| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/ParameterBlock.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/renderable/ContextualRenderedImageFactory.html)   [**NEXT CLASS**](http://docs.google.com/java/awt/image/renderable/RenderableImage.html) | [**FRAMES**](http://docs.google.com/index.html?java/awt/image/renderable/ParameterBlock.html)    [**NO FRAMES**](http://docs.google.com/ParameterBlock.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | [FIELD](#3znysh7) | [CONSTR](#2et92p0) | [METHOD](#tyjcwt) | DETAIL: [FIELD](#1t3h5sf) | [CONSTR](#17dp8vu) | [METHOD](#35nkun2) |

## **java.awt.image.renderable**

Class ParameterBlock

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

**All Implemented Interfaces:** [Serializable](http://docs.google.com/java/io/Serializable.html), [Cloneable](http://docs.google.com/java/lang/Cloneable.html)

public class **ParameterBlock**extends [Object](http://docs.google.com/java/lang/Object.html)implements [Cloneable](http://docs.google.com/java/lang/Cloneable.html), [Serializable](http://docs.google.com/java/io/Serializable.html)

A ParameterBlock encapsulates all the information about sources and parameters (Objects) required by a RenderableImageOp, or other classes that process images.

Although it is possible to place arbitrary objects in the source Vector, users of this class may impose semantic constraints such as requiring all sources to be RenderedImages or RenderableImage. ParameterBlock itself is merely a container and performs no checking on source or parameter types.

All parameters in a ParameterBlock are objects; convenience add and set methods are available that take arguments of base type and construct the appropriate subclass of Number (such as Integer or Float). Corresponding get methods perform a downward cast and have return values of base type; an exception will be thrown if the stored values do not have the correct type. There is no way to distinguish between the results of "short s; add(s)" and "add(new Short(s))".

Note that the get and set methods operate on references. Therefore, one must be careful not to share references between ParameterBlocks when this is inappropriate. For example, to create a new ParameterBlock that is equal to an old one except for an added source, one might be tempted to write:

ParameterBlock addSource(ParameterBlock pb, RenderableImage im) {  
 ParameterBlock pb1 = new ParameterBlock(pb.getSources());  
 pb1.addSource(im);  
 return pb1;  
 }

This code will have the side effect of altering the original ParameterBlock, since the getSources operation returned a reference to its source Vector. Both pb and pb1 share their source Vector, and a change in either is visible to both.

A correct way to write the addSource function is to clone the source Vector:

ParameterBlock addSource (ParameterBlock pb, RenderableImage im) {  
 ParameterBlock pb1 = new ParameterBlock(pb.getSources().clone());  
 pb1.addSource(im);  
 return pb1;  
 }

The clone method of ParameterBlock has been defined to perform a clone of both the source and parameter Vectors for this reason. A standard, shallow clone is available as shallowClone.

The addSource, setSource, add, and set methods are defined to return 'this' after adding their argument. This allows use of syntax like:

ParameterBlock pb = new ParameterBlock();  
 op = new RenderableImageOp("operation", pb.add(arg1).add(arg2));

**See Also:**[Serialized Form](http://docs.google.com/serialized-form.html#java.awt.image.renderable.ParameterBlock)

| **Field Summary** | |
| --- | --- |
| protected  [Vector](http://docs.google.com/java/util/Vector.html)<[Object](http://docs.google.com/java/lang/Object.html)> | [**parameters**](http://docs.google.com/java/awt/image/renderable/ParameterBlock.html#parameters)            A Vector of non-source parameters, stored as arbitrary Objects. |
| protected  [Vector](http://docs.google.com/java/util/Vector.html)<[Object](http://docs.google.com/java/lang/Object.html)> | [**sources**](http://docs.google.com/java/awt/image/renderable/ParameterBlock.html#sources)            A Vector of sources, stored as arbitrary Objects. |

