

# :mod:`logging.handlers` --- Logging handlers

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library] logging.handlers.rst, line 1); [backlink](#)

Unknown interpreted text role "mod".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library] logging.handlers.rst, line 4)

Unknown directive type "module".

```
.. module:: logging.handlers
   :synopsis: Handlers for the logging module.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library] logging.handlers.rst, line 7)

Unknown directive type "moduleauthor".

```
.. moduleauthor:: Vinay Sajip <vinay_sajip@red-dove.com>
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library] logging.handlers.rst, line 8)

Unknown directive type "sectionauthor".

```
.. sectionauthor:: Vinay Sajip <vinay_sajip@red-dove.com>
```

Source code: :source:`Lib/logging/handlers.py`

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library] logging.handlers.rst, line 10); [backlink](#)

Unknown interpreted text role "source".

## Important

This page contains only reference information. For tutorials, please see

- :ref:`Basic Tutorial <logging-basic-tutorial>`

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library] logging.handlers.rst, line 17); [backlink](#)

Unknown interpreted text role "ref".

- :ref:`Advanced Tutorial <logging-advanced-tutorial>`

**System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\ [cpython-main] [Doc] [library] logging.handlers.rst, line 18); [backlink](#)  
Unknown interpreted text role "ref".

- `:ref: Logging Cookbook <logging-cookbook>`

**System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\ [cpython-main] [Doc] [library] logging.handlers.rst, line 19); [backlink](#)  
Unknown interpreted text role "ref".

**System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\ [cpython-main] [Doc] [library] logging.handlers.rst, line 23)

Unknown directive type "currentmodule".

```
.. currentmodule:: logging
```

The following useful handlers are provided in the package. Note that three of the handlers (`:class: 'StreamHandler'`, `:class: 'FileHandler'` and `:class: 'NullHandler'`) are actually defined in the `:mod: 'logging'` module itself, but have been documented here along with the other handlers.

**System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\ [cpython-main] [Doc] [library] logging.handlers.rst, line 25); [backlink](#)

Unknown interpreted text role "class".

**System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\ [cpython-main] [Doc] [library] logging.handlers.rst, line 25); [backlink](#)

Unknown interpreted text role "class".

**System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\ [cpython-main] [Doc] [library] logging.handlers.rst, line 25); [backlink](#)

Unknown interpreted text role "class".

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library] logging.handlers.rst, line 25); [backlink](#)**

Unknown interpreted text role "mod".

## StreamHandler

The `:class:`StreamHandler`` class, located in the core `:mod:`logging`` package, sends logging output to streams such as `sys.stdout`, `sys.stderr` or any file-like object (or, more precisely, any object which supports `:meth:`write`` and `:meth:`flush`` methods).

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library] logging.handlers.rst, line 35); [backlink](#)**

Unknown interpreted text role "class".

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library] logging.handlers.rst, line 35); [backlink](#)**

Unknown interpreted text role "mod".

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library] logging.handlers.rst, line 35); [backlink](#)**

Unknown interpreted text role "meth".

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library] logging.handlers.rst, line 35); [backlink](#)**

Unknown interpreted text role "meth".

Returns a new instance of the `:class:`StreamHandler`` class. If *stream* is specified, the instance will use it for logging output; otherwise, `sys.stderr` will be used.

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library] logging.handlers.rst, line 43); [backlink](#)**

Unknown interpreted text role "class".

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library] logging.handlers.rst, line 48)**

Unknown directive type "method".

```
.. method:: emit(record)
```

If a formatter is specified, it is used to format the record. The record is then written to the stream followed by `:attr:`terminator``. If exception information is present, it is formatted using `:func:`traceback.print_exception`` and appended to the stream.

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library] logging.handlers.rst, line 56)**

Unknown directive type "method".

```
.. method:: flush()
```

Flushes the stream by calling its `:meth:`flush`` method. Note that the `:meth:`close`` method is inherited from `:class:`~logging.Handler`` and so does no output, so an explicit `:meth:`flush`` call may be needed at times.

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library] logging.handlers.rst, line 62)**

Unknown directive type "method".

```
.. method:: setStream(stream)
```

Sets the instance's stream to the specified value, if it is different.

```
The old stream is flushed before the new stream is set.

:param stream: The stream that the handler should use.

:return: the old stream, if the stream was changed, or *None* if it wasn't.

.. versionadded:: 3.7
```

**System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library] logging.handlers.rst, line 73)

Unknown directive type "attribute".

```
.. attribute:: terminator

String used as the terminator when writing a formatted record to a stream.
Default value is ``'\n'``.

If you don't want a newline termination, you can set the handler instance's
``terminator`` attribute to the empty string.

In earlier versions, the terminator was hardcoded as ``'\n'``.

.. versionadded:: 3.2
```

## FileHandler

The `:class:`FileHandler`` class, located in the core `:mod:`logging`` package, sends logging output to a disk file. It inherits the output functionality from `:class:`StreamHandler``.

**System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library] logging.handlers.rst, line 91); [backlink](#)

Unknown interpreted text role "class".

**System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library] logging.handlers.rst, line 91); [backlink](#)

Unknown interpreted text role "mod".

**System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library] logging.handlers.rst, line 91); [backlink](#)

Unknown interpreted text role "class".

Returns a new instance of the `:class:`FileHandler`` class. The specified file is opened and used as the stream for logging. If `mode` is not specified, `:const:`a`` is used. If `encoding` is not `None`, it is used to open the file with that encoding. If `delay` is true, then file opening is deferred until the first call to `:meth:`emit``. By default, the file grows indefinitely. If `errors` is specified, it's used to determine how encoding errors are handled.

**System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library] logging.handlers.rst, line 98); [backlink](#)

Unknown interpreted text role "class".

**System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library] logging.handlers.rst, line 98); [backlink](#)

Unknown interpreted text role "const".

**System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library] logging.handlers.rst, line 98); [backlink](#)

Unknown interpreted text role "meth".

**System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library] logging.handlers.rst, line 105)

Unknown directive type "versionchanged".

```
.. versionchanged:: 3.6
    As well as string values, :class:`~pathlib.Path` objects are also accepted
    for the *filename* argument.
```

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library]logging.handlers.rst, line 109)**

Unknown directive type "versionchanged".

```
.. versionchanged:: 3.9
    The *errors* parameter was added.
```

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library]logging.handlers.rst, line 112)**

Unknown directive type "method".

```
.. method:: close()

    Closes the file.
```

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library]logging.handlers.rst, line 116)**

Unknown directive type "method".

```
.. method:: emit(record)

    Outputs the record to the file.

    Note that if the file was closed due to logging shutdown at exit and the file
    mode is 'w', the record will not be emitted (see :issue:`42378`).
```

## NullHandler

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library]logging.handlers.rst, line 129)**

Unknown directive type "versionadded".

```
.. versionadded:: 3.1
```

The :class:`NullHandler` class, located in the core :mod:`logging` package, does not do any formatting or output. It is essentially a 'no-op' handler for use by library developers.

