPixels/setPixels.

SampleModel sm1, sm2;  
 DataBuffer db1, db2;  
 sm2.setDataElements(x, y, w, h, sm1.getDataElements(x, y, w, h,  
 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.

**Parameters:**x - The minimum X coordinate of the pixel rectangle.y - The minimum Y coordinate of the pixel rectangle.w - The width of the pixel rectangle.h - The height of the pixel rectangle.obj - A primitive array containing pixel data.data - The DataBuffer containing the image data. **Throws:** [NullPointerException](http://docs.google.com/java/lang/NullPointerException.html) - if data is null. [ArrayIndexOutOfBoundsException](http://docs.google.com/java/lang/ArrayIndexOutOfBoundsException.html) - if the coordinates are not in bounds, or if obj is too small to hold the input.**See Also:**[getNumDataElements()](http://docs.google.com/java/awt/image/SampleModel.html#getNumDataElements()), [getTransferType()](http://docs.google.com/java/awt/image/SampleModel.html#getTransferType()), [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)), [DataBuffer](http://docs.google.com/java/awt/image/DataBuffer.html)

### getPixel

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

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

**Parameters:**x - The X coordinate of the pixel location.y - The Y coordinate of the pixel location.fArray - If non-null, returns the samples in this array.data - The DataBuffer containing the image data. **Returns:**the samples for the specified pixel. **Throws:** [NullPointerException](http://docs.google.com/java/lang/NullPointerException.html) - if data is null. [ArrayIndexOutOfBoundsException](http://docs.google.com/java/lang/ArrayIndexOutOfBoundsException.html) - if the coordinates are not in bounds, or if fArray is too small to hold the output.**See Also:**[setPixel(int, int, float[], DataBuffer)](http://docs.google.com/java/awt/image/SampleModel.html#setPixel(int,%20int,%20float%5B%5D,%20java.awt.image.DataBuffer))

### getPixel

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

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

**Parameters:**x - The X coordinate of the pixel location.y - The Y coordinate of the pixel location.dArray - If non-null, returns the samples in this array.data - The DataBuffer containing the image data. **Returns:**the samples for the specified pixel. **Throws:** [NullPointerException](http://docs.google.com/java/lang/NullPointerException.html) - if data is null. [ArrayIndexOutOfBoundsException](http://docs.google.com/java/lang/ArrayIndexOutOfBoundsException.html) - if the coordinates are not in bounds, or if dArray is too small to hold the output.**See Also:**[setPixel(int, int, double[], DataBuffer)](http://docs.google.com/java/awt/image/SampleModel.html#setPixel(int,%20int,%20double%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 a rectangle of pixels in an int array, one sample per array element. ArrayIndexOutOfBoundsException may be thrown if the coordinates are not in bounds.

**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:**the samples for the specified region of pixels. **Throws:** [NullPointerException](http://docs.google.com/java/lang/NullPointerException.html) - if data is null. [ArrayIndexOutOfBoundsException](http://docs.google.com/java/lang/ArrayIndexOutOfBoundsException.html) - if the coordinates are not in bounds, or if iArray is too small to hold the output.**See Also:**[setPixels(int, int, int, int, int[], DataBuffer)](http://docs.google.com/java/awt/image/SampleModel.html#setPixels(int,%20int,%20int,%20int,%20int%5B%5D,%20java.awt.image.DataBuffer))

### getPixels

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

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

**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.fArray - If non-null, returns the samples in this array.data - The DataBuffer containing the image data. **Returns:**the samples for the specified region of pixels. **Throws:** [NullPointerException](http://docs.google.com/java/lang/NullPointerException.html) - if data is null. [ArrayIndexOutOfBoundsException](http://docs.google.com/java/lang/ArrayIndexOutOfBoundsException.html) - if the coordinates are not in bounds, or if fArray is too small to hold the output.**See Also:**[setPixels(int, int, int, int, float[], DataBuffer)](http://docs.google.com/java/awt/image/SampleModel.html#setPixels(int,%20int,%20int,%20int,%20float%5B%5D,%20java.awt.image.DataBuffer))

### getPixels

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

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

**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.dArray - If non-null, returns the samples in this array.data - The DataBuffer containing the image data. **Returns:**the samples for the specified region of pixels. **Throws:** [NullPointerException](http://docs.google.com/java/lang/NullPointerException.html) - if data is null. [ArrayIndexOutOfBoundsException](http://docs.google.com/java/lang/ArrayIndexOutOfBoundsException.html) - if the coordinates are not in bounds, or if dArray is too small to hold the output.**See Also:**[setPixels(int, int, int, int, double[], DataBuffer)](http://docs.google.com/java/awt/image/SampleModel.html#setPixels(int,%20int,%20int,%20int,%20double%5B%5D,%20java.awt.image.DataBuffer))

### getSample

public abstract int **getSample**(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 an int. ArrayIndexOutOfBoundsException may be thrown if the coordinates are not in bounds.