| **Constructor Summary** | |
| --- | --- |
| [**ParameterBlock**](http://docs.google.com/java/awt/image/renderable/ParameterBlock.html#ParameterBlock())()            A dummy constructor. |
| [**ParameterBlock**](http://docs.google.com/java/awt/image/renderable/ParameterBlock.html#ParameterBlock(java.util.Vector))([Vector](http://docs.google.com/java/util/Vector.html)<[Object](http://docs.google.com/java/lang/Object.html)> sources)            Constructs a ParameterBlock with a given Vector of sources. |
| [**ParameterBlock**](http://docs.google.com/java/awt/image/renderable/ParameterBlock.html#ParameterBlock(java.util.Vector,%20java.util.Vector))([Vector](http://docs.google.com/java/util/Vector.html)<[Object](http://docs.google.com/java/lang/Object.html)> sources, [Vector](http://docs.google.com/java/util/Vector.html)<[Object](http://docs.google.com/java/lang/Object.html)> parameters)            Constructs a ParameterBlock with a given Vector of sources and Vector of parameters. |

| **Method Summary** | |
| --- | --- |
| [ParameterBlock](http://docs.google.com/java/awt/image/renderable/ParameterBlock.html) | [**add**](http://docs.google.com/java/awt/image/renderable/ParameterBlock.html#add(byte))(byte b)            Adds a Byte to the list of parameters. |
| [ParameterBlock](http://docs.google.com/java/awt/image/renderable/ParameterBlock.html) | [**add**](http://docs.google.com/java/awt/image/renderable/ParameterBlock.html#add(char))(char c)            Adds a Character to the list of parameters. |
| [ParameterBlock](http://docs.google.com/java/awt/image/renderable/ParameterBlock.html) | [**add**](http://docs.google.com/java/awt/image/renderable/ParameterBlock.html#add(double))(double d)            Adds a Double to the list of parameters. |
| [ParameterBlock](http://docs.google.com/java/awt/image/renderable/ParameterBlock.html) | [**add**](http://docs.google.com/java/awt/image/renderable/ParameterBlock.html#add(float))(float f)            Adds a Float to the list of parameters. |
| [ParameterBlock](http://docs.google.com/java/awt/image/renderable/ParameterBlock.html) | [**add**](http://docs.google.com/java/awt/image/renderable/ParameterBlock.html#add(int))(int i)            Adds a Integer to the list of parameters. |
| [ParameterBlock](http://docs.google.com/java/awt/image/renderable/ParameterBlock.html) | [**add**](http://docs.google.com/java/awt/image/renderable/ParameterBlock.html#add(long))(long l)            Adds a Long to the list of parameters. |
| [ParameterBlock](http://docs.google.com/java/awt/image/renderable/ParameterBlock.html) | [**add**](http://docs.google.com/java/awt/image/renderable/ParameterBlock.html#add(java.lang.Object))([Object](http://docs.google.com/java/lang/Object.html) obj)            Adds an object to the list of parameters. |
| [ParameterBlock](http://docs.google.com/java/awt/image/renderable/ParameterBlock.html) | [**add**](http://docs.google.com/java/awt/image/renderable/ParameterBlock.html#add(short))(short s)            Adds a Short to the list of parameters. |
| [ParameterBlock](http://docs.google.com/java/awt/image/renderable/ParameterBlock.html) | [**addSource**](http://docs.google.com/java/awt/image/renderable/ParameterBlock.html#addSource(java.lang.Object))([Object](http://docs.google.com/java/lang/Object.html) source)            Adds an image to end of the list of sources. |
| [Object](http://docs.google.com/java/lang/Object.html) | [**clone**](http://docs.google.com/java/awt/image/renderable/ParameterBlock.html#clone())()            Creates a copy of a ParameterBlock. |
| byte | [**getByteParameter**](http://docs.google.com/java/awt/image/renderable/ParameterBlock.html#getByteParameter(int))(int index)            A convenience method to return a parameter as a byte. |
| char | [**getCharParameter**](http://docs.google.com/java/awt/image/renderable/ParameterBlock.html#getCharParameter(int))(int index)            A convenience method to return a parameter as a char. |
| double | [**getDoubleParameter**](http://docs.google.com/java/awt/image/renderable/ParameterBlock.html#getDoubleParameter(int))(int index)            A convenience method to return a parameter as a double. |
| float | [**getFloatParameter**](http://docs.google.com/java/awt/image/renderable/ParameterBlock.html#getFloatParameter(int))(int index)            A convenience method to return a parameter as a float. |
| int | [**getIntParameter**](http://docs.google.com/java/awt/image/renderable/ParameterBlock.html#getIntParameter(int))(int index)            A convenience method to return a parameter as an int. |
| long | [**getLongParameter**](http://docs.google.com/java/awt/image/renderable/ParameterBlock.html#getLongParameter(int))(int index)            A convenience method to return a parameter as a long. |
| int | [**getNumParameters**](http://docs.google.com/java/awt/image/renderable/ParameterBlock.html#getNumParameters())()            Returns the number of parameters (not including source images). |
| int | [**getNumSources**](http://docs.google.com/java/awt/image/renderable/ParameterBlock.html#getNumSources())()            Returns the number of source images. |
| [Object](http://docs.google.com/java/lang/Object.html) | [**getObjectParameter**](http://docs.google.com/java/awt/image/renderable/ParameterBlock.html#getObjectParameter(int))(int index)            Gets a parameter as an object. |
