| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/StyleContext.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/javax/swing/text/StyleConstants.ParagraphConstants.html)   [**NEXT CLASS**](http://docs.google.com/javax/swing/text/StyleContext.NamedStyle.html) | [**FRAMES**](http://docs.google.com/index.html?javax/swing/text/StyleContext.html)    [**NO FRAMES**](http://docs.google.com/StyleContext.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: [NESTED](#3znysh7) | [FIELD](#2et92p0) | [CONSTR](#tyjcwt) | [METHOD](#3dy6vkm) | DETAIL: [FIELD](#4d34og8) | [CONSTR](#17dp8vu) | [METHOD](#26in1rg) |

## **javax.swing.text**

Class StyleContext

[java.lang.Object](http://docs.google.com/java/lang/Object.html)  
 **javax.swing.text.StyleContext**

**All Implemented Interfaces:** [Serializable](http://docs.google.com/java/io/Serializable.html), [AbstractDocument.AttributeContext](http://docs.google.com/javax/swing/text/AbstractDocument.AttributeContext.html) **Direct Known Subclasses:** [StyleSheet](http://docs.google.com/javax/swing/text/html/StyleSheet.html)

public class **StyleContext**extends [Object](http://docs.google.com/java/lang/Object.html)implements [Serializable](http://docs.google.com/java/io/Serializable.html), [AbstractDocument.AttributeContext](http://docs.google.com/javax/swing/text/AbstractDocument.AttributeContext.html)

A pool of styles and their associated resources. This class determines the lifetime of a group of resources by being a container that holds caches for various resources such as font and color that get reused by the various style definitions. This can be shared by multiple documents if desired to maximize the sharing of related resources.

This class also provides efficient support for small sets of attributes and compresses them by sharing across uses and taking advantage of their immutable nature. Since many styles are replicated, the potential for sharing is significant, and copies can be extremely cheap. Larger sets reduce the possibility of sharing, and therefore revert automatically to a less space-efficient implementation.

**Warning:** Serialized objects of this class will not be compatible with future Swing releases. The current serialization support is appropriate for short term storage or RMI between applications running the same version of Swing. As of 1.4, support for long term storage of all JavaBeansTM has been added to the java.beans package. Please see [XMLEncoder](http://docs.google.com/java/beans/XMLEncoder.html).

| **Nested Class Summary** | |
| --- | --- |
| class | [**StyleContext.NamedStyle**](http://docs.google.com/javax/swing/text/StyleContext.NamedStyle.html)            A collection of attributes, typically used to represent character and paragraph styles. |
| class | [**StyleContext.SmallAttributeSet**](http://docs.google.com/javax/swing/text/StyleContext.SmallAttributeSet.html)            This class holds a small number of attributes in an array. |

| **Field Summary** | |
| --- | --- |
| static [String](http://docs.google.com/java/lang/String.html) | [**DEFAULT\_STYLE**](http://docs.google.com/javax/swing/text/StyleContext.html#DEFAULT_STYLE)            The name given to the default logical style attached to paragraphs. |

| **Constructor Summary** | |
| --- | --- |
| [**StyleContext**](http://docs.google.com/javax/swing/text/StyleContext.html#StyleContext())()            Creates a new StyleContext object. |