**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. **Throws:** [NullPointerException](http://docs.google.com/java/lang/NullPointerException.html) - if data is null. [ArrayIndexOutOfBoundsException](http://docs.google.com/java/lang/ArrayIndexOutOfBoundsException.html) - if the coordinates or the band index are not in bounds.**See Also:**[setSample(int, int, int, int, DataBuffer)](http://docs.google.com/java/awt/image/SampleModel.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.

**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. **Throws:** [NullPointerException](http://docs.google.com/java/lang/NullPointerException.html) - if data is null. [ArrayIndexOutOfBoundsException](http://docs.google.com/java/lang/ArrayIndexOutOfBoundsException.html) - if the coordinates or the band index are not in bounds.

### 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.

**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. **Throws:** [NullPointerException](http://docs.google.com/java/lang/NullPointerException.html) - if data is null. [ArrayIndexOutOfBoundsException](http://docs.google.com/java/lang/ArrayIndexOutOfBoundsException.html) - if the coordinates or the band index are not in bounds.

### 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.

**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. **Throws:** [NullPointerException](http://docs.google.com/java/lang/NullPointerException.html) - if data is null. [ArrayIndexOutOfBoundsException](http://docs.google.com/java/lang/ArrayIndexOutOfBoundsException.html) - if the coordinates or the band index are not in bounds, or if iArray is too small to hold the output.**See Also:**[setSamples(int, int, int, int, int, int[], DataBuffer)](http://docs.google.com/java/awt/image/SampleModel.html#setSamples(int,%20int,%20int,%20int,%20int,%20int%5B%5D,%20java.awt.image.DataBuffer))

### getSamples

public float[] **getSamples**(int x,  
 int y,  
 int w,  
 int h,  
 int b,  
 float[] fArray,  
 [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 a float array, one sample per array element. ArrayIndexOutOfBoundsException may be thrown if the coordinates are not in bounds.

**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.fArray - 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. **Throws:** [NullPointerException](http://docs.google.com/java/lang/NullPointerException.html) - if data is null. [ArrayIndexOutOfBoundsException](http://docs.google.com/java/lang/ArrayIndexOutOfBoundsException.html) - if the coordinates or the band index are not in bounds, or if fArray is too small to hold the output.**See Also:**[setSamples(int, int, int, int, int, float[], DataBuffer)](http://docs.google.com/java/awt/image/SampleModel.html#setSamples(int,%20int,%20int,%20int,%20int,%20float%5B%5D,%20java.awt.image.DataBuffer))

### getSamples

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

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

**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.dArray - 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. **Throws:** [NullPointerException](http://docs.google.com/java/lang/NullPointerException.html) - if data is null. [ArrayIndexOutOfBoundsException](http://docs.google.com/java/lang/ArrayIndexOutOfBoundsException.html) - if the coordinates or the band index are not in bounds, or if dArray is too small to hold the output.**See Also:**[setSamples(int, int, int, int, int, double[], DataBuffer)](http://docs.google.com/java/awt/image/SampleModel.html#setSamples(int,%20int,%20int,%20int,%20int,%20double%5B%5D,%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.

**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. **Throws:** [NullPointerException](http://docs.google.com/java/lang/NullPointerException.html) - if iArray or data is null. [ArrayIndexOutOfBoundsException](http://docs.google.com/java/lang/ArrayIndexOutOfBoundsException.html) - if the coordinates are not in bounds, or if iArray is too small to hold the input.**See Also:**[getPixel(int, int, int[], DataBuffer)](http://docs.google.com/java/awt/image/SampleModel.html#getPixel(int,%20int,%20int%5B%5D,%20java.awt.image.DataBuffer))

### setPixel

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

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

**Parameters:**x - The X coordinate of the pixel location.y - The Y coordinate of the pixel location.fArray - The input samples in a float array.data - The DataBuffer containing the image data. **Throws:** [NullPointerException](http://docs.google.com/java/lang/NullPointerException.html) - if fArray or data is null. [ArrayIndexOutOfBoundsException](http://docs.google.com/java/lang/ArrayIndexOutOfBoundsException.html) - if the coordinates are not in bounds, or if fArray is too small to hold the input.**See Also:**[getPixel(int, int, float[], DataBuffer)](http://docs.google.com/java/awt/image/SampleModel.html#getPixel(int,%20int,%20float%5B%5D,%20java.awt.image.DataBuffer))

### setPixel

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

Sets a pixel in the DataBuffer using a double array of samples for input.

