| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/Deflater.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/zip/DataFormatException.html)   [**NEXT CLASS**](http://docs.google.com/java/util/zip/DeflaterInputStream.html) | [**FRAMES**](http://docs.google.com/index.html?java/util/zip/Deflater.html)    [**NO FRAMES**](http://docs.google.com/Deflater.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | [FIELD](#3znysh7) | [CONSTR](#2et92p0) | [METHOD](#tyjcwt) | DETAIL: [FIELD](#1t3h5sf) | [CONSTR](#44sinio) | [METHOD](#1y810tw) |

## **java.util.zip**

Class Deflater

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

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

This class provides support for general purpose compression using the popular ZLIB compression library. The ZLIB compression library was initially developed as part of the PNG graphics standard and is not protected by patents. It is fully described in the specifications at the [java.util.zip package description](http://docs.google.com/package-summary.html#package_description).

The following code fragment demonstrates a trivial compression and decompression of a string using Deflater and Inflater.

try {  
 // Encode a String into bytes  
 String inputString = "blahblahblah??";  
 byte[] input = inputString.getBytes("UTF-8");  
  
 // Compress the bytes  
 byte[] output = new byte[100];  
 Deflater compresser = new Deflater();  
 compresser.setInput(input);  
 compresser.finish();  
 int compressedDataLength = compresser.deflate(output);  
  
 // Decompress the bytes  
 Inflater decompresser = new Inflater();  
 decompresser.setInput(output, 0, compressedDataLength);  
 byte[] result = new byte[100];  
 int resultLength = decompresser.inflate(result);  
 decompresser.end();  
  
 // Decode the bytes into a String  
 String outputString = new String(result, 0, resultLength, "UTF-8");  
 } catch(java.io.UnsupportedEncodingException ex) {  
 // handle  
 } catch (java.util.zip.DataFormatException ex) {  
 // handle  
 }

**See Also:**[Inflater](http://docs.google.com/java/util/zip/Inflater.html)

| **Field Summary** | |
| --- | --- |
| static int | [**BEST\_COMPRESSION**](http://docs.google.com/java/util/zip/Deflater.html#BEST_COMPRESSION)            Compression level for best compression. |
| static int | [**BEST\_SPEED**](http://docs.google.com/java/util/zip/Deflater.html#BEST_SPEED)            Compression level for fastest compression. |
| static int | [**DEFAULT\_COMPRESSION**](http://docs.google.com/java/util/zip/Deflater.html#DEFAULT_COMPRESSION)            Default compression level. |
| static int | [**DEFAULT\_STRATEGY**](http://docs.google.com/java/util/zip/Deflater.html#DEFAULT_STRATEGY)            Default compression strategy. |
| static int | [**DEFLATED**](http://docs.google.com/java/util/zip/Deflater.html#DEFLATED)            Compression method for the deflate algorithm (the only one currently supported). |
| static int | [**FILTERED**](http://docs.google.com/java/util/zip/Deflater.html#FILTERED)            Compression strategy best used for data consisting mostly of small values with a somewhat random distribution. |
| static int | [**HUFFMAN\_ONLY**](http://docs.google.com/java/util/zip/Deflater.html#HUFFMAN_ONLY)            Compression strategy for Huffman coding only. |
| static int | [**NO\_COMPRESSION**](http://docs.google.com/java/util/zip/Deflater.html#NO_COMPRESSION)            Compression level for no compression. |

| **Constructor Summary** | |
| --- | --- |
| [**Deflater**](http://docs.google.com/java/util/zip/Deflater.html#Deflater())()            Creates a new compressor with the default compression level. |
| [**Deflater**](http://docs.google.com/java/util/zip/Deflater.html#Deflater(int))(int level)            Creates a new compressor using the specified compression level. |
| [**Deflater**](http://docs.google.com/java/util/zip/Deflater.html#Deflater(int,%20boolean))(int level, boolean nowrap)            Creates a new compressor using the specified compression level. |

