| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/Raster.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/PixelInterleavedSampleModel.html)   [**NEXT CLASS**](http://docs.google.com/java/awt/image/RasterFormatException.html) | [**FRAMES**](http://docs.google.com/index.html?java/awt/image/Raster.html)    [**NO FRAMES**](http://docs.google.com/Raster.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | [FIELD](#3znysh7) | [CONSTR](#2et92p0) | [METHOD](#tyjcwt) | DETAIL: [FIELD](#1t3h5sf) | [CONSTR](#3j2qqm3) | [METHOD](#1ci93xb) |

## **java.awt.image**

Class Raster

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

**Direct Known Subclasses:** [WritableRaster](http://docs.google.com/java/awt/image/WritableRaster.html)

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

A class representing a rectangular array of pixels. A Raster encapsulates a DataBuffer that stores the sample values and a SampleModel that describes how to locate a given sample value in a DataBuffer.

A Raster defines values for pixels occupying a particular rectangular area of the plane, not necessarily including (0, 0). The rectangle, known as the Raster's bounding rectangle and available by means of the getBounds method, is defined by minX, minY, width, and height values. The minX and minY values define the coordinate of the upper left corner of the Raster. References to pixels outside of the bounding rectangle may result in an exception being thrown, or may result in references to unintended elements of the Raster's associated DataBuffer. It is the user's responsibility to avoid accessing such pixels.

A SampleModel describes how samples of a Raster are stored in the primitive array elements of a DataBuffer. Samples may be stored one per data element, as in a PixelInterleavedSampleModel or BandedSampleModel, or packed several to an element, as in a SinglePixelPackedSampleModel or MultiPixelPackedSampleModel. The SampleModel is also controls whether samples are sign extended, allowing unsigned data to be stored in signed Java data types such as byte, short, and int.

Although a Raster may live anywhere in the plane, a SampleModel makes use of a simple coordinate system that starts at (0, 0). A Raster therefore contains a translation factor that allows pixel locations to be mapped between the Raster's coordinate system and that of the SampleModel. The translation from the SampleModel coordinate system to that of the Raster may be obtained by the getSampleModelTranslateX and getSampleModelTranslateY methods.

A Raster may share a DataBuffer with another Raster either by explicit construction or by the use of the createChild and createTranslatedChild methods. Rasters created by these methods can return a reference to the Raster they were created from by means of the getParent method. For a Raster that was not constructed by means of a call to createTranslatedChild or createChild, getParent will return null.

The createTranslatedChild method returns a new Raster that shares all of the data of the current Raster, but occupies a bounding rectangle of the same width and height but with a different starting point. For example, if the parent Raster occupied the region (10, 10) to (100, 100), and the translated Raster was defined to start at (50, 50), then pixel (20, 20) of the parent and pixel (60, 60) of the child occupy the same location in the DataBuffer shared by the two Rasters. In the first case, (-10, -10) should be added to a pixel coordinate to obtain the corresponding SampleModel coordinate, and in the second case (-50, -50) should be added.

The translation between a parent and child Raster may be determined by subtracting the child's sampleModelTranslateX and sampleModelTranslateY values from those of the parent.

The createChild method may be used to create a new Raster occupying only a subset of its parent's bounding rectangle (with the same or a translated coordinate system) or with a subset of the bands of its parent.

All constructors are protected. The correct way to create a Raster is to use one of the static create methods defined in this class. These methods create instances of Raster that use the standard Interleaved, Banded, and Packed SampleModels and that may be processed more efficiently than a Raster created by combining an externally generated SampleModel and DataBuffer.

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

| **Field Summary** | |
| --- | --- |
| protected  [DataBuffer](http://docs.google.com/java/awt/image/DataBuffer.html) | [**dataBuffer**](http://docs.google.com/java/awt/image/Raster.html#dataBuffer)            The DataBuffer that stores the image data. |
| protected  int | [**height**](http://docs.google.com/java/awt/image/Raster.html#height)            The height of this Raster. |
| protected  int | [**minX**](http://docs.google.com/java/awt/image/Raster.html#minX)            The X coordinate of the upper-left pixel of this Raster. |
| protected  int | [**minY**](http://docs.google.com/java/awt/image/Raster.html#minY)            The Y coordinate of the upper-left pixel of this Raster. |
| protected  int | [**numBands**](http://docs.google.com/java/awt/image/Raster.html#numBands)            The number of bands in the Raster. |
| protected  int | [**numDataElements**](http://docs.google.com/java/awt/image/Raster.html#numDataElements)            The number of DataBuffer data elements per pixel. |
| protected  [Raster](http://docs.google.com/java/awt/image/Raster.html) | [**parent**](http://docs.google.com/java/awt/image/Raster.html#parent)            The parent of this Raster, or null. |
| protected  [SampleModel](http://docs.google.com/java/awt/image/SampleModel.html) | [**sampleModel**](http://docs.google.com/java/awt/image/Raster.html#sampleModel)            The SampleModel that describes how pixels from this Raster are stored in the DataBuffer. |
| protected  int | [**sampleModelTranslateX**](http://docs.google.com/java/awt/image/Raster.html#sampleModelTranslateX)            The X translation from the coordinate space of the Raster's SampleModel to that of the Raster. |
| protected  int | [**sampleModelTranslateY**](http://docs.google.com/java/awt/image/Raster.html#sampleModelTranslateY)            The Y translation from the coordinate space of the Raster's SampleModel to that of the Raster. |
| protected  int | [**width**](http://docs.google.com/java/awt/image/Raster.html#width)            The width of this Raster. |

| **Constructor Summary** | |
| --- | --- |
| protected | [**Raster**](http://docs.google.com/java/awt/image/Raster.html#Raster(java.awt.image.SampleModel,%20java.awt.image.DataBuffer,%20java.awt.Point))([SampleModel](http://docs.google.com/java/awt/image/SampleModel.html) sampleModel, [DataBuffer](http://docs.google.com/java/awt/image/DataBuffer.html) dataBuffer, [Point](http://docs.google.com/java/awt/Point.html) origin)            Constructs a Raster with the given SampleModel and DataBuffer. |
| protected | [**Raster**](http://docs.google.com/java/awt/image/Raster.html#Raster(java.awt.image.SampleModel,%20java.awt.image.DataBuffer,%20java.awt.Rectangle,%20java.awt.Point,%20java.awt.image.Raster))([SampleModel](http://docs.google.com/java/awt/image/SampleModel.html) sampleModel, [DataBuffer](http://docs.google.com/java/awt/image/DataBuffer.html) dataBuffer, [Rectangle](http://docs.google.com/java/awt/Rectangle.html) aRegion, [Point](http://docs.google.com/java/awt/Point.html) sampleModelTranslate, [Raster](http://docs.google.com/java/awt/image/Raster.html) parent)            Constructs a Raster with the given SampleModel, DataBuffer, and parent. |
| protected | [**Raster**](http://docs.google.com/java/awt/image/Raster.html#Raster(java.awt.image.SampleModel,%20java.awt.Point))([SampleModel](http://docs.google.com/java/awt/image/SampleModel.html) sampleModel, [Point](http://docs.google.com/java/awt/Point.html) origin)            Constructs a Raster with the given SampleModel. |