**Parameters:**x - The X coordinate of the pixel location.y - The Y coordinate of the pixel location.dArray - The input samples in a double array.data - The DataBuffer containing the image data. **Throws:** [NullPointerException](http://docs.google.com/java/lang/NullPointerException.html) - if dArray or data is null. [ArrayIndexOutOfBoundsException](http://docs.google.com/java/lang/ArrayIndexOutOfBoundsException.html) - if the coordinates are not in bounds, or if fArray is too small to hold the input.**See Also:**[getPixel(int, int, double[], DataBuffer)](http://docs.google.com/java/awt/image/SampleModel.html#getPixel(int,%20int,%20double%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.

**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. **Throws:** [NullPointerException](http://docs.google.com/java/lang/NullPointerException.html) - if iArray or data is null. [ArrayIndexOutOfBoundsException](http://docs.google.com/java/lang/ArrayIndexOutOfBoundsException.html) - if the coordinates are not in bounds, or if iArray is too small to hold the input.**See Also:**[getPixels(int, int, int, int, int[], DataBuffer)](http://docs.google.com/java/awt/image/SampleModel.html#getPixels(int,%20int,%20int,%20int,%20int%5B%5D,%20java.awt.image.DataBuffer))

### setPixels

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

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

**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.fArray - The input samples in a float array.data - The DataBuffer containing the image data. **Throws:** [NullPointerException](http://docs.google.com/java/lang/NullPointerException.html) - if fArray or data is null. [ArrayIndexOutOfBoundsException](http://docs.google.com/java/lang/ArrayIndexOutOfBoundsException.html) - if the coordinates are not in bounds, or if fArray is too small to hold the input.**See Also:**[getPixels(int, int, int, int, float[], DataBuffer)](http://docs.google.com/java/awt/image/SampleModel.html#getPixels(int,%20int,%20int,%20int,%20float%5B%5D,%20java.awt.image.DataBuffer))

### setPixels

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

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

**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.dArray - The input samples in a double array.data - The DataBuffer containing the image data. **Throws:** [NullPointerException](http://docs.google.com/java/lang/NullPointerException.html) - if dArray or data is null. [ArrayIndexOutOfBoundsException](http://docs.google.com/java/lang/ArrayIndexOutOfBoundsException.html) - if the coordinates are not in bounds, or if dArray is too small to hold the input.**See Also:**[getPixels(int, int, int, int, double[], DataBuffer)](http://docs.google.com/java/awt/image/SampleModel.html#getPixels(int,%20int,%20int,%20int,%20double%5B%5D,%20java.awt.image.DataBuffer))

### setSample

public abstract 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.

**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. **Throws:** [NullPointerException](http://docs.google.com/java/lang/NullPointerException.html) - if data is null. [ArrayIndexOutOfBoundsException](http://docs.google.com/java/lang/ArrayIndexOutOfBoundsException.html) - if the coordinates or the band index are not in bounds.**See Also:**[getSample(int, int, int, DataBuffer)](http://docs.google.com/java/awt/image/SampleModel.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. The default implementation of this method casts the input float sample to an int and then calls the setSample(int, int, int, DataBuffer) method using that int value. ArrayIndexOutOfBoundsException may be thrown if the coordinates are not in bounds.

**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 a float.data - The DataBuffer containing the image data. **Throws:** [NullPointerException](http://docs.google.com/java/lang/NullPointerException.html) - if data is null. [ArrayIndexOutOfBoundsException](http://docs.google.com/java/lang/ArrayIndexOutOfBoundsException.html) - if the coordinates or the band index are not in bounds.**See Also:**[getSample(int, int, int, DataBuffer)](http://docs.google.com/java/awt/image/SampleModel.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. The default implementation of this method casts the input double sample to an int and then calls the setSample(int, int, int, DataBuffer) method using that int value. ArrayIndexOutOfBoundsException may be thrown if the coordinates are not in bounds.

