| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/DefaultFormatter.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/DefaultEditorKit.PasteAction.html)   [**NEXT CLASS**](http://docs.google.com/javax/swing/text/DefaultFormatterFactory.html) | [**FRAMES**](http://docs.google.com/index.html?javax/swing/text/DefaultFormatter.html)    [**NO FRAMES**](http://docs.google.com/DefaultFormatter.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | FIELD | [CONSTR](#3znysh7) | [METHOD](#2et92p0) | DETAIL: FIELD | [CONSTR](#1t3h5sf) | [METHOD](#2s8eyo1) |

## **javax.swing.text**

Class DefaultFormatter

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

**All Implemented Interfaces:** [Serializable](http://docs.google.com/java/io/Serializable.html), [Cloneable](http://docs.google.com/java/lang/Cloneable.html) **Direct Known Subclasses:** [InternationalFormatter](http://docs.google.com/javax/swing/text/InternationalFormatter.html), [MaskFormatter](http://docs.google.com/javax/swing/text/MaskFormatter.html)

public class **DefaultFormatter**extends [JFormattedTextField.AbstractFormatter](http://docs.google.com/javax/swing/JFormattedTextField.AbstractFormatter.html)implements [Cloneable](http://docs.google.com/java/lang/Cloneable.html), [Serializable](http://docs.google.com/java/io/Serializable.html)

DefaultFormatter formats aribtrary objects. Formatting is done by invoking the toString method. In order to convert the value back to a String, your class must provide a constructor that takes a String argument. If no single argument constructor that takes a String is found, the returned value will be the String passed into stringToValue.

Instances of DefaultFormatter can not be used in multiple instances of JFormattedTextField. To obtain a copy of an already configured DefaultFormatter, use the clone method.

**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).

**Since:** 1.4 **See Also:**[JFormattedTextField.AbstractFormatter](http://docs.google.com/javax/swing/JFormattedTextField.AbstractFormatter.html)

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

| **Method Summary** | |
| --- | --- |
| [Object](http://docs.google.com/java/lang/Object.html) | [**clone**](http://docs.google.com/javax/swing/text/DefaultFormatter.html#clone())()            Creates a copy of the DefaultFormatter. |
| boolean | [**getAllowsInvalid**](http://docs.google.com/javax/swing/text/DefaultFormatter.html#getAllowsInvalid())()            Returns whether or not the value being edited is allowed to be invalid for a length of time. |
| boolean | [**getCommitsOnValidEdit**](http://docs.google.com/javax/swing/text/DefaultFormatter.html#getCommitsOnValidEdit())()            Returns when edits are published back to the JFormattedTextField. |
| protected  [DocumentFilter](http://docs.google.com/javax/swing/text/DocumentFilter.html) | [**getDocumentFilter**](http://docs.google.com/javax/swing/text/DefaultFormatter.html#getDocumentFilter())()            Returns the DocumentFilter used to restrict the characters that can be input into the JFormattedTextField. |
| protected  [NavigationFilter](http://docs.google.com/javax/swing/text/NavigationFilter.html) | [**getNavigationFilter**](http://docs.google.com/javax/swing/text/DefaultFormatter.html#getNavigationFilter())()            Returns the NavigationFilter used to restrict where the cursor can be placed. |
| boolean | [**getOverwriteMode**](http://docs.google.com/javax/swing/text/DefaultFormatter.html#getOverwriteMode())()            Returns the behavior when inserting characters. |
| [Class](http://docs.google.com/java/lang/Class.html)<?> | [**getValueClass**](http://docs.google.com/javax/swing/text/DefaultFormatter.html#getValueClass())()            Returns that class that is used to create new Objects. |
| void | [**install**](http://docs.google.com/javax/swing/text/DefaultFormatter.html#install(javax.swing.JFormattedTextField))([JFormattedTextField](http://docs.google.com/javax/swing/JFormattedTextField.html) ftf)            Installs the DefaultFormatter onto a particular JFormattedTextField. |
| void | [**setAllowsInvalid**](http://docs.google.com/javax/swing/text/DefaultFormatter.html#setAllowsInvalid(boolean))(boolean allowsInvalid)            Sets whether or not the value being edited is allowed to be invalid for a length of time (that is, stringToValue throws a ParseException). |
| void | [**setCommitsOnValidEdit**](http://docs.google.com/javax/swing/text/DefaultFormatter.html#setCommitsOnValidEdit(boolean))(boolean commit)            Sets when edits are published back to the JFormattedTextField. |
| void | [**setOverwriteMode**](http://docs.google.com/javax/swing/text/DefaultFormatter.html#setOverwriteMode(boolean))(boolean overwriteMode)            Configures the behavior when inserting characters. |
| void | [**setValueClass**](http://docs.google.com/javax/swing/text/DefaultFormatter.html#setValueClass(java.lang.Class))([Class](http://docs.google.com/java/lang/Class.html)<?> valueClass)            Sets that class that is used to create new Objects. |
| [Object](http://docs.google.com/java/lang/Object.html) | [**stringToValue**](http://docs.google.com/javax/swing/text/DefaultFormatter.html#stringToValue(java.lang.String))([String](http://docs.google.com/java/lang/String.html) string)            Converts the passed in String into an instance of getValueClass by way of the constructor that takes a String argument. |
| [String](http://docs.google.com/java/lang/String.html) | [**valueToString**](http://docs.google.com/javax/swing/text/DefaultFormatter.html#valueToString(java.lang.Object))([Object](http://docs.google.com/java/lang/Object.html) value)            Converts the passed in Object into a String by way of the toString method. |

