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

## **java.util**

Interface Formattable

public interface **Formattable**

The Formattable interface must be implemented by any class that needs to perform custom formatting using the 's' conversion specifier of [Formatter](http://docs.google.com/java/util/Formatter.html). This interface allows basic control for formatting arbitrary objects. For example, the following class prints out different representations of a stock's name depending on the flags and length constraints:

import java.nio.CharBuffer;  
 import java.util.Formatter;  
 import java.util.Formattable;  
 import java.util.Locale;  
 import static java.util.FormattableFlags.\*;  
  
 ...  
   
 public class StockName implements Formattable {  
 private String symbol, companyName, frenchCompanyName;  
 public StockName(String symbol, String companyName,  
 String frenchCompanyName) {  
 ...  
 }  
  
 ...  
  
 public void formatTo(Formatter fmt, int f, int width, int precision) {  
 StringBuilder sb = new StringBuilder();  
  
 // decide form of name   
 String name = companyName;  
 if (fmt.locale().equals(Locale.FRANCE))  
 name = frenchCompanyName;  
 boolean alternate = (f & ALTERNATE) == ALTERNATE;  
 boolean usesymbol = alternate || (precision != -1 && precision < 10);  
 String out = (usesymbol ? symbol : name);  
  
 // apply precision  
 if (precision == -1 || out.length() < precision) {  
 // write it all  
 sb.append(out);  
 } else {  
 sb.append(out.substring(0, precision - 1)).append('\*');  
 }  
  
 // apply width and justification  
 int len = sb.length();   
 if (len < width)  
 for (int i = 0; i < width - len; i++)  
 if ((f & LEFT\_JUSTIFY) == LEFT\_JUSTIFY)  
 sb.append(' ');  
 else  
 sb.insert(0, ' ');  
  
 fmt.format(sb.toString());  
 }  
  
 public String toString() {  
 return String.format("%s - %s", symbol, companyName);  
 }  
 }

When used in conjunction with the [Formatter](http://docs.google.com/java/util/Formatter.html), the above class produces the following output for various format strings.

Formatter fmt = new Formatter();  
 StockName sn = new StockName("HUGE", "Huge Fruit, Inc.",  
 "Fruit Titanesque, Inc.");  
 fmt.format("%s", sn); // -> "Huge Fruit, Inc."  
 fmt.format("%s", sn.toString()); // -> "HUGE - Huge Fruit, Inc."  
 fmt.format("%#s", sn); // -> "HUGE"  
 fmt.format("%-10.8s", sn); // -> "HUGE "  
 fmt.format("%.12s", sn); // -> "Huge Fruit,\*"  
 fmt.format(Locale.FRANCE, "%25s", sn); // -> " Fruit Titanesque, Inc."

Formattables are not necessarily safe for multithreaded access. Thread safety is optional and may be enforced by classes that extend and implement this interface.

Unless otherwise specified, passing a null argument to any method in this interface will cause a [NullPointerException](http://docs.google.com/java/lang/NullPointerException.html) to be thrown.

**Since:** 1.5

| **Method Summary** | |
| --- | --- |
| void | [**formatTo**](http://docs.google.com/java/util/Formattable.html#formatTo(java.util.Formatter,%20int,%20int,%20int))([Formatter](http://docs.google.com/java/util/Formatter.html) formatter, int flags, int width, int precision)            Formats the object using the provided [formatter](http://docs.google.com/java/util/Formatter.html). |

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

### formatTo

void **formatTo**([Formatter](http://docs.google.com/java/util/Formatter.html) formatter,  
 int flags,  
 int width,  
 int precision)

Formats the object using the provided [formatter](http://docs.google.com/java/util/Formatter.html).

**Parameters:**formatter - The [formatter](http://docs.google.com/java/util/Formatter.html). Implementing classes may call [formatter.out()](http://docs.google.com/java/util/Formatter.html#out()) or [formatter.locale()](http://docs.google.com/java/util/Formatter.html#locale()) to obtain the [Appendable](http://docs.google.com/java/lang/Appendable.html) or [Locale](http://docs.google.com/java/util/Locale.html) used by this formatter respectively.flags - The flags modify the output format. The value is interpreted as a bitmask. Any combination of the following flags may be set: [FormattableFlags.LEFT\_JUSTIFY](http://docs.google.com/java/util/FormattableFlags.html#LEFT_JUSTIFY), [FormattableFlags.UPPERCASE](http://docs.google.com/java/util/FormattableFlags.html#UPPERCASE), and [FormattableFlags.ALTERNATE](http://docs.google.com/java/util/FormattableFlags.html#ALTERNATE). If no flags are set, the default formatting of the implementing class will apply.width - The minimum number of characters to be written to the output. If the length of the converted value is less than the width then the output will be padded by '  ' until the total number of characters equals width. The padding is at the beginning by default. If the [FormattableFlags.LEFT\_JUSTIFY](http://docs.google.com/java/util/FormattableFlags.html#LEFT_JUSTIFY) flag is set then the padding will be at the end. If width is -1 then there is no minimum.precision - The maximum number of characters to be written to the output. The precision is applied before the width, thus the output will be truncated to precision characters even if the width is greater than the precision. If precision is -1 then there is no explicit limit on the number of characters. **Throws:** [IllegalFormatException](http://docs.google.com/java/util/IllegalFormatException.html) - If any of the parameters are invalid. For specification of all possible formatting errors, see the [Details](http://docs.google.com/util/Formatter.html#detail) section of the formatter class specification.

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

[Submit a bug or feature](http://bugs.sun.com/services/bugreport/index.jsp)

For further API reference and developer documentation, see [Java SE Developer Documentation](http://docs.google.com/webnotes/devdocs-vs-specs.html). That documentation contains more detailed, developer-targeted descriptions, with conceptual overviews, definitions of terms, workarounds, and working code examples.

Copyright 2006 Sun Microsystems, Inc. All rights reserved. Use is subject to [license terms](http://docs.google.com/legal/license.html). Also see the [documentation redistribution policy](http://java.sun.com/docs/redist.html).