# File System DEBUG logging

**THE INFORMATION IN THIS ARTICLE APPLIES TO:**

- EFT v7.2.7
- EFT v7.3.3 and later

**DISCUSSION**

EFT administrators may encounter issues related to a lack of timely responsiveness by the file system, but it can be difficult pinpoint that as the source of the issue. This is especially relevant when a network share is being leveraged for storage, but a stressed share host or network problems may also be inducing much higher latency than normal or otherwise causing abnormally slow responses. These kinds of behaviors can also be caused by directories that contain an extremely large number of files or folders, even if the EFT administrator is not aware of that being the case. By recording directory and file actions that take a long time to get a response and identifying precisely how long EFT has to wait for a response, administrators may be able to troubleshoot a problem more easily and accurately.

In EFT 7.3.3, a new provision was added to the existing debug logging capabilities. Leveraging this addition will write detailed file system activity information into a log file separate from the default EFT.log file configured in the existing logging.cfg file. By default, the output file will be called **EFT-FileSystem.log** and be located in the same local configuration directory as the logging.cfg and EFT.log files.

Example output, with emphasis on the last line's indication of a very long wait for a response in a folder containing 30,000 items:

[21524] DEBUG FileSystem <HTTP.ProcessRequest HTTP> - Operation: "GetFolderListing"; Folder: /Usr/a/; Mask: *; Client: a; Items in Listing: 7; Duration: 63 ms;

[928] DEBUG FileSystem <HTTP.ProcessRequest HTTP> - Operation: "GetFolderListing"; Folder: /Usr/a/newfolderRenamed/; Mask: *; Client: a; Items in Listing: 0; Duration: 31 ms;

[18780] DEBUG FileSystem <HTTP.ProcessRequest HTTP> - Operation: "GetFolderListing"; Folder: /Usr/a/LargeFolderListing/; Mask: *; Client: a; **Items in Listing: 30000; Duration: 1841 ms;**

This logging capability is further enhanced by the **FileSystemTimeoutMSec** registry entry documented in KB#11336 (https://kb.globalscape.com/KnowledgebaseArticle11336.aspx).

# File System DEBUG logging

The **FileSystemTimeoutMSec** registry entry defines the maximum time threshold in milliseconds that EFT should consider normal to wait for a complete response from the file system. With that value defined, any file system operation that takes longer will record a WARN-level entry to the log, which means administrators can set the logging.cfg entry to use a base of the higher level WARN rather than lower level and far more verbose DEBUG. This allows use of the file system logging capabilities without actually recording every single normal file operation, instead recording only those that may be cause for concern due to lengthy response times. This helps keep data sets and resulting log files much smaller and easier to parse for day-to-day use.

Examples of operations logged include:

- CheckPermissions
- FolderExists
- GetFolderListing
- GetListingEntry
- GetRealPath
- GetFileSize
- LockFile
- OpenFile
- UnlockFile

**INSTRUCTIONS**

The file system logging capabilities are easily enabled by appending new lines to the end of the existing logging.cfg file found in EFT's local configuration directory. Open the file in a text editor, go to the end of the file, and paste the following lines:

```
# File System Logging - Quickly log File System activities to a separate file for debug diagnostics.

#

#log4cplus.appender.FileSys=log4cplus::RollingFileAppender

#log4cplus.appender.FileSys.File=${AppDataPath}\EFT-FileSystem.log

#log4cplus.appender.FileSys.MaxFileSize=20MB
```

# File System DEBUG logging

#log4cplus.appender.FileSys.MaxBackupIndex=5

#log4cplus.appender.FileSys.layout=log4cplus::TTCCLayout

#log4cplus.logger.FileSystem=DEBUG, FileSys

#log4cplus.additivity.FileSystem=false

NOTE 1: Because of Microsoft's UAC, you may be required to run the text editor as an administrator to save your changes.

NOTE 2: Setting the log level to DEBUG will capture *all* activity and time taken. Do not set the value to TRACE, as this will create a large amount of data that isn't generally useful. The generally useful information has been purposefully contained in the DEBUG log level to keep the log file more effective and concise. Or use the aforementioned FileSystemTimeoutMSec registry entry and set log level to WARN to limit log entries to only those with long enough response times to cause concern.

To enable the file system logging, remove the comment character (the pound sign or hash mark #) from the beginning of all the lines above that otherwise start with *log4cplus*, and save your changes. To see the results, after some activity has occurred, open the automatically created EFT-FileSystem.log file in a text editor.

When sifting through large amounts of logging data, some users may find it convenient to copy the log file to a workstation to open the file in a spreadsheet editor such as Microsoft Excel.

> To make the data available for sorting and filtering, use the "Text to Columns" feature with the "Delimited" option enabled and the semicolon set as the delimiter.
>
> Advanced users may wish to bulk-add delimiters before and after the number of milliseconds to separate the number itself from the text "Duration" and "ms" for sorting or filtering (or simply bulk delete all instances of "Duration" and "ms" using an advanced text editor).

# File System DEBUG logging

GlobalSCAPE Knowledge Base
https://kb.globalscape.com/Knowledgebase/11344/File-System-DEBUG-logging