| **Methods inherited from class javax.swing.**[**JFormattedTextField.AbstractFormatter**](http://docs.google.com/javax/swing/JFormattedTextField.AbstractFormatter.html) |
| --- |
| [getActions](http://docs.google.com/javax/swing/JFormattedTextField.AbstractFormatter.html#getActions()), [getFormattedTextField](http://docs.google.com/javax/swing/JFormattedTextField.AbstractFormatter.html#getFormattedTextField()), [invalidEdit](http://docs.google.com/javax/swing/JFormattedTextField.AbstractFormatter.html#invalidEdit()), [setEditValid](http://docs.google.com/javax/swing/JFormattedTextField.AbstractFormatter.html#setEditValid(boolean)), [uninstall](http://docs.google.com/javax/swing/JFormattedTextField.AbstractFormatter.html#uninstall()) |

| **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)) |

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

### DefaultFormatter

public **DefaultFormatter**()

Creates a DefaultFormatter.

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

### install

public void **install**([JFormattedTextField](http://docs.google.com/javax/swing/JFormattedTextField.html) ftf)

Installs the DefaultFormatter onto a particular JFormattedTextField. This will invoke valueToString to convert the current value from the JFormattedTextField to a String. This will then install the Actions from getActions, the DocumentFilter returned from getDocumentFilter and the NavigationFilter returned from getNavigationFilter onto the JFormattedTextField.

Subclasses will typically only need to override this if they wish to install additional listeners on the JFormattedTextField.

If there is a ParseException in converting the current value to a String, this will set the text to an empty String, and mark the JFormattedTextField as being in an invalid state.

While this is a public method, this is typically only useful for subclassers of JFormattedTextField. JFormattedTextField will invoke this method at the appropriate times when the value changes, or its internal state changes.

**Overrides:**[install](http://docs.google.com/javax/swing/JFormattedTextField.AbstractFormatter.html#install(javax.swing.JFormattedTextField)) in class [JFormattedTextField.AbstractFormatter](http://docs.google.com/javax/swing/JFormattedTextField.AbstractFormatter.html) **Parameters:**ftf - JFormattedTextField to format for, may be null indicating uninstall from current JFormattedTextField.

### setCommitsOnValidEdit

public void **setCommitsOnValidEdit**(boolean commit)

Sets when edits are published back to the JFormattedTextField. If true, commitEdit is invoked after every valid edit (any time the text is edited). On the other hand, if this is false than the DefaultFormatter does not publish edits back to the JFormattedTextField. As such, the only time the value of the JFormattedTextField will change is when commitEdit is invoked on JFormattedTextField, typically when enter is pressed or focus leaves the JFormattedTextField.

**Parameters:**commit - Used to indicate when edits are commited back to the JTextComponent

### getCommitsOnValidEdit

public boolean **getCommitsOnValidEdit**()

Returns when edits are published back to the JFormattedTextField.

**Returns:**true if edits are commited after evey valid edit

### setOverwriteMode

public void **setOverwriteMode**(boolean overwriteMode)

Configures the behavior when inserting characters. If overwriteMode is true (the default), new characters overwrite existing characters in the model.

**Parameters:**overwriteMode - Indicates if overwrite or overstrike mode is used

### getOverwriteMode

public boolean **getOverwriteMode**()

Returns the behavior when inserting characters.

**Returns:**true if newly inserted characters overwrite existing characters

### setAllowsInvalid

public void **setAllowsInvalid**(boolean allowsInvalid)

Sets whether or not the value being edited is allowed to be invalid for a length of time (that is, stringToValue throws a ParseException). It is often convenient to allow the user to temporarily input an invalid value.

**Parameters:**allowsInvalid - Used to indicate if the edited value must always be valid

### getAllowsInvalid

public boolean **getAllowsInvalid**()

Returns whether or not the value being edited is allowed to be invalid for a length of time.

