| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/Composite.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/ComponentOrientation.html)   [**NEXT CLASS**](http://docs.google.com/java/awt/CompositeContext.html) | [**FRAMES**](http://docs.google.com/index.html?java/awt/Composite.html)    [**NO FRAMES**](http://docs.google.com/Composite.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | FIELD | CONSTR | [METHOD](#3znysh7) | DETAIL: FIELD | CONSTR | [METHOD](#2et92p0) |

## **java.awt**

Interface Composite

**All Known Implementing Classes:** [AlphaComposite](http://docs.google.com/java/awt/AlphaComposite.html)

public interface **Composite**

The Composite interface, along with [CompositeContext](http://docs.google.com/java/awt/CompositeContext.html), defines the methods to compose a draw primitive with the underlying graphics area. After the Composite is set in the [Graphics2D](http://docs.google.com/java/awt/Graphics2D.html) context, it combines a shape, text, or an image being rendered with the colors that have already been rendered according to pre-defined rules. The classes implementing this interface provide the rules and a method to create the context for a particular operation. CompositeContext is an environment used by the compositing operation, which is created by the Graphics2D prior to the start of the operation. CompositeContext contains private information and resources needed for a compositing operation. When the CompositeContext is no longer needed, the Graphics2D object disposes of it in order to reclaim resources allocated for the operation.

Instances of classes implementing Composite must be immutable because the Graphics2D does not clone these objects when they are set as an attribute with the setComposite method or when the Graphics2D object is cloned. This is to avoid undefined rendering behavior of Graphics2D, resulting from the modification of the Composite object after it has been set in the Graphics2D context.

Since this interface must expose the contents of pixels on the target device or image to potentially arbitrary code, the use of custom objects which implement this interface when rendering directly to a screen device is governed by the readDisplayPixels [AWTPermission](http://docs.google.com/java/awt/AWTPermission.html). The permission check will occur when such a custom object is passed to the setComposite method of a Graphics2D retrieved from a [Component](http://docs.google.com/java/awt/Component.html).

**See Also:**[AlphaComposite](http://docs.google.com/java/awt/AlphaComposite.html), [CompositeContext](http://docs.google.com/java/awt/CompositeContext.html), [Graphics2D.setComposite(java.awt.Composite)](http://docs.google.com/java/awt/Graphics2D.html#setComposite(java.awt.Composite))

| **Method Summary** | |
| --- | --- |
| [CompositeContext](http://docs.google.com/java/awt/CompositeContext.html) | [**createContext**](http://docs.google.com/java/awt/Composite.html#createContext(java.awt.image.ColorModel,%20java.awt.image.ColorModel,%20java.awt.RenderingHints))([ColorModel](http://docs.google.com/java/awt/image/ColorModel.html) srcColorModel, [ColorModel](http://docs.google.com/java/awt/image/ColorModel.html) dstColorModel, [RenderingHints](http://docs.google.com/java/awt/RenderingHints.html) hints)            Creates a context containing state that is used to perform the compositing operation. |

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

### createContext

[CompositeContext](http://docs.google.com/java/awt/CompositeContext.html) **createContext**([ColorModel](http://docs.google.com/java/awt/image/ColorModel.html) srcColorModel,  
 [ColorModel](http://docs.google.com/java/awt/image/ColorModel.html) dstColorModel,  
 [RenderingHints](http://docs.google.com/java/awt/RenderingHints.html) hints)

Creates a context containing state that is used to perform the compositing operation. In a multi-threaded environment, several contexts can exist simultaneously for a single Composite object.

**Parameters:**srcColorModel - the [ColorModel](http://docs.google.com/java/awt/image/ColorModel.html) of the sourcedstColorModel - the ColorModel of the destinationhints - the hint that the context object uses to choose between rendering alternatives **Returns:**the CompositeContext object used to perform the compositing operation.

| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/Composite.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/ComponentOrientation.html)   [**NEXT CLASS**](http://docs.google.com/java/awt/CompositeContext.html) | [**FRAMES**](http://docs.google.com/index.html?java/awt/Composite.html)    [**NO FRAMES**](http://docs.google.com/Composite.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | FIELD | CONSTR | [METHOD](#3znysh7) | DETAIL: FIELD | CONSTR | [METHOD](#2et92p0) |

[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).