| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/MemoryHandler.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/logging/LogRecord.html)   [**NEXT CLASS**](http://docs.google.com/java/util/logging/SimpleFormatter.html) | [**FRAMES**](http://docs.google.com/index.html?java/util/logging/MemoryHandler.html)    [**NO FRAMES**](http://docs.google.com/MemoryHandler.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | FIELD | [CONSTR](#3znysh7) | [METHOD](#2et92p0) | DETAIL: FIELD | [CONSTR](#1t3h5sf) | [METHOD](#17dp8vu) |

## **java.util.logging**

Class MemoryHandler

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

public class **MemoryHandler**extends [Handler](http://docs.google.com/java/util/logging/Handler.html)

Handler that buffers requests in a circular buffer in memory.

Normally this Handler simply stores incoming LogRecords into its memory buffer and discards earlier records. This buffering is very cheap and avoids formatting costs. On certain trigger conditions, the MemoryHandler will push out its current buffer contents to a target Handler, which will typically publish them to the outside world.

There are three main models for triggering a push of the buffer:

* An incoming LogRecord has a type that is greater than a pre-defined level, the pushLevel.
* An external class calls the push method explicitly.
* A subclass overrides the log method and scans each incoming LogRecord and calls push if a record matches some desired criteria.

**Configuration:** By default each MemoryHandler is initialized using the following LogManager configuration properties. If properties are not defined (or have invalid values) then the specified default values are used. If no default value is defined then a RuntimeException is thrown.

* java.util.logging.MemoryHandler.level specifies the level for the Handler (defaults to Level.ALL).
* java.util.logging.MemoryHandler.filter specifies the name of a Filter class to use (defaults to no Filter).
* java.util.logging.MemoryHandler.size defines the buffer size (defaults to 1000).
* java.util.logging.MemoryHandler.push defines the pushLevel (defaults to level.SEVERE).
* java.util.logging.MemoryHandler.target specifies the name of the target Handler class. (no default).

**Since:** 1.4

| **Constructor Summary** | |
| --- | --- |
| [**MemoryHandler**](http://docs.google.com/java/util/logging/MemoryHandler.html#MemoryHandler())()            Create a MemoryHandler and configure it based on LogManager configuration properties. |
| [**MemoryHandler**](http://docs.google.com/java/util/logging/MemoryHandler.html#MemoryHandler(java.util.logging.Handler,%20int,%20java.util.logging.Level))([Handler](http://docs.google.com/java/util/logging/Handler.html) target, int size, [Level](http://docs.google.com/java/util/logging/Level.html) pushLevel)            Create a MemoryHandler. |

| **Method Summary** | |
| --- | --- |
| void | [**close**](http://docs.google.com/java/util/logging/MemoryHandler.html#close())()            Close the Handler and free all associated resources. |
| void | [**flush**](http://docs.google.com/java/util/logging/MemoryHandler.html#flush())()            Causes a flush on the target Handler. |
| [Level](http://docs.google.com/java/util/logging/Level.html) | [**getPushLevel**](http://docs.google.com/java/util/logging/MemoryHandler.html#getPushLevel())()            Get the pushLevel. |
| boolean | [**isLoggable**](http://docs.google.com/java/util/logging/MemoryHandler.html#isLoggable(java.util.logging.LogRecord))([LogRecord](http://docs.google.com/java/util/logging/LogRecord.html) record)            Check if this Handler would actually log a given LogRecord into its internal buffer. |
| void | [**publish**](http://docs.google.com/java/util/logging/MemoryHandler.html#publish(java.util.logging.LogRecord))([LogRecord](http://docs.google.com/java/util/logging/LogRecord.html) record)            Store a LogRecord in an internal buffer. |
| void | [**push**](http://docs.google.com/java/util/logging/MemoryHandler.html#push())()            Push any buffered output to the target Handler. |
| void | [**setPushLevel**](http://docs.google.com/java/util/logging/MemoryHandler.html#setPushLevel(java.util.logging.Level))([Level](http://docs.google.com/java/util/logging/Level.html) newLevel)            Set the pushLevel. |