| **Method Summary** | |
| --- | --- |
| [AttributeSet](http://docs.google.com/javax/swing/text/AttributeSet.html) | [**addAttribute**](http://docs.google.com/javax/swing/text/StyleContext.html#addAttribute(javax.swing.text.AttributeSet,%20java.lang.Object,%20java.lang.Object))([AttributeSet](http://docs.google.com/javax/swing/text/AttributeSet.html) old, [Object](http://docs.google.com/java/lang/Object.html) name, [Object](http://docs.google.com/java/lang/Object.html) value)            Adds an attribute to the given set, and returns the new representative set. |
| [AttributeSet](http://docs.google.com/javax/swing/text/AttributeSet.html) | [**addAttributes**](http://docs.google.com/javax/swing/text/StyleContext.html#addAttributes(javax.swing.text.AttributeSet,%20javax.swing.text.AttributeSet))([AttributeSet](http://docs.google.com/javax/swing/text/AttributeSet.html) old, [AttributeSet](http://docs.google.com/javax/swing/text/AttributeSet.html) attr)            Adds a set of attributes to the element. |
| void | [**addChangeListener**](http://docs.google.com/javax/swing/text/StyleContext.html#addChangeListener(javax.swing.event.ChangeListener))([ChangeListener](http://docs.google.com/javax/swing/event/ChangeListener.html) l)            Adds a listener to track when styles are added or removed. |
| [Style](http://docs.google.com/javax/swing/text/Style.html) | [**addStyle**](http://docs.google.com/javax/swing/text/StyleContext.html#addStyle(java.lang.String,%20javax.swing.text.Style))([String](http://docs.google.com/java/lang/String.html) nm, [Style](http://docs.google.com/javax/swing/text/Style.html) parent)            Adds a new style into the style hierarchy. |
| protected  [MutableAttributeSet](http://docs.google.com/javax/swing/text/MutableAttributeSet.html) | [**createLargeAttributeSet**](http://docs.google.com/javax/swing/text/StyleContext.html#createLargeAttributeSet(javax.swing.text.AttributeSet))([AttributeSet](http://docs.google.com/javax/swing/text/AttributeSet.html) a)            Create a large set of attributes that should trade off space for time. |
| protected  [StyleContext.SmallAttributeSet](http://docs.google.com/javax/swing/text/StyleContext.SmallAttributeSet.html) | [**createSmallAttributeSet**](http://docs.google.com/javax/swing/text/StyleContext.html#createSmallAttributeSet(javax.swing.text.AttributeSet))([AttributeSet](http://docs.google.com/javax/swing/text/AttributeSet.html) a)            Create a compact set of attributes that might be shared. |
| [Color](http://docs.google.com/java/awt/Color.html) | [**getBackground**](http://docs.google.com/javax/swing/text/StyleContext.html#getBackground(javax.swing.text.AttributeSet))([AttributeSet](http://docs.google.com/javax/swing/text/AttributeSet.html) attr)            Takes a set of attributes and turn it into a background color specification. |
| [ChangeListener](http://docs.google.com/javax/swing/event/ChangeListener.html)[] | [**getChangeListeners**](http://docs.google.com/javax/swing/text/StyleContext.html#getChangeListeners())()            Returns an array of all the ChangeListeners added to this StyleContext with addChangeListener(). |
| protected  int | [**getCompressionThreshold**](http://docs.google.com/javax/swing/text/StyleContext.html#getCompressionThreshold())()            Returns the maximum number of key/value pairs to try and compress into unique/immutable sets. |
| static [StyleContext](http://docs.google.com/javax/swing/text/StyleContext.html) | [**getDefaultStyleContext**](http://docs.google.com/javax/swing/text/StyleContext.html#getDefaultStyleContext())()            Returns default AttributeContext shared by all documents that don't bother to define/supply their own context. |
| [AttributeSet](http://docs.google.com/javax/swing/text/AttributeSet.html) | [**getEmptySet**](http://docs.google.com/javax/swing/text/StyleContext.html#getEmptySet())()            Fetches an empty AttributeSet. |
| [Font](http://docs.google.com/java/awt/Font.html) | [**getFont**](http://docs.google.com/javax/swing/text/StyleContext.html#getFont(javax.swing.text.AttributeSet))([AttributeSet](http://docs.google.com/javax/swing/text/AttributeSet.html) attr)            Gets the font from an attribute set. |
| [Font](http://docs.google.com/java/awt/Font.html) | [**getFont**](http://docs.google.com/javax/swing/text/StyleContext.html#getFont(java.lang.String,%20int,%20int))([String](http://docs.google.com/java/lang/String.html) family, int style, int size)            Gets a new font. |
| [FontMetrics](http://docs.google.com/java/awt/FontMetrics.html) | [**getFontMetrics**](http://docs.google.com/javax/swing/text/StyleContext.html#getFontMetrics(java.awt.Font))([Font](http://docs.google.com/java/awt/Font.html) f)            Returns font metrics for a font. |
| [Color](http://docs.google.com/java/awt/Color.html) | [**getForeground**](http://docs.google.com/javax/swing/text/StyleContext.html#getForeground(javax.swing.text.AttributeSet))([AttributeSet](http://docs.google.com/javax/swing/text/AttributeSet.html) attr)            Takes a set of attributes and turn it into a foreground color specification. |
| static [Object](http://docs.google.com/java/lang/Object.html) | [**getStaticAttribute**](http://docs.google.com/javax/swing/text/StyleContext.html#getStaticAttribute(java.lang.Object))([Object](http://docs.google.com/java/lang/Object.html) key)            Returns the object previously registered with registerStaticAttributeKey. |
