| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/StringTokenizer.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/Stack.html)   [**NEXT CLASS**](http://docs.google.com/java/util/Timer.html) | [**FRAMES**](http://docs.google.com/index.html?java/util/StringTokenizer.html)    [**NO FRAMES**](http://docs.google.com/StringTokenizer.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | FIELD | [CONSTR](#3znysh7) | [METHOD](#2et92p0) | DETAIL: FIELD | [CONSTR](#3dy6vkm) | [METHOD](#17dp8vu) |

## **java.util**

Class StringTokenizer

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

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

public class **StringTokenizer**extends [Object](http://docs.google.com/java/lang/Object.html)implements [Enumeration](http://docs.google.com/java/util/Enumeration.html)<[Object](http://docs.google.com/java/lang/Object.html)>

The string tokenizer class allows an application to break a string into tokens. The tokenization method is much simpler than the one used by the StreamTokenizer class. The StringTokenizer methods do not distinguish among identifiers, numbers, and quoted strings, nor do they recognize and skip comments.

The set of delimiters (the characters that separate tokens) may be specified either at creation time or on a per-token basis.

An instance of StringTokenizer behaves in one of two ways, depending on whether it was created with the returnDelims flag having the value true or false:

* If the flag is false, delimiter characters serve to separate tokens. A token is a maximal sequence of consecutive characters that are not delimiters.
* If the flag is true, delimiter characters are themselves considered to be tokens. A token is thus either one delimiter character, or a maximal sequence of consecutive characters that are not delimiters.

A StringTokenizer object internally maintains a current position within the string to be tokenized. Some operations advance this current position past the characters processed.

A token is returned by taking a substring of the string that was used to create the StringTokenizer object.

The following is one example of the use of the tokenizer. The code:

StringTokenizer st = new StringTokenizer("this is a test");  
 while (st.hasMoreTokens()) {  
 System.out.println(st.nextToken());  
 }

prints the following output:

this  
 is  
 a  
 test

StringTokenizer is a legacy class that is retained for compatibility reasons although its use is discouraged in new code. It is recommended that anyone seeking this functionality use the split method of String or the java.util.regex package instead.

The following example illustrates how the String.split method can be used to break up a string into its basic tokens:

String[] result = "this is a test".split("\\s");  
 for (int x=0; x<result.length; x++)  
 System.out.println(result[x]);

prints the following output:

this  
 is  
 a  
 test

**Since:** JDK1.0 **See Also:**[StreamTokenizer](http://docs.google.com/java/io/StreamTokenizer.html)

| **Constructor Summary** | |
| --- | --- |
| [**StringTokenizer**](http://docs.google.com/java/util/StringTokenizer.html#StringTokenizer(java.lang.String))([String](http://docs.google.com/java/lang/String.html) str)            Constructs a string tokenizer for the specified string. |
| [**StringTokenizer**](http://docs.google.com/java/util/StringTokenizer.html#StringTokenizer(java.lang.String,%20java.lang.String))([String](http://docs.google.com/java/lang/String.html) str, [String](http://docs.google.com/java/lang/String.html) delim)            Constructs a string tokenizer for the specified string. |
| [**StringTokenizer**](http://docs.google.com/java/util/StringTokenizer.html#StringTokenizer(java.lang.String,%20java.lang.String,%20boolean))([String](http://docs.google.com/java/lang/String.html) str, [String](http://docs.google.com/java/lang/String.html) delim, boolean returnDelims)            Constructs a string tokenizer for the specified string. |

| **Method Summary** | |
| --- | --- |
| int | [**countTokens**](http://docs.google.com/java/util/StringTokenizer.html#countTokens())()            Calculates the number of times that this tokenizer's nextToken method can be called before it generates an exception. |
| boolean | [**hasMoreElements**](http://docs.google.com/java/util/StringTokenizer.html#hasMoreElements())()            Returns the same value as the hasMoreTokens method. |
| boolean | [**hasMoreTokens**](http://docs.google.com/java/util/StringTokenizer.html#hasMoreTokens())()            Tests if there are more tokens available from this tokenizer's string. |
| [Object](http://docs.google.com/java/lang/Object.html) | [**nextElement**](http://docs.google.com/java/util/StringTokenizer.html#nextElement())()            Returns the same value as the nextToken method, except that its declared return value is Object rather than String. |
| [String](http://docs.google.com/java/lang/String.html) | [**nextToken**](http://docs.google.com/java/util/StringTokenizer.html#nextToken())()            Returns the next token from this string tokenizer. |
| [String](http://docs.google.com/java/lang/String.html) | [**nextToken**](http://docs.google.com/java/util/StringTokenizer.html#nextToken(java.lang.String))([String](http://docs.google.com/java/lang/String.html) delim)            Returns the next token in this string tokenizer's string. |

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

### StringTokenizer

public **StringTokenizer**([String](http://docs.google.com/java/lang/String.html) str,  
 [String](http://docs.google.com/java/lang/String.html) delim,  
 boolean returnDelims)

Constructs a string tokenizer for the specified string. All characters in the delim argument are the delimiters for separating tokens.