| **Methods inherited from class java.util.logging.**[**Handler**](http://docs.google.com/java/util/logging/Handler.html) |
| --- |
| [getEncoding](http://docs.google.com/java/util/logging/Handler.html#getEncoding()), [getErrorManager](http://docs.google.com/java/util/logging/Handler.html#getErrorManager()), [getFilter](http://docs.google.com/java/util/logging/Handler.html#getFilter()), [getFormatter](http://docs.google.com/java/util/logging/Handler.html#getFormatter()), [getLevel](http://docs.google.com/java/util/logging/Handler.html#getLevel()), [reportError](http://docs.google.com/java/util/logging/Handler.html#reportError(java.lang.String,%20java.lang.Exception,%20int)), [setEncoding](http://docs.google.com/java/util/logging/Handler.html#setEncoding(java.lang.String)), [setErrorManager](http://docs.google.com/java/util/logging/Handler.html#setErrorManager(java.util.logging.ErrorManager)), [setFilter](http://docs.google.com/java/util/logging/Handler.html#setFilter(java.util.logging.Filter)), [setFormatter](http://docs.google.com/java/util/logging/Handler.html#setFormatter(java.util.logging.Formatter)), [setLevel](http://docs.google.com/java/util/logging/Handler.html#setLevel(java.util.logging.Level)) |

| **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** |
| --- |

### MemoryHandler

public **MemoryHandler**()

Create a MemoryHandler and configure it based on LogManager configuration properties.

### MemoryHandler

public **MemoryHandler**([Handler](http://docs.google.com/java/util/logging/Handler.html) target,  
 int size,  
 [Level](http://docs.google.com/java/util/logging/Level.html) pushLevel)

Create a MemoryHandler.

The MemoryHandler is configured based on LogManager properties (or their default values) except that the given pushLevel argument and buffer size argument are used.

**Parameters:**target - the Handler to which to publish output.size - the number of log records to buffer (must be greater than zero)pushLevel - message level to push on **Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - is size is <= 0

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

### publish

public void **publish**([LogRecord](http://docs.google.com/java/util/logging/LogRecord.html) record)

Store a LogRecord in an internal buffer.

If there is a Filter, its isLoggable method is called to check if the given log record is loggable. If not we return. Otherwise the given record is copied into an internal circular buffer. Then the record's level property is compared with the pushLevel. If the given level is greater than or equal to the pushLevel then push is called to write all buffered records to the target output Handler.

**Specified by:**[publish](http://docs.google.com/java/util/logging/Handler.html#publish(java.util.logging.LogRecord)) in class [Handler](http://docs.google.com/java/util/logging/Handler.html) **Parameters:**record - description of the log event. A null record is silently ignored and is not published

### push

public void **push**()

Push any buffered output to the target Handler.

The buffer is then cleared.

### flush

public void **flush**()

Causes a flush on the target Handler.

Note that the current contents of the MemoryHandler buffer are **not** written out. That requires a "push".

**Specified by:**[flush](http://docs.google.com/java/util/logging/Handler.html#flush()) in class [Handler](http://docs.google.com/java/util/logging/Handler.html)

### close

public void **close**()  
 throws [SecurityException](http://docs.google.com/java/lang/SecurityException.html)

Close the Handler and free all associated resources. This will also close the target Handler.

**Specified by:**[close](http://docs.google.com/java/util/logging/Handler.html#close()) in class [Handler](http://docs.google.com/java/util/logging/Handler.html) **Throws:** [SecurityException](http://docs.google.com/java/lang/SecurityException.html) - if a security manager exists and if the caller does not have LoggingPermission("control").

### setPushLevel

public void **setPushLevel**([Level](http://docs.google.com/java/util/logging/Level.html) newLevel)  
 throws [SecurityException](http://docs.google.com/java/lang/SecurityException.html)

Set the pushLevel. After a LogRecord is copied into our internal buffer, if its level is greater than or equal to the pushLevel, then push will be called.

**Parameters:**newLevel - the new value of the pushLevel **Throws:** [SecurityException](http://docs.google.com/java/lang/SecurityException.html) - if a security manager exists and if the caller does not have LoggingPermission("control").

### getPushLevel

public [Level](http://docs.google.com/java/util/logging/Level.html) **getPushLevel**()

Get the pushLevel.

**Returns:**the value of the pushLevel

### isLoggable

public boolean **isLoggable**([LogRecord](http://docs.google.com/java/util/logging/LogRecord.html) record)

Check if this Handler would actually log a given LogRecord into its internal buffer.

This method checks if the LogRecord has an appropriate level and whether it satisfies any Filter. However it does **not** check whether the LogRecord would result in a "push" of the buffer contents. It will return false if the LogRecord is Null.

**Overrides:**[isLoggable](http://docs.google.com/java/util/logging/Handler.html#isLoggable(java.util.logging.LogRecord)) in class [Handler](http://docs.google.com/java/util/logging/Handler.html) **Parameters:**record - a LogRecord **Returns:**true if the LogRecord would be logged.

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