| static [Object](http://docs.google.com/java/lang/Object.html) | [**getStaticAttributeKey**](http://docs.google.com/javax/swing/text/StyleContext.html#getStaticAttributeKey(java.lang.Object))([Object](http://docs.google.com/java/lang/Object.html) key)            Returns the String that key will be registered with |
| [Style](http://docs.google.com/javax/swing/text/Style.html) | [**getStyle**](http://docs.google.com/javax/swing/text/StyleContext.html#getStyle(java.lang.String))([String](http://docs.google.com/java/lang/String.html) nm)            Fetches a named style previously added to the document |
| [Enumeration](http://docs.google.com/java/util/Enumeration.html)<?> | [**getStyleNames**](http://docs.google.com/javax/swing/text/StyleContext.html#getStyleNames())()            Fetches the names of the styles defined. |
| void | [**readAttributes**](http://docs.google.com/javax/swing/text/StyleContext.html#readAttributes(java.io.ObjectInputStream,%20javax.swing.text.MutableAttributeSet))([ObjectInputStream](http://docs.google.com/java/io/ObjectInputStream.html) in, [MutableAttributeSet](http://docs.google.com/javax/swing/text/MutableAttributeSet.html) a)            Context-specific handling of reading in attributes |
| static void | [**readAttributeSet**](http://docs.google.com/javax/swing/text/StyleContext.html#readAttributeSet(java.io.ObjectInputStream,%20javax.swing.text.MutableAttributeSet))([ObjectInputStream](http://docs.google.com/java/io/ObjectInputStream.html) in, [MutableAttributeSet](http://docs.google.com/javax/swing/text/MutableAttributeSet.html) a)            Reads a set of attributes from the given object input stream that have been previously written out with writeAttributeSet. |
| void | [**reclaim**](http://docs.google.com/javax/swing/text/StyleContext.html#reclaim(javax.swing.text.AttributeSet))([AttributeSet](http://docs.google.com/javax/swing/text/AttributeSet.html) a)            Returns a set no longer needed by the MutableAttributeSet implmentation. |
| static void | [**registerStaticAttributeKey**](http://docs.google.com/javax/swing/text/StyleContext.html#registerStaticAttributeKey(java.lang.Object))([Object](http://docs.google.com/java/lang/Object.html) key)            Registers an object as a static object that is being used as a key in attribute sets. |
| [AttributeSet](http://docs.google.com/javax/swing/text/AttributeSet.html) | [**removeAttribute**](http://docs.google.com/javax/swing/text/StyleContext.html#removeAttribute(javax.swing.text.AttributeSet,%20java.lang.Object))([AttributeSet](http://docs.google.com/javax/swing/text/AttributeSet.html) old, [Object](http://docs.google.com/java/lang/Object.html) name)            Removes an attribute from the set. |
| [AttributeSet](http://docs.google.com/javax/swing/text/AttributeSet.html) | [**removeAttributes**](http://docs.google.com/javax/swing/text/StyleContext.html#removeAttributes(javax.swing.text.AttributeSet,%20javax.swing.text.AttributeSet))([AttributeSet](http://docs.google.com/javax/swing/text/AttributeSet.html) old, [AttributeSet](http://docs.google.com/javax/swing/text/AttributeSet.html) attrs)            Removes a set of attributes for the element. |
| [AttributeSet](http://docs.google.com/javax/swing/text/AttributeSet.html) | [**removeAttributes**](http://docs.google.com/javax/swing/text/StyleContext.html#removeAttributes(javax.swing.text.AttributeSet,%20java.util.Enumeration))([AttributeSet](http://docs.google.com/javax/swing/text/AttributeSet.html) old, [Enumeration](http://docs.google.com/java/util/Enumeration.html)<?> names)            Removes a set of attributes for the element. |
| void | [**removeChangeListener**](http://docs.google.com/javax/swing/text/StyleContext.html#removeChangeListener(javax.swing.event.ChangeListener))([ChangeListener](http://docs.google.com/javax/swing/event/ChangeListener.html) l)            Removes a listener that was tracking styles being added or removed. |
| void | [**removeStyle**](http://docs.google.com/javax/swing/text/StyleContext.html#removeStyle(java.lang.String))([String](http://docs.google.com/java/lang/String.html) nm)            Removes a named style previously added to the document. |
| [String](http://docs.google.com/java/lang/String.html) | [**toString**](http://docs.google.com/javax/swing/text/StyleContext.html#toString())()            Converts a StyleContext to a String. |
| void | [**writeAttributes**](http://docs.google.com/javax/swing/text/StyleContext.html#writeAttributes(java.io.ObjectOutputStream,%20javax.swing.text.AttributeSet))([ObjectOutputStream](http://docs.google.com/java/io/ObjectOutputStream.html) out, [AttributeSet](http://docs.google.com/javax/swing/text/AttributeSet.html) a)            Context-specific handling of writing out attributes |
| static void | [**writeAttributeSet**](http://docs.google.com/javax/swing/text/StyleContext.html#writeAttributeSet(java.io.ObjectOutputStream,%20javax.swing.text.AttributeSet))([ObjectOutputStream](http://docs.google.com/java/io/ObjectOutputStream.html) out, [AttributeSet](http://docs.google.com/javax/swing/text/AttributeSet.html) a)            Writes a set of attributes to the given object stream for the purpose of serialization. |