| **Method Summary** | |
| --- | --- |
| static [WritableRaster](http://docs.google.com/java/awt/image/WritableRaster.html) | [**createBandedRaster**](http://docs.google.com/java/awt/image/Raster.html#createBandedRaster(java.awt.image.DataBuffer,%20int,%20int,%20int,%20int%5B%5D,%20int%5B%5D,%20java.awt.Point))([DataBuffer](http://docs.google.com/java/awt/image/DataBuffer.html) dataBuffer, int w, int h, int scanlineStride, int[] bankIndices, int[] bandOffsets, [Point](http://docs.google.com/java/awt/Point.html) location)            Creates a Raster based on a BandedSampleModel with the specified DataBuffer, width, height, scanline stride, bank indices, and band offsets. |
| static [WritableRaster](http://docs.google.com/java/awt/image/WritableRaster.html) | [**createBandedRaster**](http://docs.google.com/java/awt/image/Raster.html#createBandedRaster(int,%20int,%20int,%20int,%20int%5B%5D,%20int%5B%5D,%20java.awt.Point))(int dataType, int w, int h, int scanlineStride, int[] bankIndices, int[] bandOffsets, [Point](http://docs.google.com/java/awt/Point.html) location)            Creates a Raster based on a BandedSampleModel with the specified data type, width, height, scanline stride, bank indices and band offsets. |
| static [WritableRaster](http://docs.google.com/java/awt/image/WritableRaster.html) | [**createBandedRaster**](http://docs.google.com/java/awt/image/Raster.html#createBandedRaster(int,%20int,%20int,%20int,%20java.awt.Point))(int dataType, int w, int h, int bands, [Point](http://docs.google.com/java/awt/Point.html) location)            Creates a Raster based on a BandedSampleModel with the specified data type, width, height, and number of bands. |
| [Raster](http://docs.google.com/java/awt/image/Raster.html) | [**createChild**](http://docs.google.com/java/awt/image/Raster.html#createChild(int,%20int,%20int,%20int,%20int,%20int,%20int%5B%5D))(int parentX, int parentY, int width, int height, int childMinX, int childMinY, int[] bandList)            Returns a new Raster which shares all or part of this Raster's DataBuffer. |
| [WritableRaster](http://docs.google.com/java/awt/image/WritableRaster.html) | [**createCompatibleWritableRaster**](http://docs.google.com/java/awt/image/Raster.html#createCompatibleWritableRaster())()            Create a compatible WritableRaster the same size as this Raster with the same SampleModel and a new initialized DataBuffer. |
| [WritableRaster](http://docs.google.com/java/awt/image/WritableRaster.html) | [**createCompatibleWritableRaster**](http://docs.google.com/java/awt/image/Raster.html#createCompatibleWritableRaster(int,%20int))(int w, int h)            Create a compatible WritableRaster with the specified size, a new SampleModel, and a new initialized DataBuffer. |
| [WritableRaster](http://docs.google.com/java/awt/image/WritableRaster.html) | [**createCompatibleWritableRaster**](http://docs.google.com/java/awt/image/Raster.html#createCompatibleWritableRaster(int,%20int,%20int,%20int))(int x, int y, int w, int h)            Create a compatible WritableRaster with the specified location (minX, minY) and size (width, height), a new SampleModel, and a new initialized DataBuffer. |
| [WritableRaster](http://docs.google.com/java/awt/image/WritableRaster.html) | [**createCompatibleWritableRaster**](http://docs.google.com/java/awt/image/Raster.html#createCompatibleWritableRaster(java.awt.Rectangle))([Rectangle](http://docs.google.com/java/awt/Rectangle.html) rect)            Create a compatible WritableRaster with location (minX, minY) and size (width, height) specified by rect, a new SampleModel, and a new initialized DataBuffer. |
| static [WritableRaster](http://docs.google.com/java/awt/image/WritableRaster.html) | [**createInterleavedRaster**](http://docs.google.com/java/awt/image/Raster.html#createInterleavedRaster(java.awt.image.DataBuffer,%20int,%20int,%20int,%20int,%20int%5B%5D,%20java.awt.Point))([DataBuffer](http://docs.google.com/java/awt/image/DataBuffer.html) dataBuffer, int w, int h, int scanlineStride, int pixelStride, int[] bandOffsets, [Point](http://docs.google.com/java/awt/Point.html) location)            Creates a Raster based on a PixelInterleavedSampleModel with the specified DataBuffer, width, height, scanline stride, pixel stride, and band offsets. |
| static [WritableRaster](http://docs.google.com/java/awt/image/WritableRaster.html) | [**createInterleavedRaster**](http://docs.google.com/java/awt/image/Raster.html#createInterleavedRaster(int,%20int,%20int,%20int,%20int,%20int%5B%5D,%20java.awt.Point))(int dataType, int w, int h, int scanlineStride, int pixelStride, int[] bandOffsets, [Point](http://docs.google.com/java/awt/Point.html) location)            Creates a Raster based on a PixelInterleavedSampleModel with the specified data type, width, height, scanline stride, pixel stride, and band offsets. |
| static [WritableRaster](http://docs.google.com/java/awt/image/WritableRaster.html) | [**createInterleavedRaster**](http://docs.google.com/java/awt/image/Raster.html#createInterleavedRaster(int,%20int,%20int,%20int,%20java.awt.Point))(int dataType, int w, int h, int bands, [Point](http://docs.google.com/java/awt/Point.html) location)            Creates a Raster based on a PixelInterleavedSampleModel with the specified data type, width, height, and number of bands. |
| static [WritableRaster](http://docs.google.com/java/awt/image/WritableRaster.html) | [**createPackedRaster**](http://docs.google.com/java/awt/image/Raster.html#createPackedRaster(java.awt.image.DataBuffer,%20int,%20int,%20int,%20int%5B%5D,%20java.awt.Point))([DataBuffer](http://docs.google.com/java/awt/image/DataBuffer.html) dataBuffer, int w, int h, int scanlineStride, int[] bandMasks, [Point](http://docs.google.com/java/awt/Point.html) location)            Creates a Raster based on a SinglePixelPackedSampleModel with the specified DataBuffer, width, height, scanline stride, and band masks. |
| static [WritableRaster](http://docs.google.com/java/awt/image/WritableRaster.html) | [**createPackedRaster**](http://docs.google.com/java/awt/image/Raster.html#createPackedRaster(java.awt.image.DataBuffer,%20int,%20int,%20int,%20java.awt.Point))([DataBuffer](http://docs.google.com/java/awt/image/DataBuffer.html) dataBuffer, int w, int h, int bitsPerPixel, [Point](http://docs.google.com/java/awt/Point.html) location)            Creates a Raster based on a MultiPixelPackedSampleModel with the specified DataBuffer, width, height, and bits per pixel. |
| static [WritableRaster](http://docs.google.com/java/awt/image/WritableRaster.html) | [**createPackedRaster**](http://docs.google.com/java/awt/image/Raster.html#createPackedRaster(int,%20int,%20int,%20int%5B%5D,%20java.awt.Point))(int dataType, int w, int h, int[] bandMasks, [Point](http://docs.google.com/java/awt/Point.html) location)            Creates a Raster based on a SinglePixelPackedSampleModel with the specified data type, width, height, and band masks. |
| static [WritableRaster](http://docs.google.com/java/awt/image/WritableRaster.html) | [**createPackedRaster**](http://docs.google.com/java/awt/image/Raster.html#createPackedRaster(int,%20int,%20int,%20int,%20int,%20java.awt.Point))(int dataType, int w, int h, int bands, int bitsPerBand, [Point](http://docs.google.com/java/awt/Point.html) location)            Creates a Raster based on a packed SampleModel with the specified data type, width, height, number of bands, and bits per band. |
