| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/InternationalFormatter.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/IconView.html)   [**NEXT CLASS**](http://docs.google.com/javax/swing/text/JTextComponent.html) | [**FRAMES**](http://docs.google.com/index.html?javax/swing/text/InternationalFormatter.html)    [**NO FRAMES**](http://docs.google.com/InternationalFormatter.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | FIELD | [CONSTR](#3znysh7) | [METHOD](#2et92p0) | DETAIL: FIELD | [CONSTR](#4d34og8) | [METHOD](#3rdcrjn) |

## **javax.swing.text**

Class InternationalFormatter

[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](http://docs.google.com/javax/swing/text/DefaultFormatter.html)  
 **javax.swing.text.InternationalFormatter**

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

public class **InternationalFormatter**extends [DefaultFormatter](http://docs.google.com/javax/swing/text/DefaultFormatter.html)

InternationalFormatter extends DefaultFormatter, using an instance of java.text.Format to handle the conversion to a String, and the conversion from a String.

If getAllowsInvalid() is false, this will ask the Format to format the current text on every edit.

You can specify a minimum and maximum value by way of the setMinimum and setMaximum methods. In order for this to work the values returned from stringToValue must be comparable to the min/max values by way of the Comparable interface.

Be careful how you configure the Format and the InternationalFormatter, as it is possible to create a situation where certain values can not be input. Consider the date format 'M/d/yy', an InternationalFormatter that is always valid (setAllowsInvalid(false)), is in overwrite mode (setOverwriteMode(true)) and the date 7/1/99. In this case the user will not be able to enter a two digit month or day of month. To avoid this, the format should be 'MM/dd/yy'.

If InternationalFormatter is configured to only allow valid values (setAllowsInvalid(false)), every valid edit will result in the text of the JFormattedTextField being completely reset from the Format. The cursor position will also be adjusted as literal characters are added/removed from the resulting String.

InternationalFormatter's behavior of stringToValue is slightly different than that of DefaultTextFormatter, it does the following:

1. parseObject is invoked on the Format specified by setFormat
2. If a Class has been set for the values (setValueClass), supers implementation is invoked to convert the value returned from parseObject to the appropriate class.
3. If a ParseException has not been thrown, and the value is outside the min/max a ParseException is thrown.
4. The value is returned.

InternationalFormatter implements stringToValue in this manner so that you can specify an alternate Class than Format may return.

**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:**[Format](http://docs.google.com/java/text/Format.html), [Comparable](http://docs.google.com/java/lang/Comparable.html)

| **Constructor Summary** | |
| --- | --- |
| [**InternationalFormatter**](http://docs.google.com/javax/swing/text/InternationalFormatter.html#InternationalFormatter())()            Creates an InternationalFormatter with no Format specified. |
| [**InternationalFormatter**](http://docs.google.com/javax/swing/text/InternationalFormatter.html#InternationalFormatter(java.text.Format))([Format](http://docs.google.com/java/text/Format.html) format)            Creates an InternationalFormatter with the specified Format instance. |