| [Class](http://docs.google.com/java/lang/Class.html)[] | [**getParamClasses**](http://docs.google.com/java/awt/image/renderable/ParameterBlock.html#getParamClasses())()            Returns an array of Class objects describing the types of the parameters. |
| [Vector](http://docs.google.com/java/util/Vector.html)<[Object](http://docs.google.com/java/lang/Object.html)> | [**getParameters**](http://docs.google.com/java/awt/image/renderable/ParameterBlock.html#getParameters())()            Returns the entire Vector of parameters. |
| [RenderableImage](http://docs.google.com/java/awt/image/renderable/RenderableImage.html) | [**getRenderableSource**](http://docs.google.com/java/awt/image/renderable/ParameterBlock.html#getRenderableSource(int))(int index)            Returns a source as a RenderableImage. |
| [RenderedImage](http://docs.google.com/java/awt/image/RenderedImage.html) | [**getRenderedSource**](http://docs.google.com/java/awt/image/renderable/ParameterBlock.html#getRenderedSource(int))(int index)            Returns a source as a RenderedImage. |
| short | [**getShortParameter**](http://docs.google.com/java/awt/image/renderable/ParameterBlock.html#getShortParameter(int))(int index)            A convenience method to return a parameter as a short. |
| [Object](http://docs.google.com/java/lang/Object.html) | [**getSource**](http://docs.google.com/java/awt/image/renderable/ParameterBlock.html#getSource(int))(int index)            Returns a source as a general Object. |
| [Vector](http://docs.google.com/java/util/Vector.html)<[Object](http://docs.google.com/java/lang/Object.html)> | [**getSources**](http://docs.google.com/java/awt/image/renderable/ParameterBlock.html#getSources())()            Returns the entire Vector of sources. |
| void | [**removeParameters**](http://docs.google.com/java/awt/image/renderable/ParameterBlock.html#removeParameters())()            Clears the list of parameters. |
| void | [**removeSources**](http://docs.google.com/java/awt/image/renderable/ParameterBlock.html#removeSources())()            Clears the list of source images. |
| [ParameterBlock](http://docs.google.com/java/awt/image/renderable/ParameterBlock.html) | [**set**](http://docs.google.com/java/awt/image/renderable/ParameterBlock.html#set(byte,%20int))(byte b, int index)            Replaces an Object in the list of parameters with a Byte. |
| [ParameterBlock](http://docs.google.com/java/awt/image/renderable/ParameterBlock.html) | [**set**](http://docs.google.com/java/awt/image/renderable/ParameterBlock.html#set(char,%20int))(char c, int index)            Replaces an Object in the list of parameters with a Character. |
| [ParameterBlock](http://docs.google.com/java/awt/image/renderable/ParameterBlock.html) | [**set**](http://docs.google.com/java/awt/image/renderable/ParameterBlock.html#set(double,%20int))(double d, int index)            Replaces an Object in the list of parameters with a Double. |
| [ParameterBlock](http://docs.google.com/java/awt/image/renderable/ParameterBlock.html) | [**set**](http://docs.google.com/java/awt/image/renderable/ParameterBlock.html#set(float,%20int))(float f, int index)            Replaces an Object in the list of parameters with a Float. |
| [ParameterBlock](http://docs.google.com/java/awt/image/renderable/ParameterBlock.html) | [**set**](http://docs.google.com/java/awt/image/renderable/ParameterBlock.html#set(int,%20int))(int i, int index)            Replaces an Object in the list of parameters with an Integer. |
| [ParameterBlock](http://docs.google.com/java/awt/image/renderable/ParameterBlock.html) | [**set**](http://docs.google.com/java/awt/image/renderable/ParameterBlock.html#set(long,%20int))(long l, int index)            Replaces an Object in the list of parameters with a Long. |
| [ParameterBlock](http://docs.google.com/java/awt/image/renderable/ParameterBlock.html) | [**set**](http://docs.google.com/java/awt/image/renderable/ParameterBlock.html#set(java.lang.Object,%20int))([Object](http://docs.google.com/java/lang/Object.html) obj, int index)            Replaces an Object in the list of parameters. |
| [ParameterBlock](http://docs.google.com/java/awt/image/renderable/ParameterBlock.html) | [**set**](http://docs.google.com/java/awt/image/renderable/ParameterBlock.html#set(short,%20int))(short s, int index)            Replaces an Object in the list of parameters with a Short. |
| void | [**setParameters**](http://docs.google.com/java/awt/image/renderable/ParameterBlock.html#setParameters(java.util.Vector))([Vector](http://docs.google.com/java/util/Vector.html)<[Object](http://docs.google.com/java/lang/Object.html)> parameters)            Sets the entire Vector of parameters to a given Vector. |
| [ParameterBlock](http://docs.google.com/java/awt/image/renderable/ParameterBlock.html) | [**setSource**](http://docs.google.com/java/awt/image/renderable/ParameterBlock.html#setSource(java.lang.Object,%20int))([Object](http://docs.google.com/java/lang/Object.html) source, int index)            Replaces an entry in the list of source with a new source. |
| void | [**setSources**](http://docs.google.com/java/awt/image/renderable/ParameterBlock.html#setSources(java.util.Vector))([Vector](http://docs.google.com/java/util/Vector.html)<[Object](http://docs.google.com/java/lang/Object.html)> sources)            Sets the entire Vector of sources to a given Vector. |
| [Object](http://docs.google.com/java/lang/Object.html) | [**shallowClone**](http://docs.google.com/java/awt/image/renderable/ParameterBlock.html#shallowClone())()            Creates a shallow copy of a ParameterBlock. |