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library]logging.handlers.rst, line 131); [backlink](#)**

Unknown interpreted text role "class".

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library]logging.handlers.rst, line 131); [backlink](#)**

Unknown interpreted text role "mod".

Returns a new instance of the :class:`NullHandler` class.

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library]logging.handlers.rst, line 137); [backlink](#)**

Unknown interpreted text role "class".

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library]logging.handlers.rst, line 139)**

Unknown directive type "method".

```
.. method:: emit(record)
```

```
This method does nothing.
```

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library] logging.handlers.rst, line 143)**

Unknown directive type "method".

```
.. method:: handle(record)

    This method does nothing.
```

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library] logging.handlers.rst, line 147)**

Unknown directive type "method".

```
.. method:: createLock()

    This method returns ``None`` for the lock, since there is no
    underlying I/O to which access needs to be serialized.
```

See [:ref:'library-config'](#) for more information on how to use [:class:'NullHandler'](#).

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library] logging.handlers.rst, line 153); [backlink](#)**

Unknown interpreted text role "ref".

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library] logging.handlers.rst, line 153); [backlink](#)**

Unknown interpreted text role "class".

## WatchedFileHandler

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library] logging.handlers.rst, line 161)**

Unknown directive type "currentmodule".

```
.. currentmodule:: logging.handlers
```

The [:class:'WatchedFileHandler'](#) class, located in the [:mod:'logging.handlers'](#) module, is a [:class:'FileHandler'](#) which watches the file it is logging to. If the file changes, it is closed and reopened using the file name.

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library] logging.handlers.rst, line 163); [backlink](#)**

Unknown interpreted text role "class".

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library] logging.handlers.rst, line 163); [backlink](#)**

Unknown interpreted text role "mod".

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library] logging.handlers.rst, line 163); [backlink](#)**

Unknown interpreted text role "class".

A file change can happen because of usage of programs such as *newsyslog* and *logrotate* which perform log file rotation. This handler, intended for use under Unix/Linux, watches the file to see if it has changed since the last emit. (A file is deemed to have changed if its device or inode have changed.) If the file has changed, the old file stream is closed, and the file opened to get a new stream.

This handler is not appropriate for use under Windows, because under Windows open log files cannot be moved or renamed - logging opens the files with exclusive locks - and so there is no need for such a handler. Furthermore, *ST\_INO* is not supported under

Windows; `:func:`~os.stat`` always returns zero for this value.

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library] logging.handlers.rst, line 174); [backlink](#)**

Unknown interpreted text role "func".

Returns a new instance of the `:class:`WatchedFileHandler`` class. The specified file is opened and used as the stream for logging. If *mode* is not specified, `:const:`a`` is used. If *encoding* is not `None`, it is used to open the file with that encoding. If *delay* is true, then file opening is deferred until the first call to `:meth:`emit``. By default, the file grows indefinitely. If *errors* is provided, it determines how encoding errors are handled.

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library] logging.handlers.rst, line 183); [backlink](#)**

Unknown interpreted text role "class".

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library] logging.handlers.rst, line 183); [backlink](#)**

Unknown interpreted text role "const".

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library] logging.handlers.rst, line 183); [backlink](#)**

Unknown interpreted text role "meth".

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library] logging.handlers.rst, line 190)**

Unknown directive type "versionchanged".

```
.. versionchanged:: 3.6
   As well as string values, :class:`~pathlib.Path` objects are also accepted
   for the *filename* argument.
```

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library] logging.handlers.rst, line 194)**

Unknown directive type "versionchanged".

```
.. versionchanged:: 3.9
   The *errors* parameter was added.
```

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library] logging.handlers.rst, line 197)**

Unknown directive type "method".

```
.. method:: reopenIfNeeded()

   Checks to see if the file has changed. If it has, the existing stream is
   flushed and closed and the file opened again, typically as a precursor to
   outputting the record to the file.

.. versionadded:: 3.6
```

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library] logging.handlers.rst, line 206)**

Unknown directive type "method".

```
.. method:: emit(record)

   Outputs the record to the file, but first calls :meth:`reopenIfNeeded` to
   reopen the file if it has changed.
```

The `:class:'BaseRotatingHandler'` class, located in the `:mod:'logging.handlers'` module, is the base class for the rotating file handlers, `:class:'RotatingFileHandler'` and `:class:'TimedRotatingFileHandler'`. You should not need to instantiate this class, but it has attributes and methods you may need to override.

**System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library]logging.handlers.rst, line 216); [backlink](#)

Unknown interpreted text role "class".

**System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library]logging.handlers.rst, line 216); [backlink](#)

Unknown interpreted text role "mod".

**System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library]logging.handlers.rst, line 216); [backlink](#)

Unknown interpreted text role "class".

**System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library]logging.handlers.rst, line 216); [backlink](#)

Unknown interpreted text role "class".

The parameters are as for `:class:'FileHandler'`. The attributes are:

**System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library]logging.handlers.rst, line 224); [backlink](#)

Unknown interpreted text role "class".

**System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library]logging.handlers.rst, line 226)

Unknown directive type "attribute".

```
.. attribute:: namer
```

If this attribute is set to a callable, the `:meth:`rotation_filename`` method delegates to this callable. The parameters passed to the callable are those passed to `:meth:`rotation_filename``.

.. note:: The namer function is called quite a few times during rollover, so it should be as simple and as fast as possible. It should also return the same output every time for a given input, otherwise the rollover behaviour may not work as expected.

It's also worth noting that care should be taken when using a namer to preserve certain attributes in the filename which are used during rotation. For example, `:class:`RotatingFileHandler`` expects to have a set of log files whose names contain successive integers, so that rotation works as expected, and `:class:`TimedRotatingFileHandler`` deletes old log files (based on the ``backupCount`` parameter passed to the handler's initializer) by determining the oldest files to delete. For this to happen, the filenames should be sortable using the date/time portion of the filename, and a namer needs to respect this. (If a namer is wanted that doesn't respect this scheme, it will need to be used in a subclass of `:class:`TimedRotatingFileHandler`` which overrides the `:meth:`~TimedRotatingFileHandler.getFilesToDelete`` method to fit in with the custom naming scheme.)

```
.. versionadded:: 3.3
```

**System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library]logging.handlers.rst, line 253)

Unknown directive type "attribute".

```
.. attribute:: BaseRotatingHandler.rotator
```

If this attribute is set to a callable, the `:meth:`rotate`` method delegates to this callable. The parameters passed to the callable are those passed to `:meth:`rotate``.



```
.. versionadded:: 3.3
```

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library]logging.handlers.rst, line 261)**

Unknown directive type "method".

```
.. method:: BaseRotatingHandler.rotation_filename(default_name)
```

Modify the filename of a log file when rotating.

This is provided so that a custom filename can be provided.

The default implementation calls the 'namer' attribute of the handler, if it's callable, passing the default name to it. If the attribute isn't callable (the default is ``None``), the name is returned unchanged.

