WHEN YOU NEED TO BE SURE

CyFlex® Fast Data Logger Application User Guide

Version 11

November 6, 2023

Developed by SGS North America, Inc.

Version History

Version	Date	Revision Description
1	1/25/2016	Initial publication
2	8/23/2018	Format to SGS brand
3	3/31/2020	Retrofit to new template
4	4/1/2020	Revised Section 4 Specification File Format to describe added function for the `@SCAN_INTERVAL` keyword on page 6 as of CyFlex version 6.3.X.
5	6/2/2020	Add *Section 5 logr_spec.def Example* on page 18
6	9/29/2020	Add `HI_RES` to definition of `@TIME_FORMAT` on page 13.
7	9/7/2021	Removed usage for `floger` in *Section 2 Starting the Application* on page 3 and added hypertext link to its usage help on cyflex.com
8	4/18/2022	Revised hypertext link to `floger` usage help on cyflex.com in *Section 2 Starting the Application* on page 3 to align with its updated category.
9	4/25/2022	Replaced *Section 5* `dlogger_specs.def` example on page 18 with `logr_spec.def` example.
10	5/24/2022	Updated hypertext link to `floger` usage help on cyflex.com in *Section 2 Starting the Application* on page 3
11	11/6/2023	Updated *Section 5 logr_spec.def Example* on page 18 to include implementation of CSAR labels.

Document Conventions

This document uses the following typographic and syntax conventions.

- Commands, command options, file names or any user-entered input appear in Courier type. Variables appear in Courier italic type.

  Example: Select the `cmdapp-`*`relVersion-buildVersion`*`.zip` file….

- User interface elements, such as field names, button names, menus, menu commands, and items in clickable dropdown lists, appear in Arial bold type.

  Example: **Type**: Click **Select Type** to display drop-down menu options.

- Cross-references are designated in Arial italics.

  Example: Refer to *Figure 1*…

- Click intra-document cross-references and page references to display the stated destination.

  Example: Refer to *Section 1* Overview on page 1.

  The clickable cross-references in the preceding example are *1*, *Overview*, and on page 1.

CyFlex Documentation

CyFlex documentation is available at https://cyflex.com/. View **Help & Docs** topics or use the **Search** facility to find topics of interest.

Table of Contents

1 Overview

The `floger` task, formerly known as `loger`, is used to collect engineering units data and write it into a text (ASCII) file in a columnar format that is easily imported into spreadsheets. The program is capable of logging up to 1000 channels of data at rates up to 500 samples per second. The sampling rate can be synchronized by an external event and can be started and stopped by events. Data logging can also be enabled or disabled by a logical variable.

The data logger supports dynamic flexible output file naming. The name of the output file may be constructed from variables and fixed strings. On each scan of the specified variables, the name of the file is constructed and compared to the filename from the previous scan. If the filename has changed, then the old file is closed and a new one opened. Thus, a scanning process may continue indefinitely, but can result in new files being created based on the change of time or some other variable such as a test cycle counter.

The program can log any real, integer, logical, or string variable. It can also log any member of a statistical, composition, property, or emission variable.

Any events used by `floger`, e.g. to synchronize, start, or stop sampling, will be created if they do not already exist. These events might be set by the Test Scheduler, or they could be set by the user through hotkeys or commands.

All of the variables to be logged must already exist. This application will not create them.

1.1 Additional Application Information

Up to 1000 variables can be sampled.

Available logging rates range from 500 samples/second (per channel) to one sample per minute. Note that logging a variable does not imply that a new value has been sampled. Sampling rates are controlled by other processes.

By default, the data file will appear in the `/data/PC_format/` directory. If desired, specify a different directory by using the command line option or by specifying the keyword `@OUTPUT_PATH` in the specification file. Any valid path on the network may be specified.

Example:

```
floger /specs/logr_spec outfile=/dos/a/logdos.dat &
```

Sampling start depends on the following conditions:

- IF the @START_EVENT keyword is specified, logging starts when the `start_event` is received.
- IF the @START_EVENT keyword is not specified, it starts immediately when the task is started UNLESS an enable variable is specified via the @ENABLE keyword and its state is FALSE.

Two methods are available to take data only during a certain part of a test:

- Use pairs of @START_EVENT/@STOP_EVENT to turn logging on and off.
- Set the variable specified with the @ENABLE keyword to TRUE and FALSE.

To log data when reaching a critical point in time, use the event specified by the keyword @SYNC_EVENT to control data collection between periodic samples. The `@sync_event` can be named in a test procedure with the @SET_EVENTS keyword (`gp_test` keyword) or it could be an already existing event like `fr_done` which will cause a scan every time a fuel reading is complete.

