| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/Format.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/FieldPosition.html)   [**NEXT CLASS**](http://docs.google.com/java/text/Format.Field.html) | [**FRAMES**](http://docs.google.com/index.html?java/text/Format.html)    [**NO FRAMES**](http://docs.google.com/Format.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: [NESTED](#2et92p0) | FIELD | [CONSTR](#tyjcwt) | [METHOD](#3dy6vkm) | DETAIL: FIELD | [CONSTR](#4d34og8) | [METHOD](#17dp8vu) |

## **java.text**

Class Format

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

**All Implemented Interfaces:** [Serializable](http://docs.google.com/java/io/Serializable.html), [Cloneable](http://docs.google.com/java/lang/Cloneable.html) **Direct Known Subclasses:** [DateFormat](http://docs.google.com/java/text/DateFormat.html), [MessageFormat](http://docs.google.com/java/text/MessageFormat.html), [NumberFormat](http://docs.google.com/java/text/NumberFormat.html)

public abstract class **Format**extends [Object](http://docs.google.com/java/lang/Object.html)implements [Serializable](http://docs.google.com/java/io/Serializable.html), [Cloneable](http://docs.google.com/java/lang/Cloneable.html)

Format is an abstract base class for formatting locale-sensitive information such as dates, messages, and numbers.

Format defines the programming interface for formatting locale-sensitive objects into Strings (the format method) and for parsing Strings back into objects (the parseObject method).

Generally, a format's parseObject method must be able to parse any string formatted by its format method. However, there may be exceptional cases where this is not possible. For example, a format method might create two adjacent integer numbers with no separator in between, and in this case the parseObject could not tell which digits belong to which number.

#### Subclassing

The Java Platform provides three specialized subclasses of Format-- DateFormat, MessageFormat, and NumberFormat--for formatting dates, messages, and numbers, respectively.

Concrete subclasses must implement three methods:

1. format(Object obj, StringBuffer toAppendTo, FieldPosition pos)
2. formatToCharacterIterator(Object obj)
3. parseObject(String source, ParsePosition pos)

These general methods allow polymorphic parsing and formatting of objects and are used, for example, by MessageFormat. Subclasses often also provide additional format methods for specific input types as well as parse methods for specific result types. Any parse method that does not take a ParsePosition argument should throw ParseException when no text in the required format is at the beginning of the input text.

Most subclasses will also implement the following factory methods:

1. getInstance for getting a useful format object appropriate for the current locale
2. getInstance(Locale) for getting a useful format object appropriate for the specified locale

In addition, some subclasses may also implement other getXxxxInstance methods for more specialized control. For example, the NumberFormat class provides getPercentInstance and getCurrencyInstance methods for getting specialized number formatters.

Subclasses of Format that allow programmers to create objects for locales (with getInstance(Locale) for example) must also implement the following class method:

public static Locale[] getAvailableLocales()

And finally subclasses may define a set of constants to identify the various fields in the formatted output. These constants are used to create a FieldPosition object which identifies what information is contained in the field and its position in the formatted result. These constants should be named *item*\_FIELD where *item* identifies the field. For examples of these constants, see ERA\_FIELD and its friends in [DateFormat](http://docs.google.com/java/text/DateFormat.html).

#### Synchronization

Formats are generally not synchronized. It is recommended to create separate format instances for each thread. If multiple threads access a format concurrently, it must be synchronized externally.

**See Also:**[ParsePosition](http://docs.google.com/java/text/ParsePosition.html), [FieldPosition](http://docs.google.com/java/text/FieldPosition.html), [NumberFormat](http://docs.google.com/java/text/NumberFormat.html), [DateFormat](http://docs.google.com/java/text/DateFormat.html), [MessageFormat](http://docs.google.com/java/text/MessageFormat.html), [Serialized Form](http://docs.google.com/serialized-form.html#java.text.Format)

| **Nested Class Summary** | |
| --- | --- |
| static class | [**Format.Field**](http://docs.google.com/java/text/Format.Field.html)            Defines constants that are used as attribute keys in the AttributedCharacterIterator returned from Format.formatToCharacterIterator and as field identifiers in FieldPosition. |

| **Constructor Summary** | |
| --- | --- |
| protected | [**Format**](http://docs.google.com/java/text/Format.html#Format())()            Sole constructor. |

| **Method Summary** | |
| --- | --- |
| [Object](http://docs.google.com/java/lang/Object.html) | [**clone**](http://docs.google.com/java/text/Format.html#clone())()            Creates and returns a copy of this object. |
| [String](http://docs.google.com/java/lang/String.html) | [**format**](http://docs.google.com/java/text/Format.html#format(java.lang.Object))([Object](http://docs.google.com/java/lang/Object.html) obj)            Formats an object to produce a string. |
| abstract  [StringBuffer](http://docs.google.com/java/lang/StringBuffer.html) | [**format**](http://docs.google.com/java/text/Format.html#format(java.lang.Object,%20java.lang.StringBuffer,%20java.text.FieldPosition))([Object](http://docs.google.com/java/lang/Object.html) obj, [StringBuffer](http://docs.google.com/java/lang/StringBuffer.html) toAppendTo, [FieldPosition](http://docs.google.com/java/text/FieldPosition.html) pos)            Formats an object and appends the resulting text to a given string buffer. |
| [AttributedCharacterIterator](http://docs.google.com/java/text/AttributedCharacterIterator.html) | [**formatToCharacterIterator**](http://docs.google.com/java/text/Format.html#formatToCharacterIterator(java.lang.Object))([Object](http://docs.google.com/java/lang/Object.html) obj)            Formats an Object producing an AttributedCharacterIterator. |
| [Object](http://docs.google.com/java/lang/Object.html) | [**parseObject**](http://docs.google.com/java/text/Format.html#parseObject(java.lang.String))([String](http://docs.google.com/java/lang/String.html) source)            Parses text from the beginning of the given string to produce an object. |
| abstract  [Object](http://docs.google.com/java/lang/Object.html) | [**parseObject**](http://docs.google.com/java/text/Format.html#parseObject(java.lang.String,%20java.text.ParsePosition))([String](http://docs.google.com/java/lang/String.html) source, [ParsePosition](http://docs.google.com/java/text/ParsePosition.html) pos)            Parses text from a string to produce an object. |