:param default\_name: The default name for the log file.

```
.. versionadded:: 3.3
```

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library]logging.handlers.rst, line 276)**

Unknown directive type "method".

```
.. method:: BaseRotatingHandler.rotate(source, dest)
```

When rotating, rotate the current log.

The default implementation calls the 'rotator' attribute of the handler, if it's callable, passing the source and dest arguments to it. If the attribute isn't callable (the default is ``None``), the source is simply renamed to the destination.

:param source: The source filename. This is normally the base filename, e.g. 'test.log'.

:param dest: The destination filename. This is normally what the source is rotated to, e.g. 'test.log.1'.

```
.. versionadded:: 3.3
```

The reason the attributes exist is to save you having to subclass - you can use the same callables for instances of `:class:RotatingFileHandler` and `:class:TimedRotatingFileHandler`. If either the namer or rotator callable raises an exception, this will be handled in the same way as any other exception during an `meth:emit` call, i.e. via the `meth:handleError` method of the handler.

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library]logging.handlers.rst, line 292); [backlink](#)**

Unknown interpreted text role "class".

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library]logging.handlers.rst, line 292); [backlink](#)**

Unknown interpreted text role "class".

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library]logging.handlers.rst, line 292); [backlink](#)**

Unknown interpreted text role "meth".

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library]logging.handlers.rst, line 292); [backlink](#)**

Unknown interpreted text role "meth".

If you need to make more significant changes to rotation processing, you can override the methods.

For an example, see `ref:cookbook-rotator-namer`.

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-**

main\Doc\library\[cpython-main] [Doc] [library] logging.handlers.rst, line 302); [backlink](#)

Unknown interpreted text role "ref".

## RotatingFileHandler

The `:class:`RotatingFileHandler`` class, located in the `:mod:`logging.handlers`` module, supports rotation of disk log files.

**System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library] logging.handlers.rst, line 310); [backlink](#)

Unknown interpreted text role "class".

**System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library] logging.handlers.rst, line 310); [backlink](#)

Unknown interpreted text role "mod".

Returns a new instance of the `:class:`RotatingFileHandler`` class. The specified file is opened and used as the stream for logging. If *mode* is not specified, 'a' is used. If *encoding* is not None, it is used to open the file with that encoding. If *delay* is true, then file opening is deferred until the first call to `:meth:`emit``. By default, the file grows indefinitely. If *errors* is provided, it determines how encoding errors are handled.

**System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library] logging.handlers.rst, line 316); [backlink](#)

Unknown interpreted text role "class".

**System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library] logging.handlers.rst, line 316); [backlink](#)

Unknown interpreted text role "meth".

You can use the *maxBytes* and *backupCount* values to allow the file to `:dfn:`rollover`` at a predetermined size. When the size is about to be exceeded, the file is closed and a new file is silently opened for output. Rollover occurs whenever the current log file is nearly *maxBytes* in length; but if either of *maxBytes* or *backupCount* is zero, rollover never occurs, so you generally want to set *backupCount* to at least 1, and have a non-zero *maxBytes*. When *backupCount* is non-zero, the system will save old log files by appending the extensions '.1', '.2' etc., to the filename. For example, with a *backupCount* of 5 and a base file name of `:file:`app.log``, you would get `:file:`app.log``, `:file:`app.log.1``, `:file:`app.log.2``, up to `:file:`app.log.5``. The file being written to is always `:file:`app.log``. When this file is filled, it is closed and renamed to `:file:`app.log.1``, and if files `:file:`app.log.1``, `:file:`app.log.2``, etc. exist, then they are renamed to `:file:`app.log.2``, `:file:`app.log.3`` etc. respectively.

**System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library] logging.handlers.rst, line 323); [backlink](#)

Unknown interpreted text role "dfn".

**System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library] logging.handlers.rst, line 323); [backlink](#)

Unknown interpreted text role "file".

**System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library] logging.handlers.rst, line 323); [backlink](#)

Unknown interpreted text role "file".

**System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library] logging.handlers.rst, line 323); [backlink](#)

Unknown interpreted text role "file".

**System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library] logging.handlers.rst, line 323); [backlink](#)

Unknown interpreted text role "file".

**System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library] logging.handlers.rst, line 323); [backlink](#)

Unknown interpreted text role "file".

**System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library] logging.handlers.rst, line 323); [backlink](#)

Unknown interpreted text role "file".

**System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library] logging.handlers.rst, line 323); [backlink](#)

Unknown interpreted text role "file".

**System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library] logging.handlers.rst, line 323); [backlink](#)

Unknown interpreted text role "file".

**System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library] logging.handlers.rst, line 323); [backlink](#)

Unknown interpreted text role "file".

**System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library] logging.handlers.rst, line 323); [backlink](#)

Unknown interpreted text role "file".

**System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library] logging.handlers.rst, line 323); [backlink](#)

Unknown interpreted text role "file".

**System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library] logging.handlers.rst, line 338)

Unknown directive type "versionchanged".

```
.. versionchanged:: 3.6
   As well as string values, :class:`~pathlib.Path` objects are also accepted
   for the *filename* argument.
```

**System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library] logging.handlers.rst, line 342)

Unknown directive type "versionchanged".

```
.. versionchanged:: 3.9
   The *errors* parameter was added.
```

**System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library] logging.handlers.rst, line 345)

Unknown directive type "method".

```
.. method:: doRollover()

   Does a rollover, as described above.
```

**System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library] logging.handlers.rst, line 350)

Unknown directive type "method".

```
.. method:: emit(record)
```

Outputs the record to the file, catering for rollover as described previously.

## TimedRotatingFileHandler

The `class: 'TimedRotatingFileHandler'` class, located in the `mod: 'logging.handlers'` module, supports rotation of disk log files at certain timed intervals.

**System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library] logging.handlers.rst, line 360); [backlink](#)

Unknown interpreted text role "class".

**System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library] logging.handlers.rst, line 360); [backlink](#)

Unknown interpreted text role "mod".

Returns a new instance of the `class: 'TimedRotatingFileHandler'` class. The specified file is opened and used as the stream for logging. On rotating it also sets the filename suffix. Rotating happens based on the product of *when* and *interval*.

**System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library] logging.handlers.rst, line 367); [backlink](#)

Unknown interpreted text role "class".

You can use the *when* to specify the type of *interval*. The list of possible values is below. Note that they are not case sensitive.

Value	Type of interval	If/how <i>atTime</i> is used
'S'	Seconds	Ignored
'M'	Minutes	Ignored
'H'	Hours	Ignored
'D'	Days	Ignored
'W0'-'W6'	Weekday (0=Monday)	Used to compute initial rollover time
'midnight'	Roll over at midnight, if <i>atTime</i> not specified, else at time <i>atTime</i>	Used to compute initial rollover time

When using weekday-based rotation, specify 'W0' for Monday, 'W1' for Tuesday, and so on up to 'W6' for Sunday. In this case, the value passed for *interval* isn't used.