| **Methods inherited from class java.lang.**[**Object**](http://docs.google.com/java/lang/Object.html) |
| --- |
| [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** |
| --- |

### sources

protected [Vector](http://docs.google.com/java/util/Vector.html)<[Object](http://docs.google.com/java/lang/Object.html)> **sources**

A Vector of sources, stored as arbitrary Objects.

### parameters

protected [Vector](http://docs.google.com/java/util/Vector.html)<[Object](http://docs.google.com/java/lang/Object.html)> **parameters**

A Vector of non-source parameters, stored as arbitrary Objects.

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

### ParameterBlock

public **ParameterBlock**()

A dummy constructor.

### ParameterBlock

public **ParameterBlock**([Vector](http://docs.google.com/java/util/Vector.html)<[Object](http://docs.google.com/java/lang/Object.html)> sources)

Constructs a ParameterBlock with a given Vector of sources.

**Parameters:**sources - a Vector of source images

### ParameterBlock

public **ParameterBlock**([Vector](http://docs.google.com/java/util/Vector.html)<[Object](http://docs.google.com/java/lang/Object.html)> sources,  
 [Vector](http://docs.google.com/java/util/Vector.html)<[Object](http://docs.google.com/java/lang/Object.html)> parameters)

Constructs a ParameterBlock with a given Vector of sources and Vector of parameters.

**Parameters:**sources - a Vector of source imagesparameters - a Vector of parameters to be used in the rendering operation

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

### shallowClone

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

Creates a shallow copy of a ParameterBlock. The source and parameter Vectors are copied by reference -- additions or changes will be visible to both versions.