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

### DEFAULT\_STYLE

public static final [String](http://docs.google.com/java/lang/String.html) **DEFAULT\_STYLE**

The name given to the default logical style attached to paragraphs.

**See Also:**[Constant Field Values](http://docs.google.com/constant-values.html#javax.swing.text.StyleContext.DEFAULT_STYLE)

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

### StyleContext

public **StyleContext**()

Creates a new StyleContext object.

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

### getDefaultStyleContext

public static final [StyleContext](http://docs.google.com/javax/swing/text/StyleContext.html) **getDefaultStyleContext**()

Returns default AttributeContext shared by all documents that don't bother to define/supply their own context.

**Returns:**the context

### addStyle

public [Style](http://docs.google.com/javax/swing/text/Style.html) **addStyle**([String](http://docs.google.com/java/lang/String.html) nm,  
 [Style](http://docs.google.com/javax/swing/text/Style.html) parent)

Adds a new style into the style hierarchy. Style attributes resolve from bottom up so an attribute specified in a child will override an attribute specified in the parent.

**Parameters:**nm - the name of the style (must be unique within the collection of named styles in the document). The name may be null if the style is unnamed, but the caller is responsible for managing the reference returned as an unnamed style can't be fetched by name. An unnamed style may be useful for things like character attribute overrides such as found in a style run.parent - the parent style. This may be null if unspecified attributes need not be resolved in some other style. **Returns:**the created style

### removeStyle

public void **removeStyle**([String](http://docs.google.com/java/lang/String.html) nm)

Removes a named style previously added to the document.

**Parameters:**nm - the name of the style to remove

### getStyle

public [Style](http://docs.google.com/javax/swing/text/Style.html) **getStyle**([String](http://docs.google.com/java/lang/String.html) nm)

Fetches a named style previously added to the document

**Parameters:**nm - the name of the style **Returns:**the style

### getStyleNames

public [Enumeration](http://docs.google.com/java/util/Enumeration.html)<?> **getStyleNames**()

Fetches the names of the styles defined.