The system will save old log files by appending extensions to the filename. The extensions are date-and-time based, using the strftime format `%Y-%m-%d_%H-%M-%S` or a leading portion thereof, depending on the rollover interval.

When computing the next rollover time for the first time (when the handler is created), the last modification time of an existing log file, or else the current time, is used to compute when the next rotation will occur.

If the *utc* argument is true, times in UTC will be used; otherwise local time is used.

If *backupCount* is nonzero, at most *backupCount* files will be kept, and if more would be created when rollover occurs, the oldest one is deleted. The deletion logic uses the interval to determine which files to delete, so changing the interval may leave old files lying around.

If *delay* is true, then file opening is deferred until the first call to `meth:'emit'`.

**System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library] logging.handlers.rst, line 415); [backlink](#)

Unknown interpreted text role "meth".

If *atTime* is not `None`, it must be a `datetime.time` instance which specifies the time of day when rollover occurs, for the cases where rollover is set to happen "at midnight" or "on a particular weekday". Note that in these cases, the *atTime* value is effectively used to compute the *initial* rollover, and subsequent rollovers would be calculated via the normal interval calculation.

If *errors* is specified, it's used to determine how encoding errors are handled.

### Note

Calculation of the initial rollover time is done when the handler is initialised. Calculation of subsequent rollover times is done only when rollover occurs, and rollover occurs only when emitting output. If this is not kept in mind, it might lead to some confusion. For example, if an interval of "every minute" is set, that does not mean you will always see log files with times (in the filename) separated by a minute; if, during application execution, logging output is generated

more frequently than once a minute, *then* you can expect to see log files with times separated by a minute. If, on the other hand, logging messages are only output once every five minutes (say), then there will be gaps in the file times corresponding to the minutes where no output (and hence no rollover) occurred.

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library] logging.handlers.rst, line 441)**

Unknown directive type "versionchanged".

```
.. versionchanged:: 3.4
   *atTime* parameter was added.
```

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library] logging.handlers.rst, line 444)**

Unknown directive type "versionchanged".

```
.. versionchanged:: 3.6
   As well as string values, :class:`~pathlib.Path` objects are also accepted
   for the *filename* argument.
```

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library] logging.handlers.rst, line 448)**

Unknown directive type "versionchanged".

```
.. versionchanged:: 3.9
   The *errors* parameter was added.
```

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library] logging.handlers.rst, line 451)**

Unknown directive type "method".

```
.. method:: doRollover()

   Does a rollover, as described above.
```

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library] logging.handlers.rst, line 455)**

Unknown directive type "method".

```
.. method:: emit(record)

   Outputs the record to the file, catering for rollover as described above.
```

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library] logging.handlers.rst, line 459)**

Unknown directive type "method".

```
.. method:: getFilesToDelete()

   Returns a list of filenames which should be deleted as part of rollover. These
   are the absolute paths of the oldest backup log files written by the handler.
```

## SocketHandler

The `:class:'SocketHandler'` class, located in the `:mod:'logging.handlers'` module, sends logging output to a network socket. The base class uses a TCP socket.

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library] logging.handlers.rst, line 469); [backlink](#)**

Unknown interpreted text role "class".

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-**

main\Doc\library\[cpython-main] [Doc] [library]logging.handlers.rst, line 469); [backlink](#)

Unknown interpreted text role "mod".

Returns a new instance of the `:class:`SocketHandler`` class intended to communicate with a remote machine whose address is given by *host* and *port*.

**System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library]logging.handlers.rst, line 475); [backlink](#)

Unknown interpreted text role "class".

**System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library]logging.handlers.rst, line 478)

Unknown directive type "versionchanged".

```
.. versionchanged:: 3.4
   If ``port`` is specified as ``None``, a Unix domain socket is created
   using the value in ``host`` - otherwise, a TCP socket is created.
```

**System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library]logging.handlers.rst, line 482)

Unknown directive type "method".

```
.. method:: close()

   Closes the socket.
```

**System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library]logging.handlers.rst, line 487)

Unknown directive type "method".

```
.. method:: emit()

   Pickles the record's attribute dictionary and writes it to the socket in
   binary format. If there is an error with the socket, silently drops the
   packet. If the connection was previously lost, re-establishes the
   connection. To unpickle the record at the receiving end into a
   :class:`~logging.LogRecord`, use the :func:`~logging.makeLogRecord`
   function.
```

**System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library]logging.handlers.rst, line 497)

Unknown directive type "method".

```
.. method:: handleError()

   Handles an error which has occurred during :meth:`emit`. The most likely
   cause is a lost connection. Closes the socket so that we can retry on the
   next event.
```

**System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library]logging.handlers.rst, line 504)

Unknown directive type "method".

```
.. method:: makeSocket()

   This is a factory method which allows subclasses to define the precise
   type of socket they want. The default implementation creates a TCP socket
   (:const:`socket.SOCK_STREAM`).
```

**System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\cpython-

main\Doc\library\[cpython-main] [Doc] [library]logging.handlers.rst, line 511)

Unknown directive type "method".

```
.. method:: makePickle(record)
```

Pickles the record's attribute dictionary in binary format with a length prefix, and returns it ready for transmission across the socket. The details of this operation are equivalent to::

```
data = pickle.dumps(record_attr_dict, 1)
datalen = struct.pack('>L', len(data))
return datalen + data
```

Note that pickles aren't completely secure. If you are concerned about security, you may want to override this method to implement a more secure mechanism. For example, you can sign pickles using HMAC and then verify them on the receiving end, or alternatively you can disable unpickling of global objects on the receiving end.

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library]logging.handlers.rst, line 528)**

Unknown directive type "method".

```
.. method:: send(packet)
```

Send a pickled byte-string *\*packet\** to the socket. The format of the sent byte-string is as described in the documentation for :meth:`~SocketHandler.makePickle`.

This function allows for partial sends, which can happen when the network is busy.

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library]logging.handlers.rst, line 538)**

Unknown directive type "method".

```
.. method:: createSocket()
```

Tries to create a socket; on failure, uses an exponential back-off algorithm. On initial failure, the handler will drop the message it was trying to send. When subsequent messages are handled by the same instance, it will not try connecting until some time has passed. The default parameters are such that the initial delay is one second, and if after that delay the connection still can't be made, the handler will double the delay each time up to a maximum of 30 seconds.

This behaviour is controlled by the following handler attributes:

```
* ``retryStart`` (initial delay, defaulting to 1.0 seconds).
* ``retryFactor`` (multiplier, defaulting to 2.0).
* ``retryMax`` (maximum delay, defaulting to 30.0 seconds).
```

This means that if the remote listener starts up *after* the handler has been used, you could lose messages (since the handler won't even attempt a connection until the delay has elapsed, but just silently drop messages during the delay period).

## DatagramHandler

The :class:`DatagramHandler` class, located in the :mod:`logging.handlers` module, inherits from :class:`SocketHandler` to support sending logging messages over UDP sockets.

