| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/RuleBasedCollator.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/text/ParsePosition.html)   [**NEXT CLASS**](http://docs.google.com/java/text/SimpleDateFormat.html) | [**FRAMES**](http://docs.google.com/index.html?java/text/RuleBasedCollator.html)    [**NO FRAMES**](http://docs.google.com/RuleBasedCollator.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | [FIELD](#2et92p0) | [CONSTR](#tyjcwt) | [METHOD](#3dy6vkm) | DETAIL: FIELD | [CONSTR](#2s8eyo1) | [METHOD](#3rdcrjn) |

## **java.text**

Class RuleBasedCollator

[java.lang.Object](http://docs.google.com/java/lang/Object.html)  
 [java.text.Collator](http://docs.google.com/java/text/Collator.html)  
 **java.text.RuleBasedCollator**

**All Implemented Interfaces:** [Cloneable](http://docs.google.com/java/lang/Cloneable.html), [Comparator](http://docs.google.com/java/util/Comparator.html)<[Object](http://docs.google.com/java/lang/Object.html)>

public class **RuleBasedCollator**extends [Collator](http://docs.google.com/java/text/Collator.html)

The RuleBasedCollator class is a concrete subclass of Collator that provides a simple, data-driven, table collator. With this class you can create a customized table-based Collator. RuleBasedCollator maps characters to sort keys.

RuleBasedCollator has the following restrictions for efficiency (other subclasses may be used for more complex languages) :

1. If a special collation rule controlled by a <modifier> is specified it applies to the whole collator object.
2. All non-mentioned characters are at the end of the collation order.

The collation table is composed of a list of collation rules, where each rule is of one of three forms:

<modifier>  
 <relation> <text-argument>  
 <reset> <text-argument>

The definitions of the rule elements is as follows:

* **Text-Argument**: A text-argument is any sequence of characters, excluding special characters (that is, common whitespace characters [0009-000D, 0020] and rule syntax characters [0021-002F, 003A-0040, 005B-0060, 007B-007E]). If those characters are desired, you can put them in single quotes (e.g. ampersand => '&'). Note that unquoted white space characters are ignored; e.g. b c is treated as bc.
* **Modifier**: There are currently two modifiers that turn on special collation rules.
  + '@' : Turns on backwards sorting of accents (secondary differences), as in French.
  + '!' : Turns on Thai/Lao vowel-consonant swapping. If this rule is in force when a Thai vowel of the range \U0E40-\U0E44 precedes a Thai consonant of the range \U0E01-\U0E2E OR a Lao vowel of the range \U0EC0-\U0EC4 precedes a Lao consonant of the range \U0E81-\U0EAE then the vowel is placed after the consonant for collation purposes.

'@' : Indicates that accents are sorted backwards, as in French.

* **Relation**: The relations are the following:
  + '<' : Greater, as a letter difference (primary)
  + ';' : Greater, as an accent difference (secondary)
  + ',' : Greater, as a case difference (tertiary)
  + '=' : Equal
* **Reset**: There is a single reset which is used primarily for contractions and expansions, but which can also be used to add a modification at the end of a set of rules.  
  '&' : Indicates that the next rule follows the position to where the reset text-argument would be sorted.

This sounds more complicated than it is in practice. For example, the following are equivalent ways of expressing the same thing:

a < b < c  
 a < b & b < c  
 a < c & a < b

Notice that the order is important, as the subsequent item goes immediately after the text-argument. The following are not equivalent:

a < b & a < c  
 a < c & a < b

Either the text-argument must already be present in the sequence, or some initial substring of the text-argument must be present. (e.g. "a < b & ae < e" is valid since "a" is present in the sequence before "ae" is reset). In this latter case, "ae" is not entered and treated as a single character; instead, "e" is sorted as if it were expanded to two characters: "a" followed by an "e". This difference appears in natural languages: in traditional Spanish "ch" is treated as though it contracts to a single character (expressed as "c < ch < d"), while in traditional German a-umlaut is treated as though it expanded to two characters (expressed as "a,A < b,B ... &ae;\u00e3&AE;\u00c3"). [\u00e3 and \u00c3 are, of course, the escape sequences for a-umlaut.]

**Ignorable Characters**

For ignorable characters, the first rule must start with a relation (the examples we have used above are really fragments; "a < b" really should be "< a < b"). If, however, the first relation is not "<", then all the all text-arguments up to the first "<" are ignorable. For example, ", - < a < b" makes "-" an ignorable character, as we saw earlier in the word "black-birds". In the samples for different languages, you see that most accents are ignorable.