**Returns:**an Object clone of the ParameterBlock.

### clone

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

Creates a copy of a ParameterBlock. The source and parameter Vectors are cloned, but the actual sources and parameters are copied by reference. This allows modifications to the order and number of sources and parameters in the clone to be invisible to the original ParameterBlock. Changes to the shared sources or parameters themselves will still be visible.

**Overrides:**[clone](http://docs.google.com/java/lang/Object.html#clone()) in class [Object](http://docs.google.com/java/lang/Object.html) **Returns:**an Object clone of the ParameterBlock.**See Also:**[Cloneable](http://docs.google.com/java/lang/Cloneable.html)

### addSource

public [ParameterBlock](http://docs.google.com/java/awt/image/renderable/ParameterBlock.html) **addSource**([Object](http://docs.google.com/java/lang/Object.html) source)

Adds an image to end of the list of sources. The image is stored as an object in order to allow new node types in the future.

**Parameters:**source - an image object to be stored in the source list. **Returns:**a new ParameterBlock containing the specified source.

### getSource

public [Object](http://docs.google.com/java/lang/Object.html) **getSource**(int index)

Returns a source as a general Object. The caller must cast it into an appropriate type.

**Parameters:**index - the index of the source to be returned. **Returns:**an Object that represents the source located at the specified index in the sources Vector.**See Also:**[setSource(Object, int)](http://docs.google.com/java/awt/image/renderable/ParameterBlock.html#setSource(java.lang.Object,%20int))

### setSource

public [ParameterBlock](http://docs.google.com/java/awt/image/renderable/ParameterBlock.html) **setSource**([Object](http://docs.google.com/java/lang/Object.html) source,  
 int index)

Replaces an entry in the list of source with a new source. If the index lies beyond the current source list, the list is extended with nulls as needed.

**Parameters:**source - the specified source imageindex - the index into the sources Vector at which to insert the specified source **Returns:**a new ParameterBlock that contains the specified source at the specified index.**See Also:**[getSource(int)](http://docs.google.com/java/awt/image/renderable/ParameterBlock.html#getSource(int))

### getRenderedSource

public [RenderedImage](http://docs.google.com/java/awt/image/RenderedImage.html) **getRenderedSource**(int index)

Returns a source as a RenderedImage. This method is a convenience method. An exception will be thrown if the source is not a RenderedImage.

**Parameters:**index - the index of the source to be returned **Returns:**a RenderedImage that represents the source image that is at the specified index in the sources Vector.

### getRenderableSource

public [RenderableImage](http://docs.google.com/java/awt/image/renderable/RenderableImage.html) **getRenderableSource**(int index)

Returns a source as a RenderableImage. This method is a convenience method. An exception will be thrown if the sources is not a RenderableImage.

**Parameters:**index - the index of the source to be returned **Returns:**a RenderableImage that represents the source image that is at the specified index in the sources Vector.

### getNumSources

public int **getNumSources**()

Returns the number of source images.

**Returns:**the number of source images in the sources Vector.

### getSources

public [Vector](http://docs.google.com/java/util/Vector.html)<[Object](http://docs.google.com/java/lang/Object.html)> **getSources**()

Returns the entire Vector of sources.

**Returns:**the sources Vector.**See Also:**[setSources(Vector)](http://docs.google.com/java/awt/image/renderable/ParameterBlock.html#setSources(java.util.Vector))

### setSources

public void **setSources**([Vector](http://docs.google.com/java/util/Vector.html)<[Object](http://docs.google.com/java/lang/Object.html)> sources)

Sets the entire Vector of sources to a given Vector.

**Parameters:**sources - the Vector of source images**See Also:**[getSources()](http://docs.google.com/java/awt/image/renderable/ParameterBlock.html#getSources())

### removeSources

public void **removeSources**()

Clears the list of source images.

### getNumParameters

public int **getNumParameters**()

Returns the number of parameters (not including source images).

**Returns:**the number of parameters in the parameters Vector.

### getParameters

public [Vector](http://docs.google.com/java/util/Vector.html)<[Object](http://docs.google.com/java/lang/Object.html)> **getParameters**()

Returns the entire Vector of parameters.