| **Method Summary** | |
| --- | --- |
| int | [**deflate**](http://docs.google.com/java/util/zip/Deflater.html#deflate(byte%5B%5D))(byte[] b)            Fills specified buffer with compressed data. |
| int | [**deflate**](http://docs.google.com/java/util/zip/Deflater.html#deflate(byte%5B%5D,%20int,%20int))(byte[] b, int off, int len)            Fills specified buffer with compressed data. |
| void | [**end**](http://docs.google.com/java/util/zip/Deflater.html#end())()            Closes the compressor and discards any unprocessed input. |
| protected  void | [**finalize**](http://docs.google.com/java/util/zip/Deflater.html#finalize())()            Closes the compressor when garbage is collected. |
| void | [**finish**](http://docs.google.com/java/util/zip/Deflater.html#finish())()            When called, indicates that compression should end with the current contents of the input buffer. |
| boolean | [**finished**](http://docs.google.com/java/util/zip/Deflater.html#finished())()            Returns true if the end of the compressed data output stream has been reached. |
| int | [**getAdler**](http://docs.google.com/java/util/zip/Deflater.html#getAdler())()            Returns the ADLER-32 value of the uncompressed data. |
| long | [**getBytesRead**](http://docs.google.com/java/util/zip/Deflater.html#getBytesRead())()            Returns the total number of uncompressed bytes input so far. |
| long | [**getBytesWritten**](http://docs.google.com/java/util/zip/Deflater.html#getBytesWritten())()            Returns the total number of compressed bytes output so far. |
| int | [**getTotalIn**](http://docs.google.com/java/util/zip/Deflater.html#getTotalIn())()            Returns the total number of uncompressed bytes input so far. |
| int | [**getTotalOut**](http://docs.google.com/java/util/zip/Deflater.html#getTotalOut())()            Returns the total number of compressed bytes output so far. |
| boolean | [**needsInput**](http://docs.google.com/java/util/zip/Deflater.html#needsInput())()            Returns true if the input data buffer is empty and setInput() should be called in order to provide more input. |
| void | [**reset**](http://docs.google.com/java/util/zip/Deflater.html#reset())()            Resets deflater so that a new set of input data can be processed. |
| void | [**setDictionary**](http://docs.google.com/java/util/zip/Deflater.html#setDictionary(byte%5B%5D))(byte[] b)            Sets preset dictionary for compression. |
| void | [**setDictionary**](http://docs.google.com/java/util/zip/Deflater.html#setDictionary(byte%5B%5D,%20int,%20int))(byte[] b, int off, int len)            Sets preset dictionary for compression. |
| void | [**setInput**](http://docs.google.com/java/util/zip/Deflater.html#setInput(byte%5B%5D))(byte[] b)            Sets input data for compression. |
| void | [**setInput**](http://docs.google.com/java/util/zip/Deflater.html#setInput(byte%5B%5D,%20int,%20int))(byte[] b, int off, int len)            Sets input data for compression. |
| void | [**setLevel**](http://docs.google.com/java/util/zip/Deflater.html#setLevel(int))(int level)            Sets the current compression level to the specified value. |
| void | [**setStrategy**](http://docs.google.com/java/util/zip/Deflater.html#setStrategy(int))(int strategy)            Sets the compression strategy to the specified value. |

| **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)), [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)) |

| **Field Detail** |
| --- |

### DEFLATED

public static final int **DEFLATED**

Compression method for the deflate algorithm (the only one currently supported).

**See Also:**[Constant Field Values](http://docs.google.com/constant-values.html#java.util.zip.Deflater.DEFLATED)

### NO\_COMPRESSION

public static final int **NO\_COMPRESSION**

Compression level for no compression.

**See Also:**[Constant Field Values](http://docs.google.com/constant-values.html#java.util.zip.Deflater.NO_COMPRESSION)

### BEST\_SPEED

public static final int **BEST\_SPEED**

Compression level for fastest compression.