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library]logging.handlers.rst, line 565); [backlink](#)**

Unknown interpreted text role "class".

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library]logging.handlers.rst, line 565); [backlink](#)**

Unknown interpreted text role "mod".



**System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library] logging.handlers.rst, line 565); [backlink](#)

Unknown interpreted text role "class".

Returns a new instance of the `:class:`DatagramHandler`` class intended to communicate with a remote machine whose address is given by *host* and *port*.

**System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library] logging.handlers.rst, line 572); [backlink](#)

Unknown interpreted text role "class".

**System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library] logging.handlers.rst, line 575)

Unknown directive type "versionchanged".

```
.. versionchanged:: 3.4
   If ``port`` is specified as ``None``, a Unix domain socket is created
   using the value in ``host`` - otherwise, a UDP socket is created.
```

**System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library] logging.handlers.rst, line 579)

Unknown directive type "method".

```
.. method:: emit()
```

Pickles the record's attribute dictionary and writes it to the socket in binary format. If there is an error with the socket, silently drops the packet. To unpickle the record at the receiving end into a `:class:`~logging.LogRecord``, use the `:func:`~logging.makeLogRecord`` function.

**System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library] logging.handlers.rst, line 588)

Unknown directive type "method".

```
.. method:: makeSocket()
```

The factory method of `:class:`SocketHandler`` is here overridden to create a UDP socket (`:const:`socket.SOCK_DGRAM``).

**System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library] logging.handlers.rst, line 594)

Unknown directive type "method".

```
.. method:: send(s)
```

Send a pickled byte-string to a socket. The format of the sent byte-string is as described in the documentation for `:meth:`SocketHandler.makePickle``.

## SysLogHandler

The `:class:`SysLogHandler`` class, located in the `:mod:`logging.handlers`` module, supports sending logging messages to a remote or local Unix syslog.

**System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library] logging.handlers.rst, line 605); [backlink](#)

Unknown interpreted text role "class".

**System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\cpython-



main\Doc\library\[cpython-main] [Doc] [library] logging.handlers.rst, line 605); [backlink](#)

Unknown interpreted text role "mod".

Returns a new instance of the `:class:'SysLogHandler'` class intended to communicate with a remote Unix machine whose address is given by *address* in the form of a (host, port) tuple. If *address* is not specified, ('localhost', 514) is used. The address is used to open a socket. An alternative to providing a (host, port) tuple is providing an address as a string, for example '/dev/log'. In this case, a Unix domain socket is used to send the message to the syslog. If *facility* is not specified, `:const:'LOG_USER'` is used. The type of socket opened depends on the *socktype* argument, which defaults to `:const:'socket.SOCK_DGRAM'` and thus opens a UDP socket. To open a TCP socket (for use with the newer syslog daemons such as rsyslog), specify a value of `:const:'socket.SOCK_STREAM'`.

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library] logging.handlers.rst, line 611); [backlink](#)**

Unknown interpreted text role "class".

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library] logging.handlers.rst, line 611); [backlink](#)**

Unknown interpreted text role "const".

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library] logging.handlers.rst, line 611); [backlink](#)**

Unknown interpreted text role "const".

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library] logging.handlers.rst, line 611); [backlink](#)**

Unknown interpreted text role "const".

Note that if your server is not listening on UDP port 514, `:class:'SysLogHandler'` may appear not to work. In that case, check what address you should be using for a domain socket - it's system dependent. For example, on Linux it's usually '/dev/log' but on OS/X it's '/var/run/syslog'. You'll need to check your platform and use the appropriate address (you may need to do this check at runtime if your application needs to run on several platforms). On Windows, you pretty much have to use the UDP option.

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library] logging.handlers.rst, line 623); [backlink](#)**

Unknown interpreted text role "class".

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library] logging.handlers.rst, line 632)**

Unknown directive type "versionchanged".

```
.. versionchanged:: 3.2
   *socket* was added.
```

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library] logging.handlers.rst, line 636)**

Unknown directive type "method".

```
.. method:: close()

Closes the socket to the remote host.
```

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library] logging.handlers.rst, line 641)**

Unknown directive type "method".

```
.. method:: emit(record)

The record is formatted, and then sent to the syslog server. If exception
information is present, it is *not* sent to the server.
```

```
.. versionchanged:: 3.2.1
    (See: :issue:`12168`.) In earlier versions, the message sent to the
    syslog daemons was always terminated with a NUL byte, because early
    versions of these daemons expected a NUL terminated message - even
    though it's not in the relevant specification (:rfc:`5424`). More recent
    versions of these daemons don't expect the NUL byte but strip it off
    if it's there, and even more recent daemons (which adhere more closely
    to RFC 5424) pass the NUL byte on as part of the message.
```

To enable easier handling of syslog messages in the face of all these differing daemon behaviours, the appending of the NUL byte has been made configurable, through the use of a class-level attribute, ``append\_nul``. This defaults to ``True`` (preserving the existing behaviour) but can be set to ``False`` on a ``SysLogHandler`` instance in order for that instance to \*not\* append the NUL terminator.

```
.. versionchanged:: 3.3
    (See: :issue:`12419`.) In earlier versions, there was no facility for
    an "ident" or "tag" prefix to identify the source of the message. This
    can now be specified using a class-level attribute, defaulting to
    ``""`` to preserve existing behaviour, but which can be overridden on
    a ``SysLogHandler`` instance in order for that instance to prepend
    the ident to every message handled. Note that the provided ident must
    be text, not bytes, and is prepended to the message exactly as is.
```

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library]logging.handlers.rst, line 671)**

Unknown directive type "method".

```
.. method:: encodePriority(facility, priority)
```

Encodes the facility and priority into an integer. You can pass in strings or integers - if strings are passed, internal mapping dictionaries are used to convert them to integers.

The symbolic ``LOG`` values are defined in :class:`SysLogHandler` and mirror the values defined in the ``sys/syslog.h`` header file.

**\*\*Priorities\*\***

Name (string)	Symbolic value
``alert``	LOG_ALERT
``crit`` or ``critical``	LOG_CRIT
``debug``	LOG_DEBUG
``emerg`` or ``panic``	LOG_EMERG
``err`` or ``error``	LOG_ERR
``info``	LOG_INFO
``notice``	LOG_NOTICE
``warn`` or ``warning``	LOG_WARNING

**\*\*Facilities\*\***

Name (string)	Symbolic value
``auth``	LOG_AUTH
``authpriv``	LOG_AUTHPRIV
``cron``	LOG_CRON
``daemon``	LOG_DAEMON
``ftp``	LOG_FTP
``kern``	LOG_KERN
``lpr``	LOG_LPR
``mail``	LOG_MAIL

``news``	LOG_NEWS
``syslog``	LOG_SYSLOG
``user``	LOG_USER
``uucp``	LOG_UUCP
``local0``	LOG_LOCAL0
``local1``	LOG_LOCAL1
``local2``	LOG_LOCAL2
``local3``	LOG_LOCAL3
``local4``	LOG_LOCAL4
``local5``	LOG_LOCAL5
``local6``	LOG_LOCAL6
``local7``	LOG_LOCAL7

**System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library]logging.handlers.rst, line 748)

Unknown directive type "method".

```
.. method:: mapPriority(levelname)
```

Maps a logging level name to a syslog priority name. You may need to override this if you are using custom levels, or if the default algorithm is not suitable for your needs. The default algorithm maps ``DEBUG``, ``INFO``, ``WARNING``, ``ERROR`` and ``CRITICAL`` to the equivalent syslog names, and all other level names to 'warning'.