| static [Raster](http://docs.google.com/java/awt/image/Raster.html) | [**createRaster**](http://docs.google.com/java/awt/image/Raster.html#createRaster(java.awt.image.SampleModel,%20java.awt.image.DataBuffer,%20java.awt.Point))([SampleModel](http://docs.google.com/java/awt/image/SampleModel.html) sm, [DataBuffer](http://docs.google.com/java/awt/image/DataBuffer.html) db, [Point](http://docs.google.com/java/awt/Point.html) location)            Creates a Raster with the specified SampleModel and DataBuffer. |
| [Raster](http://docs.google.com/java/awt/image/Raster.html) | [**createTranslatedChild**](http://docs.google.com/java/awt/image/Raster.html#createTranslatedChild(int,%20int))(int childMinX, int childMinY)            Create a Raster with the same size, SampleModel and DataBuffer as this one, but with a different location. |
| static [WritableRaster](http://docs.google.com/java/awt/image/WritableRaster.html) | [**createWritableRaster**](http://docs.google.com/java/awt/image/Raster.html#createWritableRaster(java.awt.image.SampleModel,%20java.awt.image.DataBuffer,%20java.awt.Point))([SampleModel](http://docs.google.com/java/awt/image/SampleModel.html) sm, [DataBuffer](http://docs.google.com/java/awt/image/DataBuffer.html) db, [Point](http://docs.google.com/java/awt/Point.html) location)            Creates a WritableRaster with the specified SampleModel and DataBuffer. |
| static [WritableRaster](http://docs.google.com/java/awt/image/WritableRaster.html) | [**createWritableRaster**](http://docs.google.com/java/awt/image/Raster.html#createWritableRaster(java.awt.image.SampleModel,%20java.awt.Point))([SampleModel](http://docs.google.com/java/awt/image/SampleModel.html) sm, [Point](http://docs.google.com/java/awt/Point.html) location)            Creates a WritableRaster with the specified SampleModel. |
| [Rectangle](http://docs.google.com/java/awt/Rectangle.html) | [**getBounds**](http://docs.google.com/java/awt/image/Raster.html#getBounds())()            Returns the bounding Rectangle of this Raster. |
| [DataBuffer](http://docs.google.com/java/awt/image/DataBuffer.html) | [**getDataBuffer**](http://docs.google.com/java/awt/image/Raster.html#getDataBuffer())()            Returns the DataBuffer associated with this Raster. |
| [Object](http://docs.google.com/java/lang/Object.html) | [**getDataElements**](http://docs.google.com/java/awt/image/Raster.html#getDataElements(int,%20int,%20int,%20int,%20java.lang.Object))(int x, int y, int w, int h, [Object](http://docs.google.com/java/lang/Object.html) outData)            Returns the pixel data for the specified rectangle of pixels in a primitive array of type TransferType. |
| [Object](http://docs.google.com/java/lang/Object.html) | [**getDataElements**](http://docs.google.com/java/awt/image/Raster.html#getDataElements(int,%20int,%20java.lang.Object))(int x, int y, [Object](http://docs.google.com/java/lang/Object.html) outData)            Returns data for a single pixel in a primitive array of type TransferType. |
| int | [**getHeight**](http://docs.google.com/java/awt/image/Raster.html#getHeight())()            Returns the height in pixels of the Raster. |
| int | [**getMinX**](http://docs.google.com/java/awt/image/Raster.html#getMinX())()            Returns the minimum valid X coordinate of the Raster. |
| int | [**getMinY**](http://docs.google.com/java/awt/image/Raster.html#getMinY())()            Returns the minimum valid Y coordinate of the Raster. |
| int | [**getNumBands**](http://docs.google.com/java/awt/image/Raster.html#getNumBands())()            Returns the number of bands (samples per pixel) in this Raster. |
| int | [**getNumDataElements**](http://docs.google.com/java/awt/image/Raster.html#getNumDataElements())()            Returns the number of data elements needed to transfer one pixel via the getDataElements and setDataElements methods. |
| [Raster](http://docs.google.com/java/awt/image/Raster.html) | [**getParent**](http://docs.google.com/java/awt/image/Raster.html#getParent())()            Returns the parent Raster (if any) of this Raster or null. |
| double[] | [**getPixel**](http://docs.google.com/java/awt/image/Raster.html#getPixel(int,%20int,%20double%5B%5D))(int x, int y, double[] dArray)            Returns the samples in an array of double for the specified pixel. |
| float[] | [**getPixel**](http://docs.google.com/java/awt/image/Raster.html#getPixel(int,%20int,%20float%5B%5D))(int x, int y, float[] fArray)            Returns the samples in an array of float for the specified pixel. |
| int[] | [**getPixel**](http://docs.google.com/java/awt/image/Raster.html#getPixel(int,%20int,%20int%5B%5D))(int x, int y, int[] iArray)            Returns the samples in an array of int for the specified pixel. |
| double[] | [**getPixels**](http://docs.google.com/java/awt/image/Raster.html#getPixels(int,%20int,%20int,%20int,%20double%5B%5D))(int x, int y, int w, int h, double[] dArray)            Returns a double array containing all samples for a rectangle of pixels, one sample per array element. |
| float[] | [**getPixels**](http://docs.google.com/java/awt/image/Raster.html#getPixels(int,%20int,%20int,%20int,%20float%5B%5D))(int x, int y, int w, int h, float[] fArray)            Returns a float array containing all samples for a rectangle of pixels, one sample per array element. |
| int[] | [**getPixels**](http://docs.google.com/java/awt/image/Raster.html#getPixels(int,%20int,%20int,%20int,%20int%5B%5D))(int x, int y, int w, int h, int[] iArray)            Returns an int array containing all samples for a rectangle of pixels, one sample per array element. |
| int | [**getSample**](http://docs.google.com/java/awt/image/Raster.html#getSample(int,%20int,%20int))(int x, int y, int b)            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/Raster.html#getSampleDouble(int,%20int,%20int))(int x, int y, int b)            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/Raster.html#getSampleFloat(int,%20int,%20int))(int x, int y, int b)            Returns the sample in a specified band for the pixel located at (x,y) as a float. |
| [SampleModel](http://docs.google.com/java/awt/image/SampleModel.html) | [**getSampleModel**](http://docs.google.com/java/awt/image/Raster.html#getSampleModel())()            Returns the SampleModel that describes the layout of the image data. |
| int | [**getSampleModelTranslateX**](http://docs.google.com/java/awt/image/Raster.html#getSampleModelTranslateX())()            Returns the X translation from the coordinate system of the SampleModel to that of the Raster. |
| int | [**getSampleModelTranslateY**](http://docs.google.com/java/awt/image/Raster.html#getSampleModelTranslateY())()            Returns the Y translation from the coordinate system of the SampleModel to that of the Raster. |
| double[] | [**getSamples**](http://docs.google.com/java/awt/image/Raster.html#getSamples(int,%20int,%20int,%20int,%20int,%20double%5B%5D))(int x, int y, int w, int h, int b, double[] dArray)            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/Raster.html#getSamples(int,%20int,%20int,%20int,%20int,%20float%5B%5D))(int x, int y, int w, int h, int b, float[] fArray)            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/Raster.html#getSamples(int,%20int,%20int,%20int,%20int,%20int%5B%5D))(int x, int y, int w, int h, int b, int[] iArray)            Returns the samples for a specified band for the specified rectangle of pixels in an int array, one sample per array element. |
| int | [**getTransferType**](http://docs.google.com/java/awt/image/Raster.html#getTransferType())()            Returns the TransferType used to transfer pixels via the getDataElements and setDataElements methods. |
| int | [**getWidth**](http://docs.google.com/java/awt/image/Raster.html#getWidth())()            Returns the width in pixels of the Raster. |