**Returns:**the list of names as an enumeration

### addChangeListener

public void **addChangeListener**([ChangeListener](http://docs.google.com/javax/swing/event/ChangeListener.html) l)

Adds a listener to track when styles are added or removed.

**Parameters:**l - the change listener

### removeChangeListener

public void **removeChangeListener**([ChangeListener](http://docs.google.com/javax/swing/event/ChangeListener.html) l)

Removes a listener that was tracking styles being added or removed.

**Parameters:**l - the change listener

### getChangeListeners

public [ChangeListener](http://docs.google.com/javax/swing/event/ChangeListener.html)[] **getChangeListeners**()

Returns an array of all the ChangeListeners added to this StyleContext with addChangeListener().

**Returns:**all of the ChangeListeners added or an empty array if no listeners have been added**Since:** 1.4

### getFont

public [Font](http://docs.google.com/java/awt/Font.html) **getFont**([AttributeSet](http://docs.google.com/javax/swing/text/AttributeSet.html) attr)

Gets the font from an attribute set. This is implemented to try and fetch a cached font for the given AttributeSet, and if that fails the font features are resolved and the font is fetched from the low-level font cache.

**Parameters:**attr - the attribute set **Returns:**the font

### getForeground

public [Color](http://docs.google.com/java/awt/Color.html) **getForeground**([AttributeSet](http://docs.google.com/javax/swing/text/AttributeSet.html) attr)

Takes a set of attributes and turn it into a foreground color specification. This might be used to specify things like brighter, more hue, etc. By default it simply returns the value specified by the StyleConstants.Foreground attribute.

**Parameters:**attr - the set of attributes **Returns:**the color

### getBackground

public [Color](http://docs.google.com/java/awt/Color.html) **getBackground**([AttributeSet](http://docs.google.com/javax/swing/text/AttributeSet.html) attr)

Takes a set of attributes and turn it into a background color specification. This might be used to specify things like brighter, more hue, etc. By default it simply returns the value specified by the StyleConstants.Background attribute.

**Parameters:**attr - the set of attributes **Returns:**the color

### getFont

public [Font](http://docs.google.com/java/awt/Font.html) **getFont**([String](http://docs.google.com/java/lang/String.html) family,  
 int style,  
 int size)

Gets a new font. This returns a Font from a cache if a cached font exists. If not, a Font is added to the cache. This is basically a low-level cache for 1.1 font features.

**Parameters:**family - the font family (such as "Monospaced")style - the style of the font (such as Font.PLAIN)size - the point size >= 1 **Returns:**the new font

### getFontMetrics

public [FontMetrics](http://docs.google.com/java/awt/FontMetrics.html) **getFontMetrics**([Font](http://docs.google.com/java/awt/Font.html) f)

Returns font metrics for a font.

**Parameters:**f - the font **Returns:**the metrics

### addAttribute

public [AttributeSet](http://docs.google.com/javax/swing/text/AttributeSet.html) **addAttribute**([AttributeSet](http://docs.google.com/javax/swing/text/AttributeSet.html) old,  
 [Object](http://docs.google.com/java/lang/Object.html) name,  
 [Object](http://docs.google.com/java/lang/Object.html) value)

Adds an attribute to the given set, and returns the new representative set.

This method is thread safe, although most Swing methods are not. Please see [How to Use Threads](http://java.sun.com/docs/books/tutorial/uiswing/misc/threads.html) for more information.

**Specified by:**[addAttribute](http://docs.google.com/javax/swing/text/AbstractDocument.AttributeContext.html#addAttribute(javax.swing.text.AttributeSet,%20java.lang.Object,%20java.lang.Object)) in interface [AbstractDocument.AttributeContext](http://docs.google.com/javax/swing/text/AbstractDocument.AttributeContext.html) **Parameters:**old - the old attribute setname - the non-null attribute namevalue - the attribute value **Returns:**the updated attribute set**See Also:**[MutableAttributeSet.addAttribute(java.lang.Object, java.lang.Object)](http://docs.google.com/javax/swing/text/MutableAttributeSet.html#addAttribute(java.lang.Object,%20java.lang.Object))

### addAttributes

public [AttributeSet](http://docs.google.com/javax/swing/text/AttributeSet.html) **addAttributes**([AttributeSet](http://docs.google.com/javax/swing/text/AttributeSet.html) old,  
 [AttributeSet](http://docs.google.com/javax/swing/text/AttributeSet.html) attr)

Adds a set of attributes to the element.