| **Method Summary** | |
| --- | --- |
| [Object](http://docs.google.com/java/lang/Object.html) | [**clone**](http://docs.google.com/javax/swing/text/InternationalFormatter.html#clone())()            Creates a copy of the DefaultFormatter. |
| protected  [Action](http://docs.google.com/javax/swing/Action.html)[] | [**getActions**](http://docs.google.com/javax/swing/text/InternationalFormatter.html#getActions())()            If getSupportsIncrement returns true, this returns two Actions suitable for incrementing/decrementing the value. |
| [Format.Field](http://docs.google.com/java/text/Format.Field.html)[] | [**getFields**](http://docs.google.com/javax/swing/text/InternationalFormatter.html#getFields(int))(int offset)            Returns the Format.Field constants associated with the text at offset. |
| [Format](http://docs.google.com/java/text/Format.html) | [**getFormat**](http://docs.google.com/javax/swing/text/InternationalFormatter.html#getFormat())()            Returns the format that dictates the legal values that can be edited and displayed. |
| [Comparable](http://docs.google.com/java/lang/Comparable.html) | [**getMaximum**](http://docs.google.com/javax/swing/text/InternationalFormatter.html#getMaximum())()            Returns the maximum permissible value. |
| [Comparable](http://docs.google.com/java/lang/Comparable.html) | [**getMinimum**](http://docs.google.com/javax/swing/text/InternationalFormatter.html#getMinimum())()            Returns the minimum permissible value. |
| void | [**install**](http://docs.google.com/javax/swing/text/InternationalFormatter.html#install(javax.swing.JFormattedTextField))([JFormattedTextField](http://docs.google.com/javax/swing/JFormattedTextField.html) ftf)            Installs the DefaultFormatter onto a particular JFormattedTextField. |
| void | [**setFormat**](http://docs.google.com/javax/swing/text/InternationalFormatter.html#setFormat(java.text.Format))([Format](http://docs.google.com/java/text/Format.html) format)            Sets the format that dictates the legal values that can be edited and displayed. |
| void | [**setMaximum**](http://docs.google.com/javax/swing/text/InternationalFormatter.html#setMaximum(java.lang.Comparable))([Comparable](http://docs.google.com/java/lang/Comparable.html) max)            Sets the maximum permissible value. |
| void | [**setMinimum**](http://docs.google.com/javax/swing/text/InternationalFormatter.html#setMinimum(java.lang.Comparable))([Comparable](http://docs.google.com/java/lang/Comparable.html) minimum)            Sets the minimum permissible value. |
| [Object](http://docs.google.com/java/lang/Object.html) | [**stringToValue**](http://docs.google.com/javax/swing/text/InternationalFormatter.html#stringToValue(java.lang.String))([String](http://docs.google.com/java/lang/String.html) text)            Returns the Object representation of the String text. |
| [String](http://docs.google.com/java/lang/String.html) | [**valueToString**](http://docs.google.com/javax/swing/text/InternationalFormatter.html#valueToString(java.lang.Object))([Object](http://docs.google.com/java/lang/Object.html) value)            Returns a String representation of the Object value. |

| **Methods inherited from class javax.swing.text.**[**DefaultFormatter**](http://docs.google.com/javax/swing/text/DefaultFormatter.html) |
| --- |
| [getAllowsInvalid](http://docs.google.com/javax/swing/text/DefaultFormatter.html#getAllowsInvalid()), [getCommitsOnValidEdit](http://docs.google.com/javax/swing/text/DefaultFormatter.html#getCommitsOnValidEdit()), [getDocumentFilter](http://docs.google.com/javax/swing/text/DefaultFormatter.html#getDocumentFilter()), [getNavigationFilter](http://docs.google.com/javax/swing/text/DefaultFormatter.html#getNavigationFilter()), [getOverwriteMode](http://docs.google.com/javax/swing/text/DefaultFormatter.html#getOverwriteMode()), [getValueClass](http://docs.google.com/javax/swing/text/DefaultFormatter.html#getValueClass()), [setAllowsInvalid](http://docs.google.com/javax/swing/text/DefaultFormatter.html#setAllowsInvalid(boolean)), [setCommitsOnValidEdit](http://docs.google.com/javax/swing/text/DefaultFormatter.html#setCommitsOnValidEdit(boolean)), [setOverwriteMode](http://docs.google.com/javax/swing/text/DefaultFormatter.html#setOverwriteMode(boolean)), [setValueClass](http://docs.google.com/javax/swing/text/DefaultFormatter.html#setValueClass(java.lang.Class)) |

| **Methods inherited from class javax.swing.**[**JFormattedTextField.AbstractFormatter**](http://docs.google.com/javax/swing/JFormattedTextField.AbstractFormatter.html) |
| --- |
| [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** |
| --- |

### InternationalFormatter

public **InternationalFormatter**()

Creates an InternationalFormatter with no Format specified.