**Returns:**the parameters Vector.**See Also:**[setParameters(Vector)](http://docs.google.com/java/awt/image/renderable/ParameterBlock.html#setParameters(java.util.Vector))

### setParameters

public void **setParameters**([Vector](http://docs.google.com/java/util/Vector.html)<[Object](http://docs.google.com/java/lang/Object.html)> parameters)

Sets the entire Vector of parameters to a given Vector.

**Parameters:**parameters - the specified Vector of parameters**See Also:**[getParameters()](http://docs.google.com/java/awt/image/renderable/ParameterBlock.html#getParameters())

### removeParameters

public void **removeParameters**()

Clears the list of parameters.

### add

public [ParameterBlock](http://docs.google.com/java/awt/image/renderable/ParameterBlock.html) **add**([Object](http://docs.google.com/java/lang/Object.html) obj)

Adds an object to the list of parameters.

**Parameters:**obj - the Object to add to the parameters Vector **Returns:**a new ParameterBlock containing the specified parameter.

### add

public [ParameterBlock](http://docs.google.com/java/awt/image/renderable/ParameterBlock.html) **add**(byte b)

Adds a Byte to the list of parameters.

**Parameters:**b - the byte to add to the parameters Vector **Returns:**a new ParameterBlock containing the specified parameter.

### add

public [ParameterBlock](http://docs.google.com/java/awt/image/renderable/ParameterBlock.html) **add**(char c)

Adds a Character to the list of parameters.

**Parameters:**c - the char to add to the parameters Vector **Returns:**a new ParameterBlock containing the specified parameter.

### add

public [ParameterBlock](http://docs.google.com/java/awt/image/renderable/ParameterBlock.html) **add**(short s)

Adds a Short to the list of parameters.

**Parameters:**s - the short to add to the parameters Vector **Returns:**a new ParameterBlock containing the specified parameter.

### add

public [ParameterBlock](http://docs.google.com/java/awt/image/renderable/ParameterBlock.html) **add**(int i)

Adds a Integer to the list of parameters.

**Parameters:**i - the int to add to the parameters Vector **Returns:**a new ParameterBlock containing the specified parameter.

### add

public [ParameterBlock](http://docs.google.com/java/awt/image/renderable/ParameterBlock.html) **add**(long l)

Adds a Long to the list of parameters.

**Parameters:**l - the long to add to the parameters Vector **Returns:**a new ParameterBlock containing the specified parameter.

### add

public [ParameterBlock](http://docs.google.com/java/awt/image/renderable/ParameterBlock.html) **add**(float f)

Adds a Float to the list of parameters.

**Parameters:**f - the float to add to the parameters Vector **Returns:**a new ParameterBlock containing the specified parameter.

### add

public [ParameterBlock](http://docs.google.com/java/awt/image/renderable/ParameterBlock.html) **add**(double d)

Adds a Double to the list of parameters.

**Parameters:**d - the double to add to the parameters Vector **Returns:**a new ParameterBlock containing the specified parameter.

### set

public [ParameterBlock](http://docs.google.com/java/awt/image/renderable/ParameterBlock.html) **set**([Object](http://docs.google.com/java/lang/Object.html) obj,  
 int index)

Replaces an Object in the list of parameters. If the index lies beyond the current source list, the list is extended with nulls as needed.

**Parameters:**obj - the parameter that replaces the parameter at the specified index in the parameters Vectorindex - the index of the parameter to be replaced with the specified parameter **Returns:**a new ParameterBlock containing the specified parameter.

### set

public [ParameterBlock](http://docs.google.com/java/awt/image/renderable/ParameterBlock.html) **set**(byte b,  
 int index)

Replaces an Object in the list of parameters with a Byte. If the index lies beyond the current source list, the list is extended with nulls as needed.

**Parameters:**b - the parameter that replaces the parameter at the specified index in the parameters Vectorindex - the index of the parameter to be replaced with the specified parameter **Returns:**a new ParameterBlock containing the specified parameter.

### set

public [ParameterBlock](http://docs.google.com/java/awt/image/renderable/ParameterBlock.html) **set**(char c,  
 int index)

Replaces an Object in the list of parameters with a Character. If the index lies beyond the current source list, the list is extended with nulls as needed.