**Normalization and Accents**

RuleBasedCollator automatically processes its rule table to include both pre-composed and combining-character versions of accented characters. Even if the provided rule string contains only base characters and separate combining accent characters, the pre-composed accented characters matching all canonical combinations of characters from the rule string will be entered in the table.

This allows you to use a RuleBasedCollator to compare accented strings even when the collator is set to NO\_DECOMPOSITION. There are two caveats, however. First, if the strings to be collated contain combining sequences that may not be in canonical order, you should set the collator to CANONICAL\_DECOMPOSITION or FULL\_DECOMPOSITION to enable sorting of combining sequences. Second, if the strings contain characters with compatibility decompositions (such as full-width and half-width forms), you must use FULL\_DECOMPOSITION, since the rule tables only include canonical mappings.

**Errors**

The following are errors:

* A text-argument contains unquoted punctuation symbols (e.g. "a < b-c < d").
* A relation or reset character not followed by a text-argument (e.g. "a < ,b").
* A reset where the text-argument (or an initial substring of the text-argument) is not already in the sequence. (e.g. "a < b & e < f")

If you produce one of these errors, a RuleBasedCollator throws a ParseException.

**Examples**

Simple: "< a < b < c < d"

Norwegian: "< a,A< b,B< c,C< d,D< e,E< f,F< g,G< h,H< i,I< j,J < k,K< l,L< m,M< n,N< o,O< p,P< q,Q< r,R< s,S< t,T < u,U< v,V< w,W< x,X< y,Y< z,Z < \u00E5=a\u030A,\u00C5=A\u030A ;aa,AA< \u00E6,\u00C6< \u00F8,\u00D8"

To create a RuleBasedCollator object with specialized rules tailored to your needs, you construct the RuleBasedCollator with the rules contained in a String object. For example:

String simple = "< a< b< c< d";  
 RuleBasedCollator mySimple = new RuleBasedCollator(simple);

Or:

String Norwegian = "< a,A< b,B< c,C< d,D< e,E< f,F< g,G< h,H< i,I< j,J" +  
 "< k,K< l,L< m,M< n,N< o,O< p,P< q,Q< r,R< s,S< t,T" +  
 "< u,U< v,V< w,W< x,X< y,Y< z,Z" +  
 "< \u00E5=a\u030A,\u00C5=A\u030A" +  
 ";aa,AA< \u00E6,\u00C6< \u00F8,\u00D8";  
 RuleBasedCollator myNorwegian = new RuleBasedCollator(Norwegian);

A new collation rules string can be created by concatenating rules strings. For example, the rules returned by [getRules()](http://docs.google.com/java/text/RuleBasedCollator.html#getRules()) could be concatenated to combine multiple RuleBasedCollators.

The following example demonstrates how to change the order of non-spacing accents,

// old rule  
 String oldRules = "=\u0301;\u0300;\u0302;\u0308" // main accents  
 + ";\u0327;\u0303;\u0304;\u0305" // main accents  
 + ";\u0306;\u0307;\u0309;\u030A" // main accents  
 + ";\u030B;\u030C;\u030D;\u030E" // main accents  
 + ";\u030F;\u0310;\u0311;\u0312" // main accents  
 + "< a , A ; ae, AE ; \u00e6 , \u00c6"  
 + "< b , B < c, C < e, E & C < d, D";  
 // change the order of accent characters  
 String addOn = "& \u0300 ; \u0308 ; \u0302";  
 RuleBasedCollator myCollator = new RuleBasedCollator(oldRules + addOn);

**See Also:**[Collator](http://docs.google.com/java/text/Collator.html), [CollationElementIterator](http://docs.google.com/java/text/CollationElementIterator.html)

| **Field Summary** | |
| --- | --- |

| **Fields inherited from class java.text.**[**Collator**](http://docs.google.com/java/text/Collator.html) |
| --- |
| [CANONICAL\_DECOMPOSITION](http://docs.google.com/java/text/Collator.html#CANONICAL_DECOMPOSITION), [FULL\_DECOMPOSITION](http://docs.google.com/java/text/Collator.html#FULL_DECOMPOSITION), [IDENTICAL](http://docs.google.com/java/text/Collator.html#IDENTICAL), [NO\_DECOMPOSITION](http://docs.google.com/java/text/Collator.html#NO_DECOMPOSITION), [PRIMARY](http://docs.google.com/java/text/Collator.html#PRIMARY), [SECONDARY](http://docs.google.com/java/text/Collator.html#SECONDARY), [TERTIARY](http://docs.google.com/java/text/Collator.html#TERTIARY) |