If the returnDelims flag is true, then the delimiter characters are also returned as tokens. Each delimiter is returned as a string of length one. If the flag is false, the delimiter characters are skipped and only serve as separators between tokens.

Note that if delim is null, this constructor does not throw an exception. However, trying to invoke other methods on the resulting StringTokenizer may result in a NullPointerException.

**Parameters:**str - a string to be parsed.delim - the delimiters.returnDelims - flag indicating whether to return the delimiters as tokens. **Throws:** [NullPointerException](http://docs.google.com/java/lang/NullPointerException.html) - if str is null

### StringTokenizer

public **StringTokenizer**([String](http://docs.google.com/java/lang/String.html) str,  
 [String](http://docs.google.com/java/lang/String.html) delim)

Constructs a string tokenizer for the specified string. The characters in the delim argument are the delimiters for separating tokens. Delimiter characters themselves will not be treated as tokens.

Note that if delim is null, this constructor does not throw an exception. However, trying to invoke other methods on the resulting StringTokenizer may result in a NullPointerException.

**Parameters:**str - a string to be parsed.delim - the delimiters. **Throws:** [NullPointerException](http://docs.google.com/java/lang/NullPointerException.html) - if str is null

### StringTokenizer

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

Constructs a string tokenizer for the specified string. The tokenizer uses the default delimiter set, which is " \t\n\r\f": the space character, the tab character, the newline character, the carriage-return character, and the form-feed character. Delimiter characters themselves will not be treated as tokens.

**Parameters:**str - a string to be parsed. **Throws:** [NullPointerException](http://docs.google.com/java/lang/NullPointerException.html) - if str is null

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

### hasMoreTokens

public boolean **hasMoreTokens**()

Tests if there are more tokens available from this tokenizer's string. If this method returns true, then a subsequent call to nextToken with no argument will successfully return a token.

**Returns:**true if and only if there is at least one token in the string after the current position; false otherwise.

### nextToken

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

Returns the next token from this string tokenizer.

**Returns:**the next token from this string tokenizer. **Throws:** [NoSuchElementException](http://docs.google.com/java/util/NoSuchElementException.html) - if there are no more tokens in this tokenizer's string.

### nextToken

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

Returns the next token in this string tokenizer's string. First, the set of characters considered to be delimiters by this StringTokenizer object is changed to be the characters in the string delim. Then the next token in the string after the current position is returned. The current position is advanced beyond the recognized token. The new delimiter set remains the default after this call.

**Parameters:**delim - the new delimiters. **Returns:**the next token, after switching to the new delimiter set. **Throws:** [NoSuchElementException](http://docs.google.com/java/util/NoSuchElementException.html) - if there are no more tokens in this tokenizer's string. [NullPointerException](http://docs.google.com/java/lang/NullPointerException.html) - if delim is null

### hasMoreElements

public boolean **hasMoreElements**()

Returns the same value as the hasMoreTokens method. It exists so that this class can implement the Enumeration interface.

**Specified by:**[hasMoreElements](http://docs.google.com/java/util/Enumeration.html#hasMoreElements()) in interface [Enumeration](http://docs.google.com/java/util/Enumeration.html)<[Object](http://docs.google.com/java/lang/Object.html)> **Returns:**true if there are more tokens; false otherwise.**See Also:**[Enumeration](http://docs.google.com/java/util/Enumeration.html), [hasMoreTokens()](http://docs.google.com/java/util/StringTokenizer.html#hasMoreTokens())

### nextElement

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

Returns the same value as the nextToken method, except that its declared return value is Object rather than String. It exists so that this class can implement the Enumeration interface.

**Specified by:**[nextElement](http://docs.google.com/java/util/Enumeration.html#nextElement()) in interface [Enumeration](http://docs.google.com/java/util/Enumeration.html)<[Object](http://docs.google.com/java/lang/Object.html)> **Returns:**the next token in the string. **Throws:** [NoSuchElementException](http://docs.google.com/java/util/NoSuchElementException.html) - if there are no more tokens in this tokenizer's string.**See Also:**[Enumeration](http://docs.google.com/java/util/Enumeration.html), [nextToken()](http://docs.google.com/java/util/StringTokenizer.html#nextToken())

### countTokens

public int **countTokens**()

Calculates the number of times that this tokenizer's nextToken method can be called before it generates an exception. The current position is not advanced.

**Returns:**the number of tokens remaining in the string using the current delimiter set.**See Also:**[nextToken()](http://docs.google.com/java/util/StringTokenizer.html#nextToken())

| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/StringTokenizer.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/Stack.html)   [**NEXT CLASS**](http://docs.google.com/java/util/Timer.html) | [**FRAMES**](http://docs.google.com/index.html?java/util/StringTokenizer.html)    [**NO FRAMES**](http://docs.google.com/StringTokenizer.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | FIELD | [CONSTR](#3znysh7) | [METHOD](#2et92p0) | DETAIL: FIELD | [CONSTR](#3dy6vkm) | [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).