| **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** |
| --- |

### Format

protected **Format**()

Sole constructor. (For invocation by subclass constructors, typically implicit.)

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

### format

public final [String](http://docs.google.com/java/lang/String.html) **format**([Object](http://docs.google.com/java/lang/Object.html) obj)

Formats an object to produce a string. This is equivalent to[format](http://docs.google.com/java/text/Format.html#format(java.lang.Object,%20java.lang.StringBuffer,%20java.text.FieldPosition))(obj, new StringBuffer(), new FieldPosition(0)).toString();

**Parameters:**obj - The object to format **Returns:**Formatted string. **Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if the Format cannot format the given object

### format

public abstract [StringBuffer](http://docs.google.com/java/lang/StringBuffer.html) **format**([Object](http://docs.google.com/java/lang/Object.html) obj,  
 [StringBuffer](http://docs.google.com/java/lang/StringBuffer.html) toAppendTo,  
 [FieldPosition](http://docs.google.com/java/text/FieldPosition.html) pos)

Formats an object and appends the resulting text to a given string buffer. If the pos argument identifies a field used by the format, then its indices are set to the beginning and end of the first such field encountered.

**Parameters:**obj - The object to formattoAppendTo - where the text is to be appendedpos - A FieldPosition identifying a field in the formatted text **Returns:**the string buffer passed in as toAppendTo, with formatted text appended **Throws:** [NullPointerException](http://docs.google.com/java/lang/NullPointerException.html) - if toAppendTo or pos is null [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if the Format cannot format the given object

### formatToCharacterIterator

public [AttributedCharacterIterator](http://docs.google.com/java/text/AttributedCharacterIterator.html) **formatToCharacterIterator**([Object](http://docs.google.com/java/lang/Object.html) obj)

Formats an Object producing an AttributedCharacterIterator. You can use the returned AttributedCharacterIterator to build the resulting String, as well as to determine information about the resulting String.

Each attribute key of the AttributedCharacterIterator will be of type Field. It is up to each Format implementation to define what the legal values are for each attribute in the AttributedCharacterIterator, but typically the attribute key is also used as the attribute value.

The default implementation creates an AttributedCharacterIterator with no attributes. Subclasses that support fields should override this and create an AttributedCharacterIterator with meaningful attributes.

**Parameters:**obj - The object to format **Returns:**AttributedCharacterIterator describing the formatted value. **Throws:** [NullPointerException](http://docs.google.com/java/lang/NullPointerException.html) - if obj is null. [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - when the Format cannot format the given object.**Since:** 1.4

### parseObject

public abstract [Object](http://docs.google.com/java/lang/Object.html) **parseObject**([String](http://docs.google.com/java/lang/String.html) source,  
 [ParsePosition](http://docs.google.com/java/text/ParsePosition.html) pos)

Parses text from a string to produce an object.

The method attempts to parse text starting at the index given by pos. If parsing succeeds, then the index of pos is updated to the index after the last character used (parsing does not necessarily use all characters up to the end of the string), and the parsed object is returned. The updated pos can be used to indicate the starting point for the next call to this method. If an error occurs, then the index of pos is not changed, the error index of pos is set to the index of the character where the error occurred, and null is returned.

**Parameters:**source - A String, part of which should be parsed.pos - A ParsePosition object with index and error index information as described above. **Returns:**An Object parsed from the string. In case of error, returns null. **Throws:** [NullPointerException](http://docs.google.com/java/lang/NullPointerException.html) - if pos is null.

### parseObject

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

Parses text from the beginning of the given string to produce an object. The method may not use the entire text of the given string.

**Parameters:**source - A String whose beginning should be parsed. **Returns:**An Object parsed from the string. **Throws:** [ParseException](http://docs.google.com/java/text/ParseException.html) - if the beginning of the specified string cannot be parsed.

### clone

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

Creates and returns a copy of this object.

**Overrides:**[clone](http://docs.google.com/java/lang/Object.html#clone()) in class [Object](http://docs.google.com/java/lang/Object.html) **Returns:**a clone of this instance.**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/Format.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/FieldPosition.html)   [**NEXT CLASS**](http://docs.google.com/java/text/Format.Field.html) | [**FRAMES**](http://docs.google.com/index.html?java/text/Format.html)    [**NO FRAMES**](http://docs.google.com/Format.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: [NESTED](#2et92p0) | FIELD | [CONSTR](#tyjcwt) | [METHOD](#3dy6vkm) | DETAIL: FIELD | [CONSTR](#4d34og8) | [METHOD](#17dp8vu) |

[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).