| **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** |
| --- |

### sampleModel

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

The SampleModel that describes how pixels from this Raster are stored in the DataBuffer.

### dataBuffer

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

The DataBuffer that stores the image data.

### minX

protected int **minX**

The X coordinate of the upper-left pixel of this Raster.

### minY

protected int **minY**

The Y coordinate of the upper-left pixel of this Raster.

### width

protected int **width**

The width of this Raster.

### height

protected int **height**

The height of this Raster.

### sampleModelTranslateX

protected int **sampleModelTranslateX**

The X translation from the coordinate space of the Raster's SampleModel to that of the Raster.

### sampleModelTranslateY

protected int **sampleModelTranslateY**

The Y translation from the coordinate space of the Raster's SampleModel to that of the Raster.

### numBands

protected int **numBands**

The number of bands in the Raster.

### numDataElements

protected int **numDataElements**

The number of DataBuffer data elements per pixel.

### parent

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

The parent of this Raster, or null.

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

### Raster

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

Constructs a Raster with the given SampleModel. The Raster's upper left corner is origin and it is the same size as the SampleModel. A DataBuffer large enough to describe the Raster is automatically created.

**Parameters:**sampleModel - The SampleModel that specifies the layoutorigin - The Point that specified the origin **Throws:** [RasterFormatException](http://docs.google.com/java/awt/image/RasterFormatException.html) - if computing either origin.x + sampleModel.getWidth() or origin.y + sampleModel.getHeight() results in integer overflow [NullPointerException](http://docs.google.com/java/lang/NullPointerException.html) - either sampleModel or origin is null

### Raster

protected **Raster**([SampleModel](http://docs.google.com/java/awt/image/SampleModel.html) sampleModel,  
 [DataBuffer](http://docs.google.com/java/awt/image/DataBuffer.html) dataBuffer,  
 [Point](http://docs.google.com/java/awt/Point.html) origin)

Constructs a Raster with the given SampleModel and DataBuffer. The Raster's upper left corner is origin and it is the same size as the SampleModel. The DataBuffer is not initialized and must be compatible with SampleModel.

**Parameters:**sampleModel - The SampleModel that specifies the layoutdataBuffer - The DataBuffer that contains the image dataorigin - The Point that specifies the origin **Throws:** [RasterFormatException](http://docs.google.com/java/awt/image/RasterFormatException.html) - if computing either origin.x + sampleModel.getWidth() or origin.y + sampleModel.getHeight() results in integer overflow [NullPointerException](http://docs.google.com/java/lang/NullPointerException.html) - either sampleModel or origin is null

### Raster

protected **Raster**([SampleModel](http://docs.google.com/java/awt/image/SampleModel.html) sampleModel,  
 [DataBuffer](http://docs.google.com/java/awt/image/DataBuffer.html) dataBuffer,  
 [Rectangle](http://docs.google.com/java/awt/Rectangle.html) aRegion,  
 [Point](http://docs.google.com/java/awt/Point.html) sampleModelTranslate,  
 [Raster](http://docs.google.com/java/awt/image/Raster.html) parent)

Constructs a Raster with the given SampleModel, DataBuffer, and parent. aRegion specifies the bounding rectangle of the new Raster. When translated into the base Raster's coordinate system, aRegion must be contained by the base Raster. (The base Raster is the Raster's ancestor which has no parent.) sampleModelTranslate specifies the sampleModelTranslateX and sampleModelTranslateY values of the new Raster. Note that this constructor should generally be called by other constructors or create methods, it should not be used directly.

**Parameters:**sampleModel - The SampleModel that specifies the layoutdataBuffer - The DataBuffer that contains the image dataaRegion - The Rectangle that specifies the image areasampleModelTranslate - The Point that specifies the translation from SampleModel to Raster coordinatesparent - The parent (if any) of this raster **Throws:** [NullPointerException](http://docs.google.com/java/lang/NullPointerException.html) - if any of sampleModel, dataBuffer, aRegion or sampleModelTranslate is null [RasterFormatException](http://docs.google.com/java/awt/image/RasterFormatException.html) - if aRegion has width or height less than or equal to zero, or computing either aRegion.x + aRegion.width or aRegion.y + aRegion.height results in integer overflow

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

### createInterleavedRaster

public static [WritableRaster](http://docs.google.com/java/awt/image/WritableRaster.html) **createInterleavedRaster**(int dataType,  
 int w,  
 int h,  
 int bands,  
 [Point](http://docs.google.com/java/awt/Point.html) location)

Creates a Raster based on a PixelInterleavedSampleModel with the specified data type, width, height, and number of bands.

The upper left corner of the Raster is given by the location argument. If location is null, (0, 0) will be used. The dataType parameter should be one of the enumerated values defined in the DataBuffer class.

Note that interleaved DataBuffer.TYPE\_INT Rasters are not supported. To create a 1-band Raster of type DataBuffer.TYPE\_INT, use Raster.createPackedRaster().

The only dataTypes supported currently are TYPE\_BYTE and TYPE\_USHORT.

**Parameters:**dataType - the data type for storing samplesw - the width in pixels of the image datah - the height in pixels of the image databands - the number of bandslocation - the upper-left corner of the Raster **Returns:**a WritableRaster object with the specified data type, width, height and number of bands. **Throws:** [RasterFormatException](http://docs.google.com/java/awt/image/RasterFormatException.html) - if w or h is less than or equal to zero, or computing either location.x + w or location.y + h results in integer overflow

### createInterleavedRaster

public static [WritableRaster](http://docs.google.com/java/awt/image/WritableRaster.html) **createInterleavedRaster**(int dataType,  
 int w,  
 int h,  
 int scanlineStride,  
 int pixelStride,  
 int[] bandOffsets,  
 [Point](http://docs.google.com/java/awt/Point.html) location)

Creates a Raster based on a PixelInterleavedSampleModel with the specified data type, width, height, scanline stride, pixel stride, and band offsets. The number of bands is inferred from bandOffsets.length.

The upper left corner of the Raster is given by the location argument. If location is null, (0, 0) will be used. The dataType parameter should be one of the enumerated values defined in the DataBuffer class.

Note that interleaved DataBuffer.TYPE\_INT Rasters are not supported. To create a 1-band Raster of type DataBuffer.TYPE\_INT, use Raster.createPackedRaster().

The only dataTypes supported currently are TYPE\_BYTE and TYPE\_USHORT.

**Parameters:**dataType - the data type for storing samplesw - the width in pixels of the image datah - the height in pixels of the image datascanlineStride - the line stride of the image datapixelStride - the pixel stride of the image databandOffsets - the offsets of all bandslocation - the upper-left corner of the Raster **Returns:**a WritableRaster object with the specified data type, width, height, scanline stride, pixel stride and band offsets. **Throws:** [RasterFormatException](http://docs.google.com/java/awt/image/RasterFormatException.html) - if w or h is less than or equal to zero, or computing either location.x + w or location.y + h results in integer overflow [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if dataType is not one of the supported data types, which are DataBuffer.TYPE\_BYTE, or DataBuffer.TYPE\_USHORT.

### createBandedRaster

public static [WritableRaster](http://docs.google.com/java/awt/image/WritableRaster.html) **createBandedRaster**(int dataType,  
 int w,  
 int h,  
 int bands,  
 [Point](http://docs.google.com/java/awt/Point.html) location)

Creates a Raster based on a BandedSampleModel with the specified data type, width, height, and number of bands.

The upper left corner of the Raster is given by the location argument. If location is null, (0, 0) will be used. The dataType parameter should be one of the enumerated values defined in the DataBuffer class.

The only dataTypes supported currently are TYPE\_BYTE, TYPE\_USHORT, and TYPE\_INT.