## NTEventLogHandler

The `:class:`NTEventLogHandler`` class, located in the `:mod:`logging.handlers`` module, supports sending logging messages to a local Windows NT, Windows 2000 or Windows XP event log. Before you can use it, you need Mark Hammond's Win32 extensions for Python installed.

**System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library]logging.handlers.rst, line 762); [backlink](#)

Unknown interpreted text role "class".

**System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library]logging.handlers.rst, line 762); [backlink](#)

Unknown interpreted text role "mod".

Returns a new instance of the `:class:`NTEventLogHandler`` class. The *appname* is used to define the application name as it appears in the event log. An appropriate registry entry is created using this name. The *dllname* should give the fully qualified pathname of a .dll or .exe which contains message definitions to hold in the log (if not specified, 'win32service.pyd' is used - this is installed with the Win32 extensions and contains some basic placeholder message definitions. Note that use of these placeholders will make your event logs big, as the entire message source is held in the log. If you want slimmer logs, you have to pass in the name of your own .dll or .exe which contains the message definitions you want to use in the event log). The *logtype* is one of 'Application', 'System' or 'Security', and defaults to 'Application'.

**System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library]logging.handlers.rst, line 770); [backlink](#)

Unknown interpreted text role "class".

**System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library]logging.handlers.rst, line 784)

Unknown directive type "method".

```
.. method:: close()
```

At this point, you can remove the application name from the registry as a source of event log entries. However, if you do this, you will not be able to see the events as you intended in the Event Log Viewer - it needs to be able to access the registry to get the .dll name. The current version does not do this.

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library]logging.handlers.rst, line 793)**

Unknown directive type "method".

```
.. method:: emit(record)
```

Determines the message ID, event category and event type, and then logs the message in the NT event log.

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library]logging.handlers.rst, line 799)**

Unknown directive type "method".

```
.. method:: getCategory(record)
```

Returns the event category for the record. Override this if you want to specify your own categories. This version returns 0.

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library]logging.handlers.rst, line 805)**

Unknown directive type "method".

```
.. method:: getEventType(record)
```

Returns the event type for the record. Override this if you want to specify your own types. This version does a mapping using the handler's typemap attribute, which is set up in :meth:`\_\_init\_\_` to a dictionary which contains mappings for :const:`DEBUG`, :const:`INFO`, :const:`WARNING`, :const:`ERROR` and :const:`CRITICAL`. If you are using your own levels, you will either need to override this method or place a suitable dictionary in the handler's \*typemap\* attribute.

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library]logging.handlers.rst, line 816)**

Unknown directive type "method".

```
.. method:: getMessageID(record)
```

Returns the message ID for the record. If you are using your own messages, you could do this by having the \*msg\* passed to the logger being an ID rather than a format string. Then, in here, you could use a dictionary lookup to get the message ID. This version returns 1, which is the base message ID in :file:`win32service.pyd`.

## SMTPHandler

The :class:`SMTPHandler` class, located in the :mod:`logging.handlers` module, supports sending logging messages to an email address via SMTP.

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library]logging.handlers.rst, line 829); [backlink](#)**

Unknown interpreted text role "class".

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-**

main\Doc\library\[cpython-main] [Doc] [library] logging.handlers.rst, line 829); [backlink](#)

Unknown interpreted text role "mod".

Returns a new instance of the `:class:'SMTPHandler'` class. The instance is initialized with the from and to addresses and subject line of the email. The *toaddrs* should be a list of strings. To specify a non-standard SMTP port, use the (host, port) tuple format for the *mailhost* argument. If you use a string, the standard SMTP port is used. If your SMTP server requires authentication, you can specify a (username, password) tuple for the *credentials* argument.

**System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library] logging.handlers.rst, line 835); [backlink](#)

Unknown interpreted text role "class".

To specify the use of a secure protocol (TLS), pass in a tuple to the *secure* argument. This will only be used when authentication credentials are supplied. The tuple should be either an empty tuple, or a single-value tuple with the name of a keyfile, or a 2-value tuple with the names of the keyfile and certificate file. (This tuple is passed to the `:meth:'smtplib.SMTP.starttls'` method.)

**System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library] logging.handlers.rst, line 842); [backlink](#)

Unknown interpreted text role "meth".

A timeout can be specified for communication with the SMTP server using the *timeout* argument.

**System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library] logging.handlers.rst, line 852)

Unknown directive type "versionadded".

```
.. versionadded:: 3.3
   The *timeout* argument was added.
```

**System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library] logging.handlers.rst, line 855)

Unknown directive type "method".

```
.. method:: emit(record)

   Formats the record and sends it to the specified addressees.
```

**System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library] logging.handlers.rst, line 860)

Unknown directive type "method".

```
.. method:: getSubject(record)

   If you want to specify a subject line which is record-dependent, override
   this method.
```

## MemoryHandler

The `:class:'MemoryHandler'` class, located in the `:mod:'logging.handlers'` module, supports buffering of logging records in memory, periodically flushing them to a `:dfn:'target'` handler. Flushing occurs whenever the buffer is full, or when an event of a certain severity or greater is seen.

**System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library] logging.handlers.rst, line 870); [backlink](#)

Unknown interpreted text role "class".

**System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library] logging.handlers.rst, line 870); [backlink](#)

Unknown interpreted text role "mod".

**System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library] logging.handlers.rst, line 870); [backlink](#)

Unknown interpreted text role "dfn".

`:class:'MemoryHandler'` is a subclass of the more general `:class:'BufferingHandler'`, which is an abstract class. This buffers logging records in memory. Whenever each record is added to the buffer, a check is made by calling `:meth:'shouldFlush'` to see if the buffer should be flushed. If it should, then `:meth:'flush'` is expected to do the flushing.

**System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library] logging.handlers.rst, line 875); [backlink](#)

Unknown interpreted text role "class".

**System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library] logging.handlers.rst, line 875); [backlink](#)

Unknown interpreted text role "class".

**System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library] logging.handlers.rst, line 875); [backlink](#)

Unknown interpreted text role "meth".

**System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library] logging.handlers.rst, line 875); [backlink](#)

Unknown interpreted text role "meth".

Initializes the handler with a buffer of the specified capacity. Here, *capacity* means the number of logging records buffered.