**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 a double.data - The DataBuffer containing the image data. **Throws:** [NullPointerException](http://docs.google.com/java/lang/NullPointerException.html) - if data is null. [ArrayIndexOutOfBoundsException](http://docs.google.com/java/lang/ArrayIndexOutOfBoundsException.html) - if the coordinates or the band index are not in bounds.**See Also:**[getSample(int, int, int, DataBuffer)](http://docs.google.com/java/awt/image/SampleModel.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.

**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. **Throws:** [NullPointerException](http://docs.google.com/java/lang/NullPointerException.html) - if iArray or data is null. [ArrayIndexOutOfBoundsException](http://docs.google.com/java/lang/ArrayIndexOutOfBoundsException.html) - if the coordinates or the band index are not in bounds, or if iArray is too small to hold the input.**See Also:**[getSamples(int, int, int, int, int, int[], DataBuffer)](http://docs.google.com/java/awt/image/SampleModel.html#getSamples(int,%20int,%20int,%20int,%20int,%20int%5B%5D,%20java.awt.image.DataBuffer))

### setSamples

public void **setSamples**(int x,  
 int y,  
 int w,  
 int h,  
 int b,  
 float[] fArray,  
 [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 a float array containing one sample per array element. ArrayIndexOutOfBoundsException may be thrown if the coordinates are not in bounds.

**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.fArray - The input samples in a float array.data - The DataBuffer containing the image data. **Throws:** [NullPointerException](http://docs.google.com/java/lang/NullPointerException.html) - if fArray or data is null. [ArrayIndexOutOfBoundsException](http://docs.google.com/java/lang/ArrayIndexOutOfBoundsException.html) - if the coordinates or the band index are not in bounds, or if fArray is too small to hold the input.**See Also:**[getSamples(int, int, int, int, int, float[], DataBuffer)](http://docs.google.com/java/awt/image/SampleModel.html#getSamples(int,%20int,%20int,%20int,%20int,%20float%5B%5D,%20java.awt.image.DataBuffer))

### setSamples

public void **setSamples**(int x,  
 int y,  
 int w,  
 int h,  
 int b,  
 double[] dArray,  
 [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 a double array containing one sample per array element. ArrayIndexOutOfBoundsException may be thrown if the coordinates are not in bounds.

**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.dArray - The input samples in a double array.data - The DataBuffer containing the image data. **Throws:** [NullPointerException](http://docs.google.com/java/lang/NullPointerException.html) - if dArray or data is null. [ArrayIndexOutOfBoundsException](http://docs.google.com/java/lang/ArrayIndexOutOfBoundsException.html) - if the coordinates or the band index are not in bounds, or if dArray is too small to hold the input.**See Also:**[getSamples(int, int, int, int, int, double[], DataBuffer)](http://docs.google.com/java/awt/image/SampleModel.html#getSamples(int,%20int,%20int,%20int,%20int,%20double%5B%5D,%20java.awt.image.DataBuffer))

### createCompatibleSampleModel

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

Creates a SampleModel which describes data in this SampleModel's format, but with a different width and height.

**Parameters:**w - the width of the image datah - the height of the image data **Returns:**a SampleModel describing the same image data as this SampleModel, but with a different size.

### createSubsetSampleModel

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

Creates a new SampleModel with a subset of the bands of this SampleModel.

**Parameters:**bands - the subset of bands of this SampleModel **Returns:**a SampleModel with a subset of bands of this SampleModel.

### createDataBuffer

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

Creates a DataBuffer that corresponds to this SampleModel. The DataBuffer's width and height will match this SampleModel's.

**Returns:**a DataBuffer corresponding to this SampleModel.

### getSampleSize

public abstract int[] **getSampleSize**()

Returns the size in bits of samples for all bands.

**Returns:**the size of samples for all bands.

### getSampleSize

public abstract int **getSampleSize**(int band)

Returns the size in bits of samples for the specified band.

**Parameters:**band - the specified band **Returns:**the size of the samples of the specified band.

| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/SampleModel.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/RGBImageFilter.html)   [**NEXT CLASS**](http://docs.google.com/java/awt/image/ShortLookupTable.html) | [**FRAMES**](http://docs.google.com/index.html?java/awt/image/SampleModel.html)    [**NO FRAMES**](http://docs.google.com/SampleModel.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | [FIELD](#3znysh7) | [CONSTR](#2et92p0) | [METHOD](#tyjcwt) | DETAIL: [FIELD](#1t3h5sf) | [CONSTR](#26in1rg) | [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).