| **Constructor Summary** | |
| --- | --- |
| [**RuleBasedCollator**](http://docs.google.com/java/text/RuleBasedCollator.html#RuleBasedCollator(java.lang.String))([String](http://docs.google.com/java/lang/String.html) rules)            RuleBasedCollator constructor. |

| **Method Summary** | |
| --- | --- |
| [Object](http://docs.google.com/java/lang/Object.html) | [**clone**](http://docs.google.com/java/text/RuleBasedCollator.html#clone())()            Standard override; no change in semantics. |
| int | [**compare**](http://docs.google.com/java/text/RuleBasedCollator.html#compare(java.lang.String,%20java.lang.String))([String](http://docs.google.com/java/lang/String.html) source, [String](http://docs.google.com/java/lang/String.html) target)            Compares the character data stored in two different strings based on the collation rules. |
| boolean | [**equals**](http://docs.google.com/java/text/RuleBasedCollator.html#equals(java.lang.Object))([Object](http://docs.google.com/java/lang/Object.html) obj)            Compares the equality of two collation objects. |
| [CollationElementIterator](http://docs.google.com/java/text/CollationElementIterator.html) | [**getCollationElementIterator**](http://docs.google.com/java/text/RuleBasedCollator.html#getCollationElementIterator(java.text.CharacterIterator))([CharacterIterator](http://docs.google.com/java/text/CharacterIterator.html) source)            Return a CollationElementIterator for the given String. |
| [CollationElementIterator](http://docs.google.com/java/text/CollationElementIterator.html) | [**getCollationElementIterator**](http://docs.google.com/java/text/RuleBasedCollator.html#getCollationElementIterator(java.lang.String))([String](http://docs.google.com/java/lang/String.html) source)            Return a CollationElementIterator for the given String. |
| [CollationKey](http://docs.google.com/java/text/CollationKey.html) | [**getCollationKey**](http://docs.google.com/java/text/RuleBasedCollator.html#getCollationKey(java.lang.String))([String](http://docs.google.com/java/lang/String.html) source)            Transforms the string into a series of characters that can be compared with CollationKey.compareTo. |
| [String](http://docs.google.com/java/lang/String.html) | [**getRules**](http://docs.google.com/java/text/RuleBasedCollator.html#getRules())()            Gets the table-based rules for the collation object. |
| int | [**hashCode**](http://docs.google.com/java/text/RuleBasedCollator.html#hashCode())()            Generates the hash code for the table-based collation object |

| **Methods inherited from class java.text.**[**Collator**](http://docs.google.com/java/text/Collator.html) |
| --- |
| [compare](http://docs.google.com/java/text/Collator.html#compare(java.lang.Object,%20java.lang.Object)), [equals](http://docs.google.com/java/text/Collator.html#equals(java.lang.String,%20java.lang.String)), [getAvailableLocales](http://docs.google.com/java/text/Collator.html#getAvailableLocales()), [getDecomposition](http://docs.google.com/java/text/Collator.html#getDecomposition()), [getInstance](http://docs.google.com/java/text/Collator.html#getInstance()), [getInstance](http://docs.google.com/java/text/Collator.html#getInstance(java.util.Locale)), [getStrength](http://docs.google.com/java/text/Collator.html#getStrength()), [setDecomposition](http://docs.google.com/java/text/Collator.html#setDecomposition(int)), [setStrength](http://docs.google.com/java/text/Collator.html#setStrength(int)) |

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

### RuleBasedCollator

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

RuleBasedCollator constructor. This takes the table rules and builds a collation table out of them. Please see RuleBasedCollator class description for more details on the collation rule syntax.

**Parameters:**rules - the collation rules to build the collation table from. **Throws:** [ParseException](http://docs.google.com/java/text/ParseException.html) - A format exception will be thrown if the build process of the rules fails. For example, build rule "a < ? < d" will cause the constructor to throw the ParseException because the '?' is not quoted.**See Also:**[Locale](http://docs.google.com/java/util/Locale.html)

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

### getRules

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

Gets the table-based rules for the collation object.

**Returns:**returns the collation rules that the table collation object was created from.

### getCollationElementIterator

public [CollationElementIterator](http://docs.google.com/java/text/CollationElementIterator.html) **getCollationElementIterator**([String](http://docs.google.com/java/lang/String.html) source)

Return a CollationElementIterator for the given String.