**Parameters:**dataType - the data type for storing samplesw - the width in pixels of the image datah - the height in pixels of the image databands - the number of bandslocation - the upper-left corner of the Raster **Returns:**a WritableRaster object with the specified data type, width, height and number of bands. **Throws:** [RasterFormatException](http://docs.google.com/java/awt/image/RasterFormatException.html) - if w or h is less than or equal to zero, or computing either location.x + w or location.y + h results in integer overflow [ArrayIndexOutOfBoundsException](http://docs.google.com/java/lang/ArrayIndexOutOfBoundsException.html) - if bands is less than 1

### createBandedRaster

public static [WritableRaster](http://docs.google.com/java/awt/image/WritableRaster.html) **createBandedRaster**(int dataType,  
 int w,  
 int h,  
 int scanlineStride,  
 int[] bankIndices,  
 int[] bandOffsets,  
 [Point](http://docs.google.com/java/awt/Point.html) location)

Creates a Raster based on a BandedSampleModel with the specified data type, width, height, scanline stride, bank indices and band offsets. The number of bands is inferred from bankIndices.length and bandOffsets.length, which must be the same.

The upper left corner of the Raster is given by the location argument. The dataType parameter should be one of the enumerated values defined in the DataBuffer class.

The only dataTypes supported currently are TYPE\_BYTE, TYPE\_USHORT, and TYPE\_INT.

**Parameters:**dataType - the data type for storing samplesw - the width in pixels of the image datah - the height in pixels of the image datascanlineStride - the line stride of the image databankIndices - the bank indices for each bandbandOffsets - the offsets of all bandslocation - the upper-left corner of the Raster **Returns:**a WritableRaster object with the specified data type, width, height, scanline stride, bank indices and band offsets. **Throws:** [RasterFormatException](http://docs.google.com/java/awt/image/RasterFormatException.html) - if w or h is less than or equal to zero, or computing either location.x + w or location.y + h results in integer overflow [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if dataType is not one of the supported data types, which are DataBuffer.TYPE\_BYTE, DataBuffer.TYPE\_USHORT or DataBuffer.TYPE\_INT [ArrayIndexOutOfBoundsException](http://docs.google.com/java/lang/ArrayIndexOutOfBoundsException.html) - if bankIndices or bandOffsets is null

### createPackedRaster

public static [WritableRaster](http://docs.google.com/java/awt/image/WritableRaster.html) **createPackedRaster**(int dataType,  
 int w,  
 int h,  
 int[] bandMasks,  
 [Point](http://docs.google.com/java/awt/Point.html) location)

Creates a Raster based on a SinglePixelPackedSampleModel with the specified data type, width, height, and band masks. The number of bands is inferred from bandMasks.length.

The upper left corner of the Raster is given by the location argument. If location is null, (0, 0) will be used. The dataType parameter should be one of the enumerated values defined in the DataBuffer class.

The only dataTypes supported currently are TYPE\_BYTE, TYPE\_USHORT, and TYPE\_INT.

**Parameters:**dataType - the data type for storing samplesw - the width in pixels of the image datah - the height in pixels of the image databandMasks - an array containing an entry for each bandlocation - the upper-left corner of the Raster **Returns:**a WritableRaster object with the specified data type, width, height, and band masks. **Throws:** [RasterFormatException](http://docs.google.com/java/awt/image/RasterFormatException.html) - if w or h is less than or equal to zero, or computing either location.x + w or location.y + h results in integer overflow [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if dataType is not one of the supported data types, which are DataBuffer.TYPE\_BYTE, DataBuffer.TYPE\_USHORT or DataBuffer.TYPE\_INT

### createPackedRaster

public static [WritableRaster](http://docs.google.com/java/awt/image/WritableRaster.html) **createPackedRaster**(int dataType,  
 int w,  
 int h,  
 int bands,  
 int bitsPerBand,  
 [Point](http://docs.google.com/java/awt/Point.html) location)

Creates a Raster based on a packed SampleModel with the specified data type, width, height, number of bands, and bits per band. If the number of bands is one, the SampleModel will be a MultiPixelPackedSampleModel.

If the number of bands is more than one, the SampleModel will be a SinglePixelPackedSampleModel, with each band having bitsPerBand bits. In either case, the requirements on dataType and bitsPerBand imposed by the corresponding SampleModel must be met.

The upper left corner of the Raster is given by the location argument. If location is null, (0, 0) will be used. The dataType parameter should be one of the enumerated values defined in the DataBuffer class.

The only dataTypes supported currently are TYPE\_BYTE, TYPE\_USHORT, and TYPE\_INT.

**Parameters:**dataType - the data type for storing samplesw - the width in pixels of the image datah - the height in pixels of the image databands - the number of bandsbitsPerBand - the number of bits per bandlocation - the upper-left corner of the Raster **Returns:**a WritableRaster object with the specified data type, width, height, number of bands, and bits per band. **Throws:** [RasterFormatException](http://docs.google.com/java/awt/image/RasterFormatException.html) - if w or h is less than or equal to zero, or computing either location.x + w or location.y + h results in integer overflow [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if the product of bitsPerBand and bands is greater than the number of bits held by dataType [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if bitsPerBand or bands is not greater than zero [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if dataType is not one of the supported data types, which are DataBuffer.TYPE\_BYTE, DataBuffer.TYPE\_USHORT or DataBuffer.TYPE\_INT

### createInterleavedRaster

public static [WritableRaster](http://docs.google.com/java/awt/image/WritableRaster.html) **createInterleavedRaster**([DataBuffer](http://docs.google.com/java/awt/image/DataBuffer.html) dataBuffer,  
 int w,  
 int h,  
 int scanlineStride,  
 int pixelStride,  
 int[] bandOffsets,  
 [Point](http://docs.google.com/java/awt/Point.html) location)

Creates a Raster based on a PixelInterleavedSampleModel with the specified DataBuffer, width, height, scanline stride, pixel stride, and band offsets. The number of bands is inferred from bandOffsets.length. The upper left corner of the Raster is given by the location argument. If location is null, (0, 0) will be used.

Note that interleaved DataBuffer.TYPE\_INT Rasters are not supported. To create a 1-band Raster of type DataBuffer.TYPE\_INT, use Raster.createPackedRaster().

**Parameters:**dataBuffer - the DataBuffer that contains the image dataw - the width in pixels of the image datah - the height in pixels of the image datascanlineStride - the line stride of the image datapixelStride - the pixel stride of the image databandOffsets - the offsets of all bandslocation - the upper-left corner of the Raster **Returns:**a WritableRaster object with the specified DataBuffer, width, height, scanline stride, pixel stride and band offsets. **Throws:** [RasterFormatException](http://docs.google.com/java/awt/image/RasterFormatException.html) - if w or h is less than or equal to zero, or computing either location.x + w or location.y + h results in integer overflow [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if dataType is not one of the supported data types, which are DataBuffer.TYPE\_BYTE, DataBuffer.TYPE\_USHORT [RasterFormatException](http://docs.google.com/java/awt/image/RasterFormatException.html) - if dataBuffer has more than one bank. [NullPointerException](http://docs.google.com/java/lang/NullPointerException.html) - if dataBuffer is null

### createBandedRaster

public static [WritableRaster](http://docs.google.com/java/awt/image/WritableRaster.html) **createBandedRaster**([DataBuffer](http://docs.google.com/java/awt/image/DataBuffer.html) dataBuffer,  
 int w,  
 int h,  
 int scanlineStride,  
 int[] bankIndices,  
 int[] bandOffsets,  
 [Point](http://docs.google.com/java/awt/Point.html) location)

Creates a Raster based on a BandedSampleModel with the specified DataBuffer, width, height, scanline stride, bank indices, and band offsets. The number of bands is inferred from bankIndices.length and bandOffsets.length, which must be the same. The upper left corner of the Raster is given by the location argument. If location is null, (0, 0) will be used.