This method is thread safe, although most Swing methods are not. Please see [How to Use Threads](http://java.sun.com/docs/books/tutorial/uiswing/misc/threads.html) for more information.

**Specified by:**[addAttributes](http://docs.google.com/javax/swing/text/AbstractDocument.AttributeContext.html#addAttributes(javax.swing.text.AttributeSet,%20javax.swing.text.AttributeSet)) in interface [AbstractDocument.AttributeContext](http://docs.google.com/javax/swing/text/AbstractDocument.AttributeContext.html) **Parameters:**old - the old attribute setattr - the attributes to add **Returns:**the updated attribute set**See Also:**[MutableAttributeSet.addAttribute(java.lang.Object, java.lang.Object)](http://docs.google.com/javax/swing/text/MutableAttributeSet.html#addAttribute(java.lang.Object,%20java.lang.Object))

### removeAttribute

public [AttributeSet](http://docs.google.com/javax/swing/text/AttributeSet.html) **removeAttribute**([AttributeSet](http://docs.google.com/javax/swing/text/AttributeSet.html) old,  
 [Object](http://docs.google.com/java/lang/Object.html) name)

Removes an attribute from the set.

This method is thread safe, although most Swing methods are not. Please see [How to Use Threads](http://java.sun.com/docs/books/tutorial/uiswing/misc/threads.html) for more information.

**Specified by:**[removeAttribute](http://docs.google.com/javax/swing/text/AbstractDocument.AttributeContext.html#removeAttribute(javax.swing.text.AttributeSet,%20java.lang.Object)) in interface [AbstractDocument.AttributeContext](http://docs.google.com/javax/swing/text/AbstractDocument.AttributeContext.html) **Parameters:**old - the old set of attributesname - the non-null attribute name **Returns:**the updated attribute set**See Also:**[MutableAttributeSet.removeAttribute(java.lang.Object)](http://docs.google.com/javax/swing/text/MutableAttributeSet.html#removeAttribute(java.lang.Object))

### removeAttributes

public [AttributeSet](http://docs.google.com/javax/swing/text/AttributeSet.html) **removeAttributes**([AttributeSet](http://docs.google.com/javax/swing/text/AttributeSet.html) old,  
 [Enumeration](http://docs.google.com/java/util/Enumeration.html)<?> names)

Removes a set of attributes for the element.

This method is thread safe, although most Swing methods are not. Please see [How to Use Threads](http://java.sun.com/docs/books/tutorial/uiswing/misc/threads.html) for more information.

**Specified by:**[removeAttributes](http://docs.google.com/javax/swing/text/AbstractDocument.AttributeContext.html#removeAttributes(javax.swing.text.AttributeSet,%20java.util.Enumeration)) in interface [AbstractDocument.AttributeContext](http://docs.google.com/javax/swing/text/AbstractDocument.AttributeContext.html) **Parameters:**old - the old attribute setnames - the attribute names **Returns:**the updated attribute set**See Also:**[MutableAttributeSet.removeAttributes(java.util.Enumeration)](http://docs.google.com/javax/swing/text/MutableAttributeSet.html#removeAttributes(java.util.Enumeration))

### removeAttributes

public [AttributeSet](http://docs.google.com/javax/swing/text/AttributeSet.html) **removeAttributes**([AttributeSet](http://docs.google.com/javax/swing/text/AttributeSet.html) old,  
 [AttributeSet](http://docs.google.com/javax/swing/text/AttributeSet.html) attrs)

Removes a set of attributes for the element.

This method is thread safe, although most Swing methods are not. Please see [How to Use Threads](http://java.sun.com/docs/books/tutorial/uiswing/misc/threads.html) for more information.