### InternationalFormatter

public **InternationalFormatter**([Format](http://docs.google.com/java/text/Format.html) format)

Creates an InternationalFormatter with the specified Format instance.

**Parameters:**format - Format instance used for converting from/to Strings

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

### setFormat

public void **setFormat**([Format](http://docs.google.com/java/text/Format.html) format)

Sets the format that dictates the legal values that can be edited and displayed.

**Parameters:**format - Format instance used for converting from/to Strings

### getFormat

public [Format](http://docs.google.com/java/text/Format.html) **getFormat**()

Returns the format that dictates the legal values that can be edited and displayed.

**Returns:**Format instance used for converting from/to Strings

### setMinimum

public void **setMinimum**([Comparable](http://docs.google.com/java/lang/Comparable.html) minimum)

Sets the minimum permissible value. If the valueClass has not been specified, and minimum is non null, the valueClass will be set to that of the class of minimum.

**Parameters:**minimum - Minimum legal value that can be input**See Also:**[DefaultFormatter.setValueClass(java.lang.Class)](http://docs.google.com/javax/swing/text/DefaultFormatter.html#setValueClass(java.lang.Class))

### getMinimum

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

Returns the minimum permissible value.

**Returns:**Minimum legal value that can be input

### setMaximum

public void **setMaximum**([Comparable](http://docs.google.com/java/lang/Comparable.html) max)

Sets the maximum permissible value. If the valueClass has not been specified, and max is non null, the valueClass will be set to that of the class of max.

**Parameters:**max - Maximum legal value that can be input**See Also:**[DefaultFormatter.setValueClass(java.lang.Class)](http://docs.google.com/javax/swing/text/DefaultFormatter.html#setValueClass(java.lang.Class))

### getMaximum

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

Returns the maximum permissible value.

**Returns:**Maximum legal value that can be input

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

### 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)

Returns a String representation of the Object value. This invokes format on the current Format.

**Overrides:**[valueToString](http://docs.google.com/javax/swing/text/DefaultFormatter.html#valueToString(java.lang.Object)) in class [DefaultFormatter](http://docs.google.com/javax/swing/text/DefaultFormatter.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

### stringToValue

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

Returns the Object representation of the String text.

**Overrides:**[stringToValue](http://docs.google.com/javax/swing/text/DefaultFormatter.html#stringToValue(java.lang.String)) in class [DefaultFormatter](http://docs.google.com/javax/swing/text/DefaultFormatter.html) **Parameters:**text - 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

### getFields

public [Format.Field](http://docs.google.com/java/text/Format.Field.html)[] **getFields**(int offset)

Returns the Format.Field constants associated with the text at offset. If offset is not a valid location into the current text, this will return an empty array.

**Parameters:**offset - offset into text to be examined **Returns:**Format.Field constants associated with the text at the given position.

### 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/text/DefaultFormatter.html#clone()) in class [DefaultFormatter](http://docs.google.com/javax/swing/text/DefaultFormatter.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)

### getActions

protected [Action](http://docs.google.com/javax/swing/Action.html)[] **getActions**()

If getSupportsIncrement returns true, this returns two Actions suitable for incrementing/decrementing the value.

**Overrides:**[getActions](http://docs.google.com/javax/swing/JFormattedTextField.AbstractFormatter.html#getActions()) in class [JFormattedTextField.AbstractFormatter](http://docs.google.com/javax/swing/JFormattedTextField.AbstractFormatter.html) **Returns:**Array of Actions to install on JFormattedTextField

| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/InternationalFormatter.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/IconView.html)   [**NEXT CLASS**](http://docs.google.com/javax/swing/text/JTextComponent.html) | [**FRAMES**](http://docs.google.com/index.html?javax/swing/text/InternationalFormatter.html)    [**NO FRAMES**](http://docs.google.com/InternationalFormatter.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | FIELD | [CONSTR](#3znysh7) | [METHOD](#2et92p0) | DETAIL: FIELD | [CONSTR](#4d34og8) | [METHOD](#3rdcrjn) |

[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).