| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/Handler.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/Formatter.html)   [**NEXT CLASS**](http://docs.google.com/java/util/logging/Level.html) | [**FRAMES**](http://docs.google.com/index.html?java/util/logging/Handler.html)    [**NO FRAMES**](http://docs.google.com/Handler.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | FIELD | [CONSTR](#3znysh7) | [METHOD](#2et92p0) | DETAIL: FIELD | [CONSTR](#3dy6vkm) | [METHOD](#4d34og8) |

## **java.util.logging**

Class Handler

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

**Direct Known Subclasses:** [MemoryHandler](http://docs.google.com/java/util/logging/MemoryHandler.html), [StreamHandler](http://docs.google.com/java/util/logging/StreamHandler.html)

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

A Handler object takes log messages from a Logger and exports them. It might for example, write them to a console or write them to a file, or send them to a network logging service, or forward them to an OS log, or whatever.

A Handler can be disabled by doing a setLevel(Level.OFF) and can be re-enabled by doing a setLevel with an appropriate level.

Handler classes typically use LogManager properties to set default values for the Handler's Filter, Formatter, and Level. See the specific documentation for each concrete Handler class.

**Since:** 1.4

| **Constructor Summary** | |
| --- | --- |
| protected | [**Handler**](http://docs.google.com/java/util/logging/Handler.html#Handler())()            Default constructor. |

| **Method Summary** | |
| --- | --- |
| abstract  void | [**close**](http://docs.google.com/java/util/logging/Handler.html#close())()            Close the Handler and free all associated resources. |
| abstract  void | [**flush**](http://docs.google.com/java/util/logging/Handler.html#flush())()            Flush any buffered output. |
| [String](http://docs.google.com/java/lang/String.html) | [**getEncoding**](http://docs.google.com/java/util/logging/Handler.html#getEncoding())()            Return the character encoding for this Handler. |
| [ErrorManager](http://docs.google.com/java/util/logging/ErrorManager.html) | [**getErrorManager**](http://docs.google.com/java/util/logging/Handler.html#getErrorManager())()            Retrieves the ErrorManager for this Handler. |
| [Filter](http://docs.google.com/java/util/logging/Filter.html) | [**getFilter**](http://docs.google.com/java/util/logging/Handler.html#getFilter())()            Get the current Filter for this Handler. |
| [Formatter](http://docs.google.com/java/util/logging/Formatter.html) | [**getFormatter**](http://docs.google.com/java/util/logging/Handler.html#getFormatter())()            Return the Formatter for this Handler. |
| [Level](http://docs.google.com/java/util/logging/Level.html) | [**getLevel**](http://docs.google.com/java/util/logging/Handler.html#getLevel())()            Get the log level specifying which messages will be logged by this Handler. |
| boolean | [**isLoggable**](http://docs.google.com/java/util/logging/Handler.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. |
| abstract  void | [**publish**](http://docs.google.com/java/util/logging/Handler.html#publish(java.util.logging.LogRecord))([LogRecord](http://docs.google.com/java/util/logging/LogRecord.html) record)            Publish a LogRecord. |
| protected  void | [**reportError**](http://docs.google.com/java/util/logging/Handler.html#reportError(java.lang.String,%20java.lang.Exception,%20int))([String](http://docs.google.com/java/lang/String.html) msg, [Exception](http://docs.google.com/java/lang/Exception.html) ex, int code)            Protected convenience method to report an error to this Handler's ErrorManager. |
| void | [**setEncoding**](http://docs.google.com/java/util/logging/Handler.html#setEncoding(java.lang.String))([String](http://docs.google.com/java/lang/String.html) encoding)            Set the character encoding used by this Handler. |
| void | [**setErrorManager**](http://docs.google.com/java/util/logging/Handler.html#setErrorManager(java.util.logging.ErrorManager))([ErrorManager](http://docs.google.com/java/util/logging/ErrorManager.html) em)            Define an ErrorManager for this Handler. |
| void | [**setFilter**](http://docs.google.com/java/util/logging/Handler.html#setFilter(java.util.logging.Filter))([Filter](http://docs.google.com/java/util/logging/Filter.html) newFilter)            Set a Filter to control output on this Handler. |
| void | [**setFormatter**](http://docs.google.com/java/util/logging/Handler.html#setFormatter(java.util.logging.Formatter))([Formatter](http://docs.google.com/java/util/logging/Formatter.html) newFormatter)            Set a Formatter. |
| void | [**setLevel**](http://docs.google.com/java/util/logging/Handler.html#setLevel(java.util.logging.Level))([Level](http://docs.google.com/java/util/logging/Level.html) newLevel)            Set the log level specifying which message levels will be logged by this Handler. |

| **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** |
| --- |

### Handler

protected **Handler**()

Default constructor. The resulting Handler has a log level of Level.ALL, no Formatter, and no Filter. A default ErrorManager instance is installed as the ErrorManager.