**Parameters:**c - the parameter that replaces the parameter at the specified index in the parameters Vectorindex - the index of the parameter to be replaced with the specified parameter **Returns:**a new ParameterBlock containing the specified parameter.

### set

public [ParameterBlock](http://docs.google.com/java/awt/image/renderable/ParameterBlock.html) **set**(short s,  
 int index)

Replaces an Object in the list of parameters with a Short. If the index lies beyond the current source list, the list is extended with nulls as needed.

**Parameters:**s - the parameter that replaces the parameter at the specified index in the parameters Vectorindex - the index of the parameter to be replaced with the specified parameter **Returns:**a new ParameterBlock containing the specified parameter.

### set

public [ParameterBlock](http://docs.google.com/java/awt/image/renderable/ParameterBlock.html) **set**(int i,  
 int index)

Replaces an Object in the list of parameters with an Integer. If the index lies beyond the current source list, the list is extended with nulls as needed.

**Parameters:**i - the parameter that replaces the parameter at the specified index in the parameters Vectorindex - the index of the parameter to be replaced with the specified parameter **Returns:**a new ParameterBlock containing the specified parameter.

### set

public [ParameterBlock](http://docs.google.com/java/awt/image/renderable/ParameterBlock.html) **set**(long l,  
 int index)

Replaces an Object in the list of parameters with a Long. If the index lies beyond the current source list, the list is extended with nulls as needed.

**Parameters:**l - the parameter that replaces the parameter at the specified index in the parameters Vectorindex - the index of the parameter to be replaced with the specified parameter **Returns:**a new ParameterBlock containing the specified parameter.

### set

public [ParameterBlock](http://docs.google.com/java/awt/image/renderable/ParameterBlock.html) **set**(float f,  
 int index)

Replaces an Object in the list of parameters with a Float. If the index lies beyond the current source list, the list is extended with nulls as needed.

**Parameters:**f - the parameter that replaces the parameter at the specified index in the parameters Vectorindex - the index of the parameter to be replaced with the specified parameter **Returns:**a new ParameterBlock containing the specified parameter.

### set

public [ParameterBlock](http://docs.google.com/java/awt/image/renderable/ParameterBlock.html) **set**(double d,  
 int index)

Replaces an Object in the list of parameters with a Double. If the index lies beyond the current source list, the list is extended with nulls as needed.

**Parameters:**d - the parameter that replaces the parameter at the specified index in the parameters Vectorindex - the index of the parameter to be replaced with the specified parameter **Returns:**a new ParameterBlock containing the specified parameter.

### getObjectParameter

public [Object](http://docs.google.com/java/lang/Object.html) **getObjectParameter**(int index)

Gets a parameter as an object.

**Parameters:**index - the index of the parameter to get **Returns:**an Object representing the the parameter at the specified index into the parameters Vector.

### getByteParameter

public byte **getByteParameter**(int index)

A convenience method to return a parameter as a byte. An exception is thrown if the parameter is null or not a Byte.

**Parameters:**index - the index of the parameter to be returned. **Returns:**the parameter at the specified index as a byte value. **Throws:** [ClassCastException](http://docs.google.com/java/lang/ClassCastException.html) - if the parameter at the specified index is not a Byte [NullPointerException](http://docs.google.com/java/lang/NullPointerException.html) - if the parameter at the specified index is null [ArrayIndexOutOfBoundsException](http://docs.google.com/java/lang/ArrayIndexOutOfBoundsException.html) - if index is negative or not less than the current size of this ParameterBlock object

### getCharParameter

public char **getCharParameter**(int index)

A convenience method to return a parameter as a char. An exception is thrown if the parameter is null or not a Character.

**Parameters:**index - the index of the parameter to be returned. **Returns:**the parameter at the specified index as a char value. **Throws:** [ClassCastException](http://docs.google.com/java/lang/ClassCastException.html) - if the parameter at the specified index is not a Character [NullPointerException](http://docs.google.com/java/lang/NullPointerException.html) - if the parameter at the specified index is null [ArrayIndexOutOfBoundsException](http://docs.google.com/java/lang/ArrayIndexOutOfBoundsException.html) - if index is negative or not less than the current size of this ParameterBlock object

### getShortParameter

public short **getShortParameter**(int index)

A convenience method to return a parameter as a short. An exception is thrown if the parameter is null or not a Short.