**See Also:**[Constant Field Values](http://docs.google.com/constant-values.html#java.util.zip.Deflater.BEST_SPEED)

### BEST\_COMPRESSION

public static final int **BEST\_COMPRESSION**

Compression level for best compression.

**See Also:**[Constant Field Values](http://docs.google.com/constant-values.html#java.util.zip.Deflater.BEST_COMPRESSION)

### DEFAULT\_COMPRESSION

public static final int **DEFAULT\_COMPRESSION**

Default compression level.

**See Also:**[Constant Field Values](http://docs.google.com/constant-values.html#java.util.zip.Deflater.DEFAULT_COMPRESSION)

### FILTERED

public static final int **FILTERED**

Compression strategy best used for data consisting mostly of small values with a somewhat random distribution. Forces more Huffman coding and less string matching.

**See Also:**[Constant Field Values](http://docs.google.com/constant-values.html#java.util.zip.Deflater.FILTERED)

### HUFFMAN\_ONLY

public static final int **HUFFMAN\_ONLY**

Compression strategy for Huffman coding only.

**See Also:**[Constant Field Values](http://docs.google.com/constant-values.html#java.util.zip.Deflater.HUFFMAN_ONLY)

### DEFAULT\_STRATEGY

public static final int **DEFAULT\_STRATEGY**

Default compression strategy.

**See Also:**[Constant Field Values](http://docs.google.com/constant-values.html#java.util.zip.Deflater.DEFAULT_STRATEGY)

| **Constructor Detail** |
| --- |

### Deflater

public **Deflater**(int level,  
 boolean nowrap)

Creates a new compressor using the specified compression level. If 'nowrap' is true then the ZLIB header and checksum fields will not be used in order to support the compression format used in both GZIP and PKZIP.

**Parameters:**level - the compression level (0-9)nowrap - if true then use GZIP compatible compression

### Deflater

public **Deflater**(int level)

Creates a new compressor using the specified compression level. Compressed data will be generated in ZLIB format.

**Parameters:**level - the compression level (0-9)

### Deflater

public **Deflater**()

Creates a new compressor with the default compression level. Compressed data will be generated in ZLIB format.

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

### setInput

public void **setInput**(byte[] b,  
 int off,  
 int len)

Sets input data for compression. This should be called whenever needsInput() returns true indicating that more input data is required.

**Parameters:**b - the input data bytesoff - the start offset of the datalen - the length of the data**See Also:**[needsInput()](http://docs.google.com/java/util/zip/Deflater.html#needsInput())

### setInput

public void **setInput**(byte[] b)

Sets input data for compression. This should be called whenever needsInput() returns true indicating that more input data is required.

**Parameters:**b - the input data bytes**See Also:**[needsInput()](http://docs.google.com/java/util/zip/Deflater.html#needsInput())

### setDictionary

public void **setDictionary**(byte[] b,  
 int off,  
 int len)

Sets preset dictionary for compression. A preset dictionary is used when the history buffer can be predetermined. When the data is later uncompressed with Inflater.inflate(), Inflater.getAdler() can be called in order to get the Adler-32 value of the dictionary required for decompression.

**Parameters:**b - the dictionary data bytesoff - the start offset of the datalen - the length of the data**See Also:**[Inflater.inflate(byte[], int, int)](http://docs.google.com/java/util/zip/Inflater.html#inflate(byte%5B%5D,%20int,%20int)), [Inflater.getAdler()](http://docs.google.com/java/util/zip/Inflater.html#getAdler())

### setDictionary

public void **setDictionary**(byte[] b)

Sets preset dictionary for compression. A preset dictionary is used when the history buffer can be predetermined. When the data is later uncompressed with Inflater.inflate(), Inflater.getAdler() can be called in order to get the Adler-32 value of the dictionary required for decompression.

**Parameters:**b - the dictionary data bytes**See Also:**[Inflater.inflate(byte[], int, int)](http://docs.google.com/java/util/zip/Inflater.html#inflate(byte%5B%5D,%20int,%20int)), [Inflater.getAdler()](http://docs.google.com/java/util/zip/Inflater.html#getAdler())

### setStrategy

public void **setStrategy**(int strategy)

Sets the compression strategy to the specified value.