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

### publish

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

Publish a LogRecord.

The logging request was made initially to a Logger object, which initialized the LogRecord and forwarded it here.

The Handler is responsible for formatting the message, when and if necessary. The formatting should include localization.

**Parameters:**record - description of the log event. A null record is silently ignored and is not published

### flush

public abstract void **flush**()

Flush any buffered output.

### close

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

Close the Handler and free all associated resources.

The close method will perform a flush and then close the Handler. After close has been called this Handler should no longer be used. Method calls may either be silently ignored or may throw runtime exceptions.

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

### setFormatter

public void **setFormatter**([Formatter](http://docs.google.com/java/util/logging/Formatter.html) newFormatter)  
 throws [SecurityException](http://docs.google.com/java/lang/SecurityException.html)

Set a Formatter. This Formatter will be used to format LogRecords for this Handler.

Some Handlers may not use Formatters, in which case the Formatter will be remembered, but not used.

**Parameters:**newFormatter - the Formatter to use (may not be null) **Throws:** [SecurityException](http://docs.google.com/java/lang/SecurityException.html) - if a security manager exists and if the caller does not have LoggingPermission("control").

### getFormatter

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

Return the Formatter for this Handler.

**Returns:**the Formatter (may be null).

### setEncoding

public void **setEncoding**([String](http://docs.google.com/java/lang/String.html) encoding)  
 throws [SecurityException](http://docs.google.com/java/lang/SecurityException.html),  
 [UnsupportedEncodingException](http://docs.google.com/java/io/UnsupportedEncodingException.html)

Set the character encoding used by this Handler.

The encoding should be set before any LogRecords are written to the Handler.

**Parameters:**encoding - The name of a supported character encoding. May be null, to indicate the default platform encoding. **Throws:** [SecurityException](http://docs.google.com/java/lang/SecurityException.html) - if a security manager exists and if the caller does not have LoggingPermission("control"). [UnsupportedEncodingException](http://docs.google.com/java/io/UnsupportedEncodingException.html) - if the named encoding is not supported.

### getEncoding

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

Return the character encoding for this Handler.

**Returns:**The encoding name. May be null, which indicates the default encoding should be used.

### setFilter

public void **setFilter**([Filter](http://docs.google.com/java/util/logging/Filter.html) newFilter)  
 throws [SecurityException](http://docs.google.com/java/lang/SecurityException.html)

Set a Filter to control output on this Handler.

For each call of publish the Handler will call this Filter (if it is non-null) to check if the LogRecord should be published or discarded.

**Parameters:**newFilter - a Filter object (may be null) **Throws:** [SecurityException](http://docs.google.com/java/lang/SecurityException.html) - if a security manager exists and if the caller does not have LoggingPermission("control").

### getFilter

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

Get the current Filter for this Handler.

**Returns:**a Filter object (may be null)

### setErrorManager

public void **setErrorManager**([ErrorManager](http://docs.google.com/java/util/logging/ErrorManager.html) em)

Define an ErrorManager for this Handler.

The ErrorManager's "error" method will be invoked if any errors occur while using this Handler.

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

### getErrorManager

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

Retrieves the ErrorManager for this Handler.

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

### reportError

protected void **reportError**([String](http://docs.google.com/java/lang/String.html) msg,  
 [Exception](http://docs.google.com/java/lang/Exception.html) ex,  
 int code)

Protected convenience method to report an error to this Handler's ErrorManager. Note that this method retrieves and uses the ErrorManager without doing a security check. It can therefore be used in environments where the caller may be non-privileged.

**Parameters:**msg - a descriptive string (may be null)ex - an exception (may be null)code - an error code defined in ErrorManager

### setLevel

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

Set the log level specifying which message levels will be logged by this Handler. Message levels lower than this value will be discarded.

The intention is to allow developers to turn on voluminous logging, but to limit the messages that are sent to certain Handlers.

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

### getLevel

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

Get the log level specifying which messages will be logged by this Handler. Message levels lower than this level will be discarded.

**Returns:**the level of messages being logged.

### 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.

This method checks if the LogRecord has an appropriate Level and whether it satisfies any Filter. It also may make other Handler specific checks that might prevent a handler from logging the LogRecord. It will return false if the LogRecord is Null.

**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/Handler.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/Formatter.html)   [**NEXT CLASS**](http://docs.google.com/java/util/logging/Level.html) | [**FRAMES**](http://docs.google.com/index.html?java/util/logging/Handler.html)    [**NO FRAMES**](http://docs.google.com/Handler.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | FIELD | [CONSTR](#3znysh7) | [METHOD](#2et92p0) | DETAIL: FIELD | [CONSTR](#3dy6vkm) | [METHOD](#4d34og8) |

[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).