**Parameters:**dataBuffer - the DataBuffer that contains the image dataw - the width in pixels of the image datah - the height in pixels of the image datascanlineStride - the line stride of the image databankIndices - the bank indices for each bandbandOffsets - the offsets of all bandslocation - the upper-left corner of the Raster **Returns:**a WritableRaster object with the specified DataBuffer, width, height, scanline stride, bank indices and band offsets. **Throws:** [RasterFormatException](http://docs.google.com/java/awt/image/RasterFormatException.html) - if w or h is less than or equal to zero, or computing either location.x + w or location.y + h results in integer overflow [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if dataType is not one of the supported data types, which are DataBuffer.TYPE\_BYTE, DataBuffer.TYPE\_USHORT or DataBuffer.TYPE\_INT [NullPointerException](http://docs.google.com/java/lang/NullPointerException.html) - if dataBuffer is null

### createPackedRaster

public static [WritableRaster](http://docs.google.com/java/awt/image/WritableRaster.html) **createPackedRaster**([DataBuffer](http://docs.google.com/java/awt/image/DataBuffer.html) dataBuffer,  
 int w,  
 int h,  
 int scanlineStride,  
 int[] bandMasks,  
 [Point](http://docs.google.com/java/awt/Point.html) location)

Creates a Raster based on a SinglePixelPackedSampleModel with the specified DataBuffer, width, height, scanline stride, and band masks. The number of bands is inferred from bandMasks.length. The upper left corner of the Raster is given by the location argument. If location is null, (0, 0) will be used.

**Parameters:**dataBuffer - the DataBuffer that contains the image dataw - the width in pixels of the image datah - the height in pixels of the image datascanlineStride - the line stride of the image databandMasks - an array containing an entry for each bandlocation - the upper-left corner of the Raster **Returns:**a WritableRaster object with the specified DataBuffer, width, height, scanline stride, and band masks. **Throws:** [RasterFormatException](http://docs.google.com/java/awt/image/RasterFormatException.html) - if w or h is less than or equal to zero, or computing either location.x + w or location.y + h results in integer overflow [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if dataType is not one of the supported data types, which are DataBuffer.TYPE\_BYTE, DataBuffer.TYPE\_USHORT or DataBuffer.TYPE\_INT [RasterFormatException](http://docs.google.com/java/awt/image/RasterFormatException.html) - if dataBuffer has more than one bank. [NullPointerException](http://docs.google.com/java/lang/NullPointerException.html) - if dataBuffer is null

### createPackedRaster

public static [WritableRaster](http://docs.google.com/java/awt/image/WritableRaster.html) **createPackedRaster**([DataBuffer](http://docs.google.com/java/awt/image/DataBuffer.html) dataBuffer,  
 int w,  
 int h,  
 int bitsPerPixel,  
 [Point](http://docs.google.com/java/awt/Point.html) location)

Creates a Raster based on a MultiPixelPackedSampleModel with the specified DataBuffer, width, height, and bits per pixel. The upper left corner of the Raster is given by the location argument. If location is null, (0, 0) will be used.

**Parameters:**dataBuffer - the DataBuffer that contains the image dataw - the width in pixels of the image datah - the height in pixels of the image databitsPerPixel - the number of bits for each pixellocation - the upper-left corner of the Raster **Returns:**a WritableRaster object with the specified DataBuffer, width, height, and bits per pixel. **Throws:** [RasterFormatException](http://docs.google.com/java/awt/image/RasterFormatException.html) - if w or h is less than or equal to zero, or computing either location.x + w or location.y + h results in integer overflow [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if dataType is not one of the supported data types, which are DataBuffer.TYPE\_BYTE, DataBuffer.TYPE\_USHORT or DataBuffer.TYPE\_INT [RasterFormatException](http://docs.google.com/java/awt/image/RasterFormatException.html) - if dataBuffer has more than one bank. [NullPointerException](http://docs.google.com/java/lang/NullPointerException.html) - if dataBuffer is null

### createRaster

public static [Raster](http://docs.google.com/java/awt/image/Raster.html) **createRaster**([SampleModel](http://docs.google.com/java/awt/image/SampleModel.html) sm,  
 [DataBuffer](http://docs.google.com/java/awt/image/DataBuffer.html) db,  
 [Point](http://docs.google.com/java/awt/Point.html) location)

Creates a Raster with the specified SampleModel and DataBuffer. The upper left corner of the Raster is given by the location argument. If location is null, (0, 0) will be used.

**Parameters:**sm - the specified SampleModeldb - the specified DataBufferlocation - the upper-left corner of the Raster **Returns:**a Raster with the specified SampleModel, DataBuffer, and location. **Throws:** [RasterFormatException](http://docs.google.com/java/awt/image/RasterFormatException.html) - if computing either location.x + sm.getWidth() or location.y + sm.getHeight() results in integer overflow [RasterFormatException](http://docs.google.com/java/awt/image/RasterFormatException.html) - if dataBuffer has more than one bank and the sampleModel is PixelInterleavedSampleModel, SinglePixelPackedSampleModel, or MultiPixelPackedSampleModel. [NullPointerException](http://docs.google.com/java/lang/NullPointerException.html) - if either SampleModel or DataBuffer is null

### createWritableRaster

public static [WritableRaster](http://docs.google.com/java/awt/image/WritableRaster.html) **createWritableRaster**([SampleModel](http://docs.google.com/java/awt/image/SampleModel.html) sm,  
 [Point](http://docs.google.com/java/awt/Point.html) location)

Creates a WritableRaster with the specified SampleModel. The upper left corner of the Raster is given by the location argument. If location is null, (0, 0) will be used.

**Parameters:**sm - the specified SampleModellocation - the upper-left corner of the WritableRaster **Returns:**a WritableRaster with the specified SampleModel and location. **Throws:** [RasterFormatException](http://docs.google.com/java/awt/image/RasterFormatException.html) - if computing either location.x + sm.getWidth() or location.y + sm.getHeight() results in integer overflow

### createWritableRaster

public static [WritableRaster](http://docs.google.com/java/awt/image/WritableRaster.html) **createWritableRaster**([SampleModel](http://docs.google.com/java/awt/image/SampleModel.html) sm,  
 [DataBuffer](http://docs.google.com/java/awt/image/DataBuffer.html) db,  
 [Point](http://docs.google.com/java/awt/Point.html) location)

Creates a WritableRaster with the specified SampleModel and DataBuffer. The upper left corner of the Raster is given by the location argument. If location is null, (0, 0) will be used.

**Parameters:**sm - the specified SampleModeldb - the specified DataBufferlocation - the upper-left corner of the WritableRaster **Returns:**a WritableRaster with the specified SampleModel, DataBuffer, and location. **Throws:** [RasterFormatException](http://docs.google.com/java/awt/image/RasterFormatException.html) - if computing either location.x + sm.getWidth() or location.y + sm.getHeight() results in integer overflow [RasterFormatException](http://docs.google.com/java/awt/image/RasterFormatException.html) - if dataBuffer has more than one bank and the sampleModel is PixelInterleavedSampleModel, SinglePixelPackedSampleModel, or MultiPixelPackedSampleModel. [NullPointerException](http://docs.google.com/java/lang/NullPointerException.html) - if either SampleModel or DataBuffer is null

### getParent

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

Returns the parent Raster (if any) of this Raster or null.

**Returns:**the parent Raster or null.

### getSampleModelTranslateX

public final int **getSampleModelTranslateX**()

Returns the X translation from the coordinate system of the SampleModel to that of the Raster. To convert a pixel's X coordinate from the Raster coordinate system to the SampleModel coordinate system, this value must be subtracted.

**Returns:**the X translation from the coordinate space of the Raster's SampleModel to that of the Raster.

### getSampleModelTranslateY

public final int **getSampleModelTranslateY**()

Returns the Y translation from the coordinate system of the SampleModel to that of the Raster. To convert a pixel's Y coordinate from the Raster coordinate system to the SampleModel coordinate system, this value must be subtracted.