**Parameters:**strategy - the new compression strategy **Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if the compression strategy is invalid

### setLevel

public void **setLevel**(int level)

Sets the current compression level to the specified value.

**Parameters:**level - the new compression level (0-9) **Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if the compression level is invalid

### needsInput

public boolean **needsInput**()

Returns true if the input data buffer is empty and setInput() should be called in order to provide more input.

**Returns:**true if the input data buffer is empty and setInput() should be called in order to provide more input

### finish

public void **finish**()

When called, indicates that compression should end with the current contents of the input buffer.

### finished

public boolean **finished**()

Returns true if the end of the compressed data output stream has been reached.

**Returns:**true if the end of the compressed data output stream has been reached

### deflate

public int **deflate**(byte[] b,  
 int off,  
 int len)

Fills specified buffer with compressed data. Returns actual number of bytes of compressed data. A return value of 0 indicates that needsInput() should be called in order to determine if more input data is required.

**Parameters:**b - the buffer for the compressed dataoff - the start offset of the datalen - the maximum number of bytes of compressed data **Returns:**the actual number of bytes of compressed data

### deflate

public int **deflate**(byte[] b)

Fills specified buffer with compressed data. Returns actual number of bytes of compressed data. A return value of 0 indicates that needsInput() should be called in order to determine if more input data is required.

**Parameters:**b - the buffer for the compressed data **Returns:**the actual number of bytes of compressed data

### getAdler

public int **getAdler**()

Returns the ADLER-32 value of the uncompressed data.

**Returns:**the ADLER-32 value of the uncompressed data

### getTotalIn

public int **getTotalIn**()

Returns the total number of uncompressed bytes input so far.

Since the number of bytes may be greater than Integer.MAX\_VALUE, the [getBytesRead()](http://docs.google.com/java/util/zip/Deflater.html#getBytesRead()) method is now the preferred means of obtaining this information.

**Returns:**the total number of uncompressed bytes input so far

### getBytesRead

public long **getBytesRead**()

Returns the total number of uncompressed bytes input so far.

**Returns:**the total (non-negative) number of uncompressed bytes input so far**Since:** 1.5

### getTotalOut

public int **getTotalOut**()

Returns the total number of compressed bytes output so far.

Since the number of bytes may be greater than Integer.MAX\_VALUE, the [getBytesWritten()](http://docs.google.com/java/util/zip/Deflater.html#getBytesWritten()) method is now the preferred means of obtaining this information.

**Returns:**the total number of compressed bytes output so far

### getBytesWritten

public long **getBytesWritten**()

Returns the total number of compressed bytes output so far.

**Returns:**the total (non-negative) number of compressed bytes output so far**Since:** 1.5

### reset

public void **reset**()

Resets deflater so that a new set of input data can be processed. Keeps current compression level and strategy settings.

### end

public void **end**()

Closes the compressor and discards any unprocessed input. This method should be called when the compressor is no longer being used, but will also be called automatically by the finalize() method. Once this method is called, the behavior of the Deflater object is undefined.

### finalize

protected void **finalize**()

Closes the compressor when garbage is collected.

**Overrides:**[finalize](http://docs.google.com/java/lang/Object.html#finalize()) in class [Object](http://docs.google.com/java/lang/Object.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/Deflater.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/zip/DataFormatException.html)   [**NEXT CLASS**](http://docs.google.com/java/util/zip/DeflaterInputStream.html) | [**FRAMES**](http://docs.google.com/index.html?java/util/zip/Deflater.html)    [**NO FRAMES**](http://docs.google.com/Deflater.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | [FIELD](#3znysh7) | [CONSTR](#2et92p0) | [METHOD](#tyjcwt) | DETAIL: [FIELD](#1t3h5sf) | [CONSTR](#44sinio) | [METHOD](#1y810tw) |

[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).