Configuration File Format

The configuration file format understood by fileConfig() is based on configparser functionality. The following table listed the “heading” of a logging configuration file:

Section Name	Required	Description
[loggers]	Yes	keys=root,*log02,log03,log04,log05,log06,log07*
[handlers]	Yes	keys=*hand01,hand02,hand03,consoleHandler, fileHandler*
[formatters]	Yes	keys=*form01,form02,simpleFormatter,complicatedFormatter*

In the above table, the blue colored text are the mandatory attributes. All the italic text are optional and customized per your use case. We call them as “**entity**”. Re. *log02*, *hand01*, *form01*, *consoleHandler* and so on. Based on the above “heading”, for each such **entity**, there is a separate section which identifies how that entity is configured. For example, for a logger named *log02* in the [loggers] section, the relevant configuration details are held in a section [logger_*log02*]. Similarly, a handler called hand01 in the [handlers] section will have its configuration held in a section called [handler_hand01], while a formatter called form01 in the [formatters] section will have its configuration specified in a section called [formatter_form01]. The root logger configuration must be specified in a section called [logger_root].

Here are the examples:

```
[logger_root]
level=DEBUG
handlers=hand02

[handler_hand02]
class=FileHandler
level=DEBUG
formatter=form02
args=('python.log', 'w')

[formatter_form02]
format=F1 %(asctime)s %(levelname)s %(message)s
datefmt=
```

In our code example, we specified TWO handlers: *consoleHandler* and *fileHandler.* We have also specified TWO formatters: *simpleFormatter* and *complicatedFormatter.* We will see how they are being used in the actual code.

LogRecord Attributes

Attribute name	Format	Description
asctime	%(asctime)s	Human-readable time when the LogRecord was created. By default this is of the form ‘2003-07-08 16:49:45,896’ (the numbers after the comma are millisecond portion of the time).
created	%(created)f	Time when the LogRecord was created (as returned by time.time()).

filename	%(filename)s	Filename portion of pathname.
funcName	%(funcName)s	Name of function containing the logging call.
levelname	%(levelname)s	Text logging level for the message ('DEBUG', 'INFO', 'WARNING', 'ERROR', 'CRITICAL').
levelno	%(levelno)s	Numeric logging level for the message (DEBUG, INFO, WARNING, ERROR, CRITICAL).
lineno	%(lineno)d	Source line number where the logging call was issued (if available).
module	%(module)s	Module (name portion of filename).
msecs	%(msecs)d	Millisecond portion of the time when the LogRecord was created.
message	%(message)s	The logged message, computed as msg % args. This is set when Formatter.format() is invoked.
name	%(name)s	Name of the logger used to log the call.
pathname	%(pathname)s	Full pathname of the source file where the logging call was issued (if available).
process	%(process)d	Process ID (if available).
processName	%(processName)s	Process name (if available).
relativeCreated	%(relativeCreated)d	Time in milliseconds when the LogRecord was created, relative to the time the logging module was loaded.
thread	%(thread)d	Thread ID (if available).
threadName	%(threadName)s	Thread name (if available).