**System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library] logging.handlers.rst, line 888)

Unknown directive type "method".

```
.. method:: emit(record)
```

Append the record to the buffer. If `:meth:'shouldFlush'` returns true, call `:meth:'flush'` to process the buffer.

**System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library] logging.handlers.rst, line 894)

Unknown directive type "method".

```
.. method:: flush()
```

You can override this to implement custom flushing behavior. This version just zaps the buffer to empty.

**System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library] logging.handlers.rst, line 900)

Unknown directive type "method".

```
.. method:: shouldFlush(record)
```

Return ```True``` if the buffer is up to capacity. This method can be overridden to implement custom flushing strategies.

Returns a new instance of the `:class:'MemoryHandler'` class. The instance is initialized with a buffer size of *capacity* (number of records buffered). If *flushLevel* is not specified, `:const:'ERROR'` is used. If no *target* is specified, the target will need to be set using `:meth:'setTarget'` before this handler does anything useful. If *flushOnClose* is specified as `False`, then the buffer is *not* flushed when the handler is closed. If not specified or specified as `True`, the previous behaviour of flushing the buffer will occur when the handler is closed.

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library] logging.handlers.rst, line 908); [backlink](#)**

Unknown interpreted text role "class".

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library] logging.handlers.rst, line 908); [backlink](#)**

Unknown interpreted text role "const".

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library] logging.handlers.rst, line 908); [backlink](#)**

Unknown interpreted text role "meth".

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library] logging.handlers.rst, line 917)**

Unknown directive type "versionchanged".

```
.. versionchanged:: 3.6
   The *flushOnClose* parameter was added.
```

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library] logging.handlers.rst, line 921)**

Unknown directive type "method".

```
.. method:: close()

   Calls :meth:`flush`, sets the target to ``None`` and clears the
   buffer.
```

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library] logging.handlers.rst, line 927)**

Unknown directive type "method".

```
.. method:: flush()

   For a :class:`MemoryHandler`, flushing means just sending the buffered
   records to the target, if there is one. The buffer is also cleared when
   this happens. Override if you want different behavior.
```

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library] logging.handlers.rst, line 934)**

Unknown directive type "method".

```
.. method:: setTarget(target)

   Sets the target handler for this handler.
```

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library] logging.handlers.rst, line 939)**

Unknown directive type "method".

```
.. method:: shouldFlush(record)

   Checks for buffer full or a record at the *flushLevel* or higher.
```

## HTTPHandler

The :class:`HTTPHandler` class, located in the :mod:`logging.handlers` module, supports sending logging messages to a web server,



using either GET or POST semantics.

**System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library] logging.handlers.rst, line 949); [backlink](#)

Unknown interpreted text role "class".

**System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library] logging.handlers.rst, line 949); [backlink](#)

Unknown interpreted text role "mod".

Returns a new instance of the `:class:`HTTPHandler`` class. The *host* can be of the form `host:port`, should you need to use a specific port number. If no *method* is specified, GET is used. If *secure* is true, a HTTPS connection will be used. The *context* parameter may be set to a `:class:`ssl.SSLContext`` instance to configure the SSL settings used for the HTTPS connection. If *credentials* is specified, it should be a 2-tuple consisting of *userid* and *password*, which will be placed in a HTTP 'Authorization' header using Basic authentication. If you specify *credentials*, you should also specify *secure=True* so that your *userid* and *password* are not passed in cleartext across the wire.

**System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library] logging.handlers.rst, line 956); [backlink](#)

Unknown interpreted text role "class".

**System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library] logging.handlers.rst, line 956); [backlink](#)

Unknown interpreted text role "class".

**System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library] logging.handlers.rst, line 967)

Unknown directive type "versionchanged".

```
.. versionchanged:: 3.5
   The *context* parameter was added.
```

**System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library] logging.handlers.rst, line 970)

Unknown directive type "method".

```
.. method:: mapLogRecord(record)

Provides a dictionary, based on ``record``, which is to be URL-encoded
and sent to the web server. The default implementation just returns
``record.__dict__``. This method can be overridden if e.g. only a
subset of :class:`~logging.LogRecord` is to be sent to the web server, or
if more specific customization of what's sent to the server is required.
```

**System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library] logging.handlers.rst, line 978)

Unknown directive type "method".

```
.. method:: emit(record)

Sends the record to the web server as a URL-encoded dictionary. The
:meth:`mapLogRecord` method is used to convert the record to the
dictionary to be sent.
```

#### Note

Since preparing a record for sending it to a web server is not the same as a generic formatting operation, using `:meth:`~logging.Handler.setFormatter`` to specify a `:class:`~logging.Formatter`` for a `:class:`HTTPHandler`` has no effect. Instead of calling `:meth:`~logging.Handler.format``, this handler calls `:meth:`mapLogRecord`` and then `:func:`urllib.parse.urlencode`` to encode the dictionary in a form suitable for sending to a web server.

**System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-



resources\cpython-main\Doc\library\[cpython-main] [Doc]  
[library] logging.handlers.rst, line 984); [backlink](#)

Unknown interpreted text role "meth".

**System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc]  
[library] logging.handlers.rst, line 984); [backlink](#)

Unknown interpreted text role "class".

**System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc]  
[library] logging.handlers.rst, line 984); [backlink](#)

Unknown interpreted text role "class".

**System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc]  
[library] logging.handlers.rst, line 984); [backlink](#)

Unknown interpreted text role "meth".

**System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc]  
[library] logging.handlers.rst, line 984); [backlink](#)

Unknown interpreted text role "meth".

**System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc]  
[library] logging.handlers.rst, line 984); [backlink](#)

Unknown interpreted text role "func".

## QueueHandler

**System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library] logging.handlers.rst, line 999)

Unknown directive type "versionadded".

```
.. versionadded:: 3.2
```

The `class:QueueHandler` class, located in the `mod:logging.handlers` module, supports sending logging messages to a queue, such as those implemented in the `mod:queue` or `mod:multiprocessing` modules.

**System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library] logging.handlers.rst, line 1001); [backlink](#)

Unknown interpreted text role "class".

**System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library] logging.handlers.rst, line 1001); [backlink](#)

Unknown interpreted text role "mod".

**System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library] logging.handlers.rst, line 1001); [backlink](#)

Unknown interpreted text role "mod".

**System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library] logging.handlers.rst, line 1001); [backlink](#)

Unknown interpreted text role "mod".

Along with the `:class:`QueueListener`` class, `:class:`QueueHandler`` can be used to let handlers do their work on a separate thread from the one which does the logging. This is important in web applications and also other service applications where threads servicing clients need to respond as quickly as possible, while any potentially slow operations (such as sending an email via `:class:`SMTPHandler``) are done on a separate thread.

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library] logging.handlers.rst, line 1005); [backlink](#)**

Unknown interpreted text role "class".

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library] logging.handlers.rst, line 1005); [backlink](#)**

Unknown interpreted text role "class".

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library] logging.handlers.rst, line 1005); [backlink](#)**

Unknown interpreted text role "class".