**Specified by:**[removeAttributes](http://docs.google.com/javax/swing/text/AbstractDocument.AttributeContext.html#removeAttributes(javax.swing.text.AttributeSet,%20javax.swing.text.AttributeSet)) in interface [AbstractDocument.AttributeContext](http://docs.google.com/javax/swing/text/AbstractDocument.AttributeContext.html) **Parameters:**old - the old attribute setattrs - the attributes **Returns:**the updated attribute set**See Also:**[MutableAttributeSet.removeAttributes(java.util.Enumeration)](http://docs.google.com/javax/swing/text/MutableAttributeSet.html#removeAttributes(java.util.Enumeration))

### getEmptySet

public [AttributeSet](http://docs.google.com/javax/swing/text/AttributeSet.html) **getEmptySet**()

Fetches an empty AttributeSet.

**Specified by:**[getEmptySet](http://docs.google.com/javax/swing/text/AbstractDocument.AttributeContext.html#getEmptySet()) in interface [AbstractDocument.AttributeContext](http://docs.google.com/javax/swing/text/AbstractDocument.AttributeContext.html) **Returns:**the set

### reclaim

public void **reclaim**([AttributeSet](http://docs.google.com/javax/swing/text/AttributeSet.html) a)

Returns a set no longer needed by the MutableAttributeSet implmentation. This is useful for operation under 1.1 where there are no weak references. This would typically be called by the finalize method of the MutableAttributeSet implementation.

This method is thread safe, although most Swing methods are not. Please see [How to Use Threads](http://java.sun.com/docs/books/tutorial/uiswing/misc/threads.html) for more information.

**Specified by:**[reclaim](http://docs.google.com/javax/swing/text/AbstractDocument.AttributeContext.html#reclaim(javax.swing.text.AttributeSet)) in interface [AbstractDocument.AttributeContext](http://docs.google.com/javax/swing/text/AbstractDocument.AttributeContext.html) **Parameters:**a - the set to reclaim

### getCompressionThreshold

protected int **getCompressionThreshold**()

Returns the maximum number of key/value pairs to try and compress into unique/immutable sets. Any sets above this limit will use hashtables and be a MutableAttributeSet.

**Returns:**the threshold

### createSmallAttributeSet

protected [StyleContext.SmallAttributeSet](http://docs.google.com/javax/swing/text/StyleContext.SmallAttributeSet.html) **createSmallAttributeSet**([AttributeSet](http://docs.google.com/javax/swing/text/AttributeSet.html) a)

Create a compact set of attributes that might be shared. This is a hook for subclasses that want to alter the behavior of SmallAttributeSet. This can be reimplemented to return an AttributeSet that provides some sort of attribute conversion.

**Parameters:**a - The set of attributes to be represented in the the compact form.

### createLargeAttributeSet

protected [MutableAttributeSet](http://docs.google.com/javax/swing/text/MutableAttributeSet.html) **createLargeAttributeSet**([AttributeSet](http://docs.google.com/javax/swing/text/AttributeSet.html) a)

Create a large set of attributes that should trade off space for time. This set will not be shared. This is a hook for subclasses that want to alter the behavior of the larger attribute storage format (which is SimpleAttributeSet by default). This can be reimplemented to return a MutableAttributeSet that provides some sort of attribute conversion.

**Parameters:**a - The set of attributes to be represented in the the larger form.

### toString

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

Converts a StyleContext to a String.