**See Also:**[CollationElementIterator](http://docs.google.com/java/text/CollationElementIterator.html)

### getCollationElementIterator

public [CollationElementIterator](http://docs.google.com/java/text/CollationElementIterator.html) **getCollationElementIterator**([CharacterIterator](http://docs.google.com/java/text/CharacterIterator.html) source)

Return a CollationElementIterator for the given String.

**Since:** 1.2 **See Also:**[CollationElementIterator](http://docs.google.com/java/text/CollationElementIterator.html)

### compare

public int **compare**([String](http://docs.google.com/java/lang/String.html) source,  
 [String](http://docs.google.com/java/lang/String.html) target)

Compares the character data stored in two different strings based on the collation rules. Returns information about whether a string is less than, greater than or equal to another string in a language. This can be overriden in a subclass.

**Specified by:**[compare](http://docs.google.com/java/text/Collator.html#compare(java.lang.String,%20java.lang.String)) in class [Collator](http://docs.google.com/java/text/Collator.html) **Parameters:**source - the source string.target - the target string. **Returns:**Returns an integer value. Value is less than zero if source is less than target, value is zero if source and target are equal, value is greater than zero if source is greater than target.**See Also:**[CollationKey](http://docs.google.com/java/text/CollationKey.html), [Collator.getCollationKey(java.lang.String)](http://docs.google.com/java/text/Collator.html#getCollationKey(java.lang.String))

### getCollationKey

public [CollationKey](http://docs.google.com/java/text/CollationKey.html) **getCollationKey**([String](http://docs.google.com/java/lang/String.html) source)

Transforms the string into a series of characters that can be compared with CollationKey.compareTo. This overrides java.text.Collator.getCollationKey. It can be overriden in a subclass.

**Specified by:**[getCollationKey](http://docs.google.com/java/text/Collator.html#getCollationKey(java.lang.String)) in class [Collator](http://docs.google.com/java/text/Collator.html) **Parameters:**source - the string to be transformed into a collation key. **Returns:**the CollationKey for the given String based on this Collator's collation rules. If the source String is null, a null CollationKey is returned.**See Also:**[CollationKey](http://docs.google.com/java/text/CollationKey.html), [Collator.compare(java.lang.String, java.lang.String)](http://docs.google.com/java/text/Collator.html#compare(java.lang.String,%20java.lang.String))

### clone

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

Standard override; no change in semantics.

**Overrides:**[clone](http://docs.google.com/java/text/Collator.html#clone()) in class [Collator](http://docs.google.com/java/text/Collator.html) **Returns:**a clone of this instance.**See Also:**[Cloneable](http://docs.google.com/java/lang/Cloneable.html)

### equals

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

Compares the equality of two collation objects.

**Specified by:**[equals](http://docs.google.com/java/util/Comparator.html#equals(java.lang.Object)) in interface [Comparator](http://docs.google.com/java/util/Comparator.html)<[Object](http://docs.google.com/java/lang/Object.html)>**Overrides:**[equals](http://docs.google.com/java/text/Collator.html#equals(java.lang.Object)) in class [Collator](http://docs.google.com/java/text/Collator.html) **Parameters:**obj - the table-based collation object to be compared with this. **Returns:**true if the current table-based collation object is the same as the table-based collation object obj; false otherwise.**See Also:**[Object.hashCode()](http://docs.google.com/java/lang/Object.html#hashCode()), [Hashtable](http://docs.google.com/java/util/Hashtable.html)

### hashCode

public int **hashCode**()

Generates the hash code for the table-based collation object

**Specified by:**[hashCode](http://docs.google.com/java/text/Collator.html#hashCode()) in class [Collator](http://docs.google.com/java/text/Collator.html) **Returns:**a hash code value for this object.**See Also:**[Object.equals(java.lang.Object)](http://docs.google.com/java/lang/Object.html#equals(java.lang.Object)), [Hashtable](http://docs.google.com/java/util/Hashtable.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/RuleBasedCollator.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/text/ParsePosition.html)   [**NEXT CLASS**](http://docs.google.com/java/text/SimpleDateFormat.html) | [**FRAMES**](http://docs.google.com/index.html?java/text/RuleBasedCollator.html)    [**NO FRAMES**](http://docs.google.com/RuleBasedCollator.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | [FIELD](#2et92p0) | [CONSTR](#tyjcwt) | [METHOD](#3dy6vkm) | DETAIL: FIELD | [CONSTR](#2s8eyo1) | [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).