**Returns:**false if the edited value must always be valid

### setValueClass

public void **setValueClass**([Class](http://docs.google.com/java/lang/Class.html)<?> valueClass)

Sets that class that is used to create new Objects. If the passed in class does not have a single argument constructor that takes a String, String values will be used.

**Parameters:**valueClass - Class used to construct return value from stringToValue

### getValueClass

public [Class](http://docs.google.com/java/lang/Class.html)<?> **getValueClass**()

Returns that class that is used to create new Objects.

**Returns:**Class used to constuct return value from stringToValue

### stringToValue

public [Object](http://docs.google.com/java/lang/Object.html) **stringToValue**([String](http://docs.google.com/java/lang/String.html) string)  
 throws [ParseException](http://docs.google.com/java/text/ParseException.html)

Converts the passed in String into an instance of getValueClass by way of the constructor that takes a String argument. If getValueClass returns null, the Class of the current value in the JFormattedTextField will be used. If this is null, a String will be returned. If the constructor thows an exception, a ParseException will be thrown. If there is no single argument String constructor, string will be returned.

**Specified by:**[stringToValue](http://docs.google.com/javax/swing/JFormattedTextField.AbstractFormatter.html#stringToValue(java.lang.String)) in class [JFormattedTextField.AbstractFormatter](http://docs.google.com/javax/swing/JFormattedTextField.AbstractFormatter.html) **Parameters:**string - String to convert **Returns:**Object representation of text **Throws:** [ParseException](http://docs.google.com/java/text/ParseException.html) - if there is an error in the conversion

### valueToString

public [String](http://docs.google.com/java/lang/String.html) **valueToString**([Object](http://docs.google.com/java/lang/Object.html) value)  
 throws [ParseException](http://docs.google.com/java/text/ParseException.html)

Converts the passed in Object into a String by way of the toString method.

**Specified by:**[valueToString](http://docs.google.com/javax/swing/JFormattedTextField.AbstractFormatter.html#valueToString(java.lang.Object)) in class [JFormattedTextField.AbstractFormatter](http://docs.google.com/javax/swing/JFormattedTextField.AbstractFormatter.html) **Parameters:**value - Value to convert **Returns:**String representation of value **Throws:** [ParseException](http://docs.google.com/java/text/ParseException.html) - if there is an error in the conversion

### getDocumentFilter

protected [DocumentFilter](http://docs.google.com/javax/swing/text/DocumentFilter.html) **getDocumentFilter**()

Returns the DocumentFilter used to restrict the characters that can be input into the JFormattedTextField.

**Overrides:**[getDocumentFilter](http://docs.google.com/javax/swing/JFormattedTextField.AbstractFormatter.html#getDocumentFilter()) in class [JFormattedTextField.AbstractFormatter](http://docs.google.com/javax/swing/JFormattedTextField.AbstractFormatter.html) **Returns:**DocumentFilter to restrict edits

### getNavigationFilter

protected [NavigationFilter](http://docs.google.com/javax/swing/text/NavigationFilter.html) **getNavigationFilter**()

Returns the NavigationFilter used to restrict where the cursor can be placed.

**Overrides:**[getNavigationFilter](http://docs.google.com/javax/swing/JFormattedTextField.AbstractFormatter.html#getNavigationFilter()) in class [JFormattedTextField.AbstractFormatter](http://docs.google.com/javax/swing/JFormattedTextField.AbstractFormatter.html) **Returns:**NavigationFilter to restrict navigation

### clone

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

Creates a copy of the DefaultFormatter.

**Overrides:**[clone](http://docs.google.com/javax/swing/JFormattedTextField.AbstractFormatter.html#clone()) in class [JFormattedTextField.AbstractFormatter](http://docs.google.com/javax/swing/JFormattedTextField.AbstractFormatter.html) **Returns:**copy of the DefaultFormatter **Throws:** [CloneNotSupportedException](http://docs.google.com/java/lang/CloneNotSupportedException.html) - if the object's class does not support the Cloneable interface. Subclasses that override the clone method can also throw this exception to indicate that an instance cannot be cloned.**See Also:**[Cloneable](http://docs.google.com/java/lang/Cloneable.html)

| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/DefaultFormatter.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/DefaultEditorKit.PasteAction.html)   [**NEXT CLASS**](http://docs.google.com/javax/swing/text/DefaultFormatterFactory.html) | [**FRAMES**](http://docs.google.com/index.html?javax/swing/text/DefaultFormatter.html)    [**NO FRAMES**](http://docs.google.com/DefaultFormatter.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | FIELD | [CONSTR](#3znysh7) | [METHOD](#2et92p0) | DETAIL: FIELD | [CONSTR](#1t3h5sf) | [METHOD](#2s8eyo1) |

[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).