**Overrides:**[toString](http://docs.google.com/java/lang/Object.html#toString()) in class [Object](http://docs.google.com/java/lang/Object.html) **Returns:**the string

### writeAttributes

public void **writeAttributes**([ObjectOutputStream](http://docs.google.com/java/io/ObjectOutputStream.html) out,  
 [AttributeSet](http://docs.google.com/javax/swing/text/AttributeSet.html) a)  
 throws [IOException](http://docs.google.com/java/io/IOException.html)

Context-specific handling of writing out attributes

**Throws:** [IOException](http://docs.google.com/java/io/IOException.html)

### readAttributes

public void **readAttributes**([ObjectInputStream](http://docs.google.com/java/io/ObjectInputStream.html) in,  
 [MutableAttributeSet](http://docs.google.com/javax/swing/text/MutableAttributeSet.html) a)  
 throws [ClassNotFoundException](http://docs.google.com/java/lang/ClassNotFoundException.html),  
 [IOException](http://docs.google.com/java/io/IOException.html)

Context-specific handling of reading in attributes

**Throws:** [ClassNotFoundException](http://docs.google.com/java/lang/ClassNotFoundException.html) [IOException](http://docs.google.com/java/io/IOException.html)

### writeAttributeSet

public static void **writeAttributeSet**([ObjectOutputStream](http://docs.google.com/java/io/ObjectOutputStream.html) out,  
 [AttributeSet](http://docs.google.com/javax/swing/text/AttributeSet.html) a)  
 throws [IOException](http://docs.google.com/java/io/IOException.html)

Writes a set of attributes to the given object stream for the purpose of serialization. This will take special care to deal with static attribute keys that have been registered wit the registerStaticAttributeKey method. Any attribute key not regsitered as a static key will be serialized directly. All values are expected to be serializable.

**Parameters:**out - the output streama - the attribute set **Throws:** [IOException](http://docs.google.com/java/io/IOException.html) - on any I/O error

### readAttributeSet

public static void **readAttributeSet**([ObjectInputStream](http://docs.google.com/java/io/ObjectInputStream.html) in,  
 [MutableAttributeSet](http://docs.google.com/javax/swing/text/MutableAttributeSet.html) a)  
 throws [ClassNotFoundException](http://docs.google.com/java/lang/ClassNotFoundException.html),  
 [IOException](http://docs.google.com/java/io/IOException.html)

Reads a set of attributes from the given object input stream that have been previously written out with writeAttributeSet. This will try to restore keys that were static objects to the static objects in the current virtual machine considering only those keys that have been registered with the registerStaticAttributeKey method. The attributes retrieved from the stream will be placed into the given mutable set.

**Parameters:**in - the object stream to read the attribute data from.a - the attribute set to place the attribute definitions in. **Throws:** [ClassNotFoundException](http://docs.google.com/java/lang/ClassNotFoundException.html) - passed upward if encountered when reading the object stream. [IOException](http://docs.google.com/java/io/IOException.html) - passed upward if encountered when reading the object stream.

### registerStaticAttributeKey

public static void **registerStaticAttributeKey**([Object](http://docs.google.com/java/lang/Object.html) key)

Registers an object as a static object that is being used as a key in attribute sets. This allows the key to be treated specially for serialization.

For operation under a 1.1 virtual machine, this uses the value returned by toString concatenated to the classname. The value returned by toString should not have the class reference in it (ie it should be reimplemented from the definition in Object) in order to be the same when recomputed later.

**Parameters:**key - the non-null object key

### getStaticAttribute

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

Returns the object previously registered with registerStaticAttributeKey.

### getStaticAttributeKey

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

Returns the String that key will be registered with

**See Also:**[getStaticAttribute(java.lang.Object)](http://docs.google.com/javax/swing/text/StyleContext.html#getStaticAttribute(java.lang.Object)), [registerStaticAttributeKey(java.lang.Object)](http://docs.google.com/javax/swing/text/StyleContext.html#registerStaticAttributeKey(java.lang.Object))

| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/StyleContext.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/javax/swing/text/StyleConstants.ParagraphConstants.html)   [**NEXT CLASS**](http://docs.google.com/javax/swing/text/StyleContext.NamedStyle.html) | [**FRAMES**](http://docs.google.com/index.html?javax/swing/text/StyleContext.html)    [**NO FRAMES**](http://docs.google.com/StyleContext.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: [NESTED](#3znysh7) | [FIELD](#2et92p0) | [CONSTR](#tyjcwt) | [METHOD](#3dy6vkm) | DETAIL: [FIELD](#4d34og8) | [CONSTR](#17dp8vu) | [METHOD](#26in1rg) |

[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).