Returns a new instance of the `:class:`QueueHandler`` class. The instance is initialized with the queue to send messages to. The *queue* can be any queue-like object; it's used as-is by the `:meth:`enqueue`` method, which needs to know how to send messages to it. The queue is not *required* to have the task tracking API, which means that you can use `:class:`~queue.SimpleQueue`` instances for *queue*.

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library] logging.handlers.rst, line 1014); [backlink](#)**

Unknown interpreted text role "class".

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library] logging.handlers.rst, line 1014); [backlink](#)**

Unknown interpreted text role "meth".

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library] logging.handlers.rst, line 1014); [backlink](#)**

Unknown interpreted text role "class".

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library] logging.handlers.rst, line 1022)**

Unknown directive type "method".

```
.. method:: emit(record)
```

Enqueues the result of preparing the LogRecord. Should an exception occur (e.g. because a bounded queue has filled up), the `:meth:`~logging.Handler.handleError`` method is called to handle the error. This can result in the record silently being dropped (if `:attr:`~logging.raiseExceptions`` is `False`) or a message printed to `sys.stderr` (if `:attr:`~logging.raiseExceptions`` is `True`).

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library] logging.handlers.rst, line 1031)**

Unknown directive type "method".

```
.. method:: prepare(record)
```

Prepares a record for queuing. The object returned by this method is enqueued.

The base implementation formats the record to merge the message, arguments, and exception information, if present. It also removes unpickleable items from the record in-place. Specifically, it overwrites the record's `:attr:`msg`` and `:attr:`message`` attributes with the merged message (obtained by calling the handler's `:meth:`format`` method), and sets the `:attr:`args``, `:attr:`exc_info`` and `:attr:`exc_text`` attributes to `None`.

You might want to override this method if you want to convert the record to a dict or JSON string, or send a modified copy of the record while leaving the original intact.

**System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library]logging.handlers.rst, line 1048)

Unknown directive type "method".

```
.. method:: enqueue(record)
```

Enqueues the record on the queue using `put_nowait()`; you may want to override this if you want to use blocking behaviour, or a timeout, or a customized queue implementation.

## QueueListener

**System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library]logging.handlers.rst, line 1061)

Unknown directive type "versionadded".

```
.. versionadded:: 3.2
```

The `:class:`QueueListener`` class, located in the `:mod:`logging.handlers`` module, supports receiving logging messages from a queue, such as those implemented in the `:mod:`queue`` or `:mod:`multiprocessing`` modules. The messages are received from a queue in an internal thread and passed, on the same thread, to one or more handlers for processing. While `:class:`QueueListener`` is not itself a handler, it is documented here because it works hand-in-hand with `:class:`QueueHandler``.

**System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library]logging.handlers.rst, line 1063); [backlink](#)

Unknown interpreted text role "class".

**System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library]logging.handlers.rst, line 1063); [backlink](#)

Unknown interpreted text role "mod".

**System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library]logging.handlers.rst, line 1063); [backlink](#)

Unknown interpreted text role "mod".

**System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library]logging.handlers.rst, line 1063); [backlink](#)

Unknown interpreted text role "mod".

**System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library]logging.handlers.rst, line 1063); [backlink](#)

Unknown interpreted text role "class".

**System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library]logging.handlers.rst, line 1063); [backlink](#)

Unknown interpreted text role "class".

Along with the `:class:`QueueHandler`` class, `:class:`QueueListener`` can be used to let handlers do their work on a separate thread from the one which does the logging. This is important in web applications and also other service applications where threads servicing clients need to respond as quickly as possible, while any potentially slow operations (such as sending an email via `:class:`SMTPHandler``) are done on a separate thread.

**System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library]logging.handlers.rst, line 1071); [backlink](#)

Unknown interpreted text role "class".

**System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library] logging.handlers.rst, line 1071); [backlink](#)

Unknown interpreted text role "class".

**System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library] logging.handlers.rst, line 1071); [backlink](#)

Unknown interpreted text role "class".

Returns a new instance of the `class: 'QueueListener'` class. The instance is initialized with the queue to send messages to and a list of handlers which will handle entries placed on the queue. The queue can be any queue-like object; it's passed as-is to the `meth: 'dequeue'` method, which needs to know how to get messages from it. The queue is not *required* to have the task tracking API (though it's used if available), which means that you can use `class: '~queue.SimpleQueue'` instances for *queue*.

**System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library] logging.handlers.rst, line 1080); [backlink](#)

Unknown interpreted text role "class".

**System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library] logging.handlers.rst, line 1080); [backlink](#)

Unknown interpreted text role "meth".

**System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library] logging.handlers.rst, line 1080); [backlink](#)

Unknown interpreted text role "class".

If `respect_handler_level` is `True`, a handler's level is respected (compared with the level for the message) when deciding whether to pass messages to that handler; otherwise, the behaviour is as in previous Python versions - to always pass each message to each handler.

**System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library] logging.handlers.rst, line 1093)

Unknown directive type "versionchanged".

```
.. versionchanged:: 3.5
   The ``respect_handler_level`` argument was added.
```

**System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library] logging.handlers.rst, line 1096)

Unknown directive type "method".

```
.. method:: dequeue(block)

   Dequeues a record and return it, optionally blocking.

   The base implementation uses ``get()``. You may want to override this
   method if you want to use timeouts or work with custom queue
   implementations.
```

**System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library] logging.handlers.rst, line 1104)

Unknown directive type "method".

```
.. method:: prepare(record)

   Prepare a record for handling.

   This implementation just returns the passed-in record. You may want to
   override this method if you need to do any custom marshalling or
   manipulation of the record before passing it to the handlers.
```

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library] logging.handlers.rst, line 1112)**

Unknown directive type "method".

```
.. method:: handle(record)
```

Handle a record.

This just loops through the handlers offering them the record to handle. The actual object passed to the handlers is that which is returned from `:meth:`prepare``.

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library] logging.handlers.rst, line 1120)**

Unknown directive type "method".

```
.. method:: start()
```

Starts the listener.

This starts up a background thread to monitor the queue for LogRecords to process.

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library] logging.handlers.rst, line 1127)**

Unknown directive type "method".

```
.. method:: stop()
```

Stops the listener.

This asks the thread to terminate, and then waits for it to do so. Note that if you don't call this before your application exits, there may be some records still left on the queue, which won't be processed.

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library] logging.handlers.rst, line 1135)**

Unknown directive type "method".

```
.. method:: enqueue_sentinel()
```

Writes a sentinel to the queue to tell the listener to quit. This implementation uses ``put_nowait()``. You may want to override this method if you want to use timeouts or work with custom queue implementations.

```
.. versionadded:: 3.3
```

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library] logging.handlers.rst, line 1145)**

Unknown directive type "seealso".

```
.. seealso::
```

Module `:mod:`logging``  
API reference for the logging module.

Module `:mod:`logging.config``  
Configuration API for the logging module.