**Returns:**the Y translation from the coordinate space of the Raster's SampleModel to that of the Raster.

### createCompatibleWritableRaster

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

Create a compatible WritableRaster the same size as this Raster with the same SampleModel and a new initialized DataBuffer.

**Returns:**a compatible WritableRaster with the same sample model and a new data buffer.

### createCompatibleWritableRaster

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

Create a compatible WritableRaster with the specified size, a new SampleModel, and a new initialized DataBuffer.

**Parameters:**w - the specified width of the new WritableRasterh - the specified height of the new WritableRaster **Returns:**a compatible WritableRaster with the specified size and a new sample model and data buffer. **Throws:** [RasterFormatException](http://docs.google.com/java/awt/image/RasterFormatException.html) - if the width or height is less than or equal to zero.

### createCompatibleWritableRaster

public [WritableRaster](http://docs.google.com/java/awt/image/WritableRaster.html) **createCompatibleWritableRaster**([Rectangle](http://docs.google.com/java/awt/Rectangle.html) rect)

Create a compatible WritableRaster with location (minX, minY) and size (width, height) specified by rect, a new SampleModel, and a new initialized DataBuffer.

**Parameters:**rect - a Rectangle that specifies the size and location of the WritableRaster **Returns:**a compatible WritableRaster with the specified size and location and a new sample model and data buffer. **Throws:** [RasterFormatException](http://docs.google.com/java/awt/image/RasterFormatException.html) - if rect has width or height less than or equal to zero, or computing either rect.x + rect.width or rect.y + rect.height results in integer overflow [NullPointerException](http://docs.google.com/java/lang/NullPointerException.html) - if rect is null

### createCompatibleWritableRaster

public [WritableRaster](http://docs.google.com/java/awt/image/WritableRaster.html) **createCompatibleWritableRaster**(int x,  
 int y,  
 int w,  
 int h)

Create a compatible WritableRaster with the specified location (minX, minY) and size (width, height), a new SampleModel, and a new initialized DataBuffer.

**Parameters:**x - the X coordinate of the upper-left corner of the WritableRastery - the Y coordinate of the upper-left corner of the WritableRasterw - the specified width of the WritableRasterh - the specified height of the WritableRaster **Returns:**a compatible WritableRaster with the specified size and location and a new sample model and data buffer. **Throws:** [RasterFormatException](http://docs.google.com/java/awt/image/RasterFormatException.html) - if w or h is less than or equal to zero, or computing either x + w or y + h results in integer overflow

### createTranslatedChild

public [Raster](http://docs.google.com/java/awt/image/Raster.html) **createTranslatedChild**(int childMinX,  
 int childMinY)

Create a Raster with the same size, SampleModel and DataBuffer as this one, but with a different location. The new Raster will possess a reference to the current Raster, accessible through its getParent() method.

**Parameters:**childMinX - the X coordinate of the upper-left corner of the new RasterchildMinY - the Y coordinate of the upper-left corner of the new Raster **Returns:**a new Raster with the same size, SampleModel, and DataBuffer as this Raster, but with the specified location. **Throws:** [RasterFormatException](http://docs.google.com/java/awt/image/RasterFormatException.html) - if computing either childMinX + this.getWidth() or childMinY + this.getHeight() results in integer overflow

### createChild

public [Raster](http://docs.google.com/java/awt/image/Raster.html) **createChild**(int parentX,  
 int parentY,  
 int width,  
 int height,  
 int childMinX,  
 int childMinY,  
 int[] bandList)

Returns a new Raster which shares all or part of this Raster's DataBuffer. The new Raster will possess a reference to the current Raster, accessible through its getParent() method.

The parentX, parentY, width and height parameters form a Rectangle in this Raster's coordinate space, indicating the area of pixels to be shared. An error will be thrown if this Rectangle is not contained with the bounds of the current Raster.

The new Raster may additionally be translated to a different coordinate system for the plane than that used by the current Raster. The childMinX and childMinY parameters give the new (x, y) coordinate of the upper-left pixel of the returned Raster; the coordinate (childMinX, childMinY) in the new Raster will map to the same pixel as the coordinate (parentX, parentY) in the current Raster.

The new Raster may be defined to contain only a subset of the bands of the current Raster, possibly reordered, by means of the bandList parameter. If bandList is null, it is taken to include all of the bands of the current Raster in their current order.

To create a new Raster that contains a subregion of the current Raster, but shares its coordinate system and bands, this method should be called with childMinX equal to parentX, childMinY equal to parentY, and bandList equal to null.

**Parameters:**parentX - The X coordinate of the upper-left corner in this Raster's coordinatesparentY - The Y coordinate of the upper-left corner in this Raster's coordinateswidth - Width of the region starting at (parentX, parentY)height - Height of the region starting at (parentX, parentY).childMinX - The X coordinate of the upper-left corner of the returned RasterchildMinY - The Y coordinate of the upper-left corner of the returned RasterbandList - Array of band indices, or null to use all bands **Returns:**a new Raster. **Throws:** [RasterFormatException](http://docs.google.com/java/awt/image/RasterFormatException.html) - if the specified subregion is outside of the raster bounds. [RasterFormatException](http://docs.google.com/java/awt/image/RasterFormatException.html) - if width or height is less than or equal to zero, or computing any of parentX + width, parentY + height, childMinX + width, or childMinY + height results in integer overflow

### getBounds

public [Rectangle](http://docs.google.com/java/awt/Rectangle.html) **getBounds**()

Returns the bounding Rectangle of this Raster. This function returns the same information as getMinX/MinY/Width/Height.

**Returns:**the bounding box of this Raster.

### getMinX

public final int **getMinX**()

Returns the minimum valid X coordinate of the Raster.

**Returns:**the minimum x coordinate of this Raster.

### getMinY

public final int **getMinY**()

Returns the minimum valid Y coordinate of the Raster.

**Returns:**the minimum y coordinate of this Raster.

### getWidth

public final int **getWidth**()

Returns the width in pixels of the Raster.

**Returns:**the width of this Raster.

### getHeight

public final int **getHeight**()

Returns the height in pixels of the Raster.

**Returns:**the height of this Raster.

### getNumBands

public final int **getNumBands**()

Returns the number of bands (samples per pixel) in this Raster.

**Returns:**the number of bands of this Raster.

### getNumDataElements

public final int **getNumDataElements**()

Returns the number of data elements needed to transfer one 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 underlying 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 data type of the DataBuffer.

**Returns:**the number of data elements.

### getTransferType

public final 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 underlying 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 data type of the DataBuffer. The TransferType will be one of the types defined in DataBuffer.

**Returns:**this transfer type.

### getDataBuffer

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

Returns the DataBuffer associated with this Raster.

**Returns:**the DataBuffer of this Raster.

### getSampleModel

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

Returns the SampleModel that describes the layout of the image data.

**Returns:**the SampleModel of this Raster.

### 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) outData)

Returns data for a single pixel in a primitive array of type TransferType. For image data supported by the Java 2D(tm) 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. An ArrayIndexOutOfBoundsException may be thrown if the coordinates are not in bounds. However, explicit bounds checking is not guaranteed. A ClassCastException will be thrown if the input object is non null and references anything other than an array of TransferType.