**Parameters:**index - the index of the parameter to be returned. **Returns:**the parameter at the specified index as a short value. **Throws:** [ClassCastException](http://docs.google.com/java/lang/ClassCastException.html) - if the parameter at the specified index is not a Short [NullPointerException](http://docs.google.com/java/lang/NullPointerException.html) - if the parameter at the specified index is null [ArrayIndexOutOfBoundsException](http://docs.google.com/java/lang/ArrayIndexOutOfBoundsException.html) - if index is negative or not less than the current size of this ParameterBlock object

### getIntParameter

public int **getIntParameter**(int index)

A convenience method to return a parameter as an int. An exception is thrown if the parameter is null or not an Integer.

**Parameters:**index - the index of the parameter to be returned. **Returns:**the parameter at the specified index as a int value. **Throws:** [ClassCastException](http://docs.google.com/java/lang/ClassCastException.html) - if the parameter at the specified index is not a Integer [NullPointerException](http://docs.google.com/java/lang/NullPointerException.html) - if the parameter at the specified index is null [ArrayIndexOutOfBoundsException](http://docs.google.com/java/lang/ArrayIndexOutOfBoundsException.html) - if index is negative or not less than the current size of this ParameterBlock object

### getLongParameter

public long **getLongParameter**(int index)

A convenience method to return a parameter as a long. An exception is thrown if the parameter is null or not a Long.

**Parameters:**index - the index of the parameter to be returned. **Returns:**the parameter at the specified index as a long value. **Throws:** [ClassCastException](http://docs.google.com/java/lang/ClassCastException.html) - if the parameter at the specified index is not a Long [NullPointerException](http://docs.google.com/java/lang/NullPointerException.html) - if the parameter at the specified index is null [ArrayIndexOutOfBoundsException](http://docs.google.com/java/lang/ArrayIndexOutOfBoundsException.html) - if index is negative or not less than the current size of this ParameterBlock object

### getFloatParameter

public float **getFloatParameter**(int index)

A convenience method to return a parameter as a float. An exception is thrown if the parameter is null or not a Float.

**Parameters:**index - the index of the parameter to be returned. **Returns:**the parameter at the specified index as a float value. **Throws:** [ClassCastException](http://docs.google.com/java/lang/ClassCastException.html) - if the parameter at the specified index is not a Float [NullPointerException](http://docs.google.com/java/lang/NullPointerException.html) - if the parameter at the specified index is null [ArrayIndexOutOfBoundsException](http://docs.google.com/java/lang/ArrayIndexOutOfBoundsException.html) - if index is negative or not less than the current size of this ParameterBlock object

### getDoubleParameter

public double **getDoubleParameter**(int index)

A convenience method to return a parameter as a double. An exception is thrown if the parameter is null or not a Double.

**Parameters:**index - the index of the parameter to be returned. **Returns:**the parameter at the specified index as a double value. **Throws:** [ClassCastException](http://docs.google.com/java/lang/ClassCastException.html) - if the parameter at the specified index is not a Double [NullPointerException](http://docs.google.com/java/lang/NullPointerException.html) - if the parameter at the specified index is null [ArrayIndexOutOfBoundsException](http://docs.google.com/java/lang/ArrayIndexOutOfBoundsException.html) - if index is negative or not less than the current size of this ParameterBlock object

### getParamClasses

public [Class](http://docs.google.com/java/lang/Class.html)[] **getParamClasses**()

Returns an array of Class objects describing the types of the parameters.

**Returns:**an array of Class objects.

| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/ParameterBlock.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/renderable/ContextualRenderedImageFactory.html)   [**NEXT CLASS**](http://docs.google.com/java/awt/image/renderable/RenderableImage.html) | [**FRAMES**](http://docs.google.com/index.html?java/awt/image/renderable/ParameterBlock.html)    [**NO FRAMES**](http://docs.google.com/ParameterBlock.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | [FIELD](#3znysh7) | [CONSTR](#2et92p0) | [METHOD](#tyjcwt) | DETAIL: [FIELD](#1t3h5sf) | [CONSTR](#17dp8vu) | [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).