**Parameters:**x - The X coordinate of the pixel locationy - The Y coordinate of the pixel locationoutData - An object reference to an array of type defined by getTransferType() and length getNumDataElements(). If null, an array of appropriate type and size will be allocated **Returns:**An object reference to an array of type defined by getTransferType() with the requested pixel data. **Throws:** [ArrayIndexOutOfBoundsException](http://docs.google.com/java/lang/ArrayIndexOutOfBoundsException.html) - if the coordinates are not in bounds, or if outData is too small to hold the output.**See Also:**[SampleModel.getDataElements(int, int, Object, DataBuffer)](http://docs.google.com/java/awt/image/SampleModel.html#getDataElements(int,%20int,%20java.lang.Object,%20java.awt.image.DataBuffer))

### 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) outData)

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. An ArrayIndexOutOfBoundsException may be thrown if the coordinates are not in bounds. However, explicit bounds checking is not guaranteed. A ClassCastException will be thrown if the input object is non null and references anything other than an array of TransferType.

**Parameters:**x - The X coordinate of the upper-left pixel locationy - The Y coordinate of the upper-left pixel locationw - Width of the pixel rectangleh - Height of the pixel rectangleoutData - An object reference to an array of type defined by getTransferType() and length w\*h\*getNumDataElements(). If null, an array of appropriate type and size will be allocated. **Returns:**An object reference to an array of type defined by getTransferType() with the requested pixel data. **Throws:** [ArrayIndexOutOfBoundsException](http://docs.google.com/java/lang/ArrayIndexOutOfBoundsException.html) - if the coordinates are not in bounds, or if outData is too small to hold the output.**See Also:**[SampleModel.getDataElements(int, int, int, int, Object, DataBuffer)](http://docs.google.com/java/awt/image/SampleModel.html#getDataElements(int,%20int,%20int,%20int,%20java.lang.Object,%20java.awt.image.DataBuffer))

### getPixel

public int[] **getPixel**(int x,  
 int y,  
 int[] iArray)

Returns the samples in an array of int for the specified pixel. An ArrayIndexOutOfBoundsException may be thrown if the coordinates are not in bounds. However, explicit bounds checking is not guaranteed.

**Parameters:**x - The X coordinate of the pixel locationy - The Y coordinate of the pixel locationiArray - An optionally preallocated int array **Returns:**the samples for the specified pixel. **Throws:** [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.

### getPixel

public float[] **getPixel**(int x,  
 int y,  
 float[] fArray)

Returns the samples in an array of float for the specified pixel. An ArrayIndexOutOfBoundsException may be thrown if the coordinates are not in bounds. However, explicit bounds checking is not guaranteed.

**Parameters:**x - The X coordinate of the pixel locationy - The Y coordinate of the pixel locationfArray - An optionally preallocated float array **Returns:**the samples for the specified pixel. **Throws:** [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.

### getPixel

public double[] **getPixel**(int x,  
 int y,  
 double[] dArray)

Returns the samples in an array of double for the specified pixel. An ArrayIndexOutOfBoundsException may be thrown if the coordinates are not in bounds. However, explicit bounds checking is not guaranteed.

**Parameters:**x - The X coordinate of the pixel locationy - The Y coordinate of the pixel locationdArray - An optionally preallocated double array **Returns:**the samples for the specified pixel. **Throws:** [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.

### getPixels

public int[] **getPixels**(int x,  
 int y,  
 int w,  
 int h,  
 int[] iArray)

Returns an int array containing all samples for a rectangle of pixels, one sample per array element. An ArrayIndexOutOfBoundsException may be thrown if the coordinates are not in bounds. However, explicit bounds checking is not guaranteed.

**Parameters:**x - The X coordinate of the upper-left pixel locationy - The Y coordinate of the upper-left pixel locationw - Width of the pixel rectangleh - Height of the pixel rectangleiArray - An optionally pre-allocated int array **Returns:**the samples for the specified rectangle of pixels. **Throws:** [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.

### getPixels

public float[] **getPixels**(int x,  
 int y,  
 int w,  
 int h,  
 float[] fArray)

Returns a float array containing all samples for a rectangle of pixels, one sample per array element. An ArrayIndexOutOfBoundsException may be thrown if the coordinates are not in bounds. However, explicit bounds checking is not guaranteed.

**Parameters:**x - The X coordinate of the pixel locationy - The Y coordinate of the pixel locationw - Width of the pixel rectangleh - Height of the pixel rectanglefArray - An optionally pre-allocated float array **Returns:**the samples for the specified rectangle of pixels. **Throws:** [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.

### getPixels

public double[] **getPixels**(int x,  
 int y,  
 int w,  
 int h,  
 double[] dArray)

Returns a double array containing all samples for a rectangle of pixels, one sample per array element. An ArrayIndexOutOfBoundsException may be thrown if the coordinates are not in bounds. However, explicit bounds checking is not guaranteed.

**Parameters:**x - The X coordinate of the upper-left pixel locationy - The Y coordinate of the upper-left pixel locationw - Width of the pixel rectangleh - Height of the pixel rectangledArray - An optionally pre-allocated double array **Returns:**the samples for the specified rectangle of pixels. **Throws:** [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.

### getSample

public int **getSample**(int x,  
 int y,  
 int b)

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

**Parameters:**x - The X coordinate of the pixel locationy - The Y coordinate of the pixel locationb - The band to return **Returns:**the sample in the specified band for the pixel at the specified coordinate. **Throws:** [ArrayIndexOutOfBoundsException](http://docs.google.com/java/lang/ArrayIndexOutOfBoundsException.html) - if the coordinates or the band index are not in bounds.

### getSampleFloat

public float **getSampleFloat**(int x,  
 int y,  
 int b)

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

**Parameters:**x - The X coordinate of the pixel locationy - The Y coordinate of the pixel locationb - The band to return **Returns:**the sample in the specified band for the pixel at the specified coordinate. **Throws:** [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)

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

**Parameters:**x - The X coordinate of the pixel locationy - The Y coordinate of the pixel locationb - The band to return **Returns:**the sample in the specified band for the pixel at the specified coordinate. **Throws:** [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)

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

**Parameters:**x - The X coordinate of the upper-left pixel locationy - The Y coordinate of the upper-left pixel locationw - Width of the pixel rectangleh - Height of the pixel rectangleb - The band to returniArray - An optionally pre-allocated int array **Returns:**the samples for the specified band for the specified rectangle of pixels. **Throws:** [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.

### getSamples

public float[] **getSamples**(int x,  
 int y,  
 int w,  
 int h,  
 int b,  
 float[] fArray)

Returns the samples for a specified band for the specified rectangle of pixels in a float array, one sample per array element. An ArrayIndexOutOfBoundsException may be thrown if the coordinates are not in bounds. However, explicit bounds checking is not guaranteed.

**Parameters:**x - The X coordinate of the upper-left pixel locationy - The Y coordinate of the upper-left pixel locationw - Width of the pixel rectangleh - Height of the pixel rectangleb - The band to returnfArray - An optionally pre-allocated float array **Returns:**the samples for the specified band for the specified rectangle of pixels. **Throws:** [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.

### getSamples

public double[] **getSamples**(int x,  
 int y,  
 int w,  
 int h,  
 int b,  
 double[] dArray)

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

**Parameters:**x - The X coordinate of the upper-left pixel locationy - The Y coordinate of the upper-left pixel locationw - Width of the pixel rectangleh - Height of the pixel rectangleb - The band to returndArray - An optionally pre-allocated double array **Returns:**the samples for the specified band for the specified rectangle of pixels. **Throws:** [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.

| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/Raster.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/PixelInterleavedSampleModel.html)   [**NEXT CLASS**](http://docs.google.com/java/awt/image/RasterFormatException.html) | [**FRAMES**](http://docs.google.com/index.html?java/awt/image/Raster.html)    [**NO FRAMES**](http://docs.google.com/Raster.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | [FIELD](#3znysh7) | [CONSTR](#2et92p0) | [METHOD](#tyjcwt) | DETAIL: [FIELD](#1t3h5sf) | [CONSTR](#3j2qqm3) | [METHOD](#1ci93xb) |

[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).