1.2 Computed Expressions

Computed expressions must be enclosed with double quotes ("…"). A literal string must be enclosed with a single quote ('…'). Strings may be joined by using the plus (+) sign.

For example, assume the following computed expression was entered for the test description keyword:

```
@DESCRIPTION
" 'Engine model = ' + model + ' S/N = ' + serial"
```

Given that the CyFlex string variable <*model*> had a value of <Enforcer 02> and the string variable <*serial*> had a value of <14014957>, the test description for the logger file would be equal to:

```
Engine model = Enforcer 02 S/N = 14014957
```

2 Starting the Application

Specify `floger` to start the process from the command line or from a script file. Multiple copies of the process may run simultaneously with different specifications. Specifications for the data logging are read from a file, the name of which is a command line argument. There is no default filename.

Options may also be specified via the command line. Specified options override the options that are read from the specification file. Refer to cyflex.com usage help for `floger`.

3 Terminating the Application

Specify the `release_event` to terminate the `floger` task. If each copy of `floger` that is running is specified with a different `release_event`, they may be released selectively. They may also be released at the same time by specifying the same `release_event`.

If a `floger` task has no `start_event` specified, it will automatically be terminated when the maximum number of scans is reached. Refer to *Section 1.1 Additional Application Information* on page 1 for related information.

4 Specification File Format

The specification file is made up of keywords, which are identified by a leading '@', followed by the value that should be assigned to that keyword. On the line following the keyword is the value that should be assigned to that keyword. In the general case this would look like:

```
@keyword
    keyword_value
```

There are several keywords that must be specified before `floger` can begin its execution. All other keywords do not need to specify if their functionality is not desired.

The following table lists all the keywords, whether or not they are required, if the keyword supports computed expressions, and a description.

Keyword	Required	Computed Expression	Description

`@FILENAME`	Yes	Yes	This entry contains the name of the output file. It may be a fixed name or it can be 'constructed' from a computed expression.
			Examples:
			A fixed name is specified as a literal string with the filename enclosed in single quotes:
			`@FILENAME`
			`'my_file'`
			A computed expression may contain something like `test_mode = 3` and `test_cycles = 1230`. Then the filename of `log3.123.dat` could be constructed. Refer to Section 1.2 Computed Expressions on page 2.
			`@FILENAME`
			`"('log' + test_mode + test_cycles/10[none] + '.dat')"`

`@REG_NAME`	No	No	The name that should be used to register this instance of `floger` with the OS. The name must be unique thought out the system or the task will fail to Initialize correctly. Example: `CVS_FTP75`

Keyword	Required	Computed Expression	Description
@SCAN_INTERVAL	Yes	Yes	The time between lines of data in the output file. Example: @SCAN_INTERVAL 0.20[sec] **Note:** Scan intervals that are less than one second will cause data to be saved in memory and written to the output file when a STOP_EVENT is received or the @MAX_SCANS value has been reached. This feature can be overridden with the @FORCE_DIRECT_FILE_WRITE command; see page 13. This value also affects the default format of the time being written to output file. If the scan interval is less than 1.0 second then the time, in seconds, relative to the beginning of sampling will be written to the file. If the scan interval is 1.0 second or greater, then the time will be written in the DEFAULT_TIME format. Refer to @TIME_FORMAT on page 13. If the SCAN_INTERVAL is entered as a computed expression, the expression will be evaluated each time that the START_EVENT is set. **Note:** Starting with CyFlex version 6.3.X, the interval can also be entered as any of the standard CyFlex intervals (FAS, MED, etc.). This includes all phased timer intervals.
@GET_NEW_SCAN_INTERVAL	No	No	The name of an event that can be used to trigger a reevaluation of the SCAN_INTERVAL computed expression. If the event does not exist at the time that the floger task is started, it will be created.

Keyword	Required	Computed Expression	Description

@DESCRIPTION	No	Yes	The user-configurable description will appear at the top of the output file.
			Example of a simple string description should be enclosed in single quotes:
			`@DESCRIPTION`
			`'This is a description of my test'`
			A more complex description can be constructed using a computed expression.
			`@DESCRIPTION`
			`'Torque sweep, model=' + model + ', S/N= ' + serial`
			If the CyFlex variables `model` and `serial` had values of `Sig 600` and `14026490` respectively, the following would be written to the output file:
			`Torque sweep, model=Sig 600, S/N= 1402690`
			The keyword `@WRITE_HEADER` controls when the description is written to the output file.

@PACKED	No	No	This flag specifies that the output should be compressed with a comma delimiter between fields. The default is a 14-character column, space-delimited, format which is easier to read but which takes more disk space. Example: `@PACKED`
@SINGLE_ PRECISION	No	No	For sample rates faster than 1Hz, the data is stored in RAM. The default storage size a double precision floating point value. By using the `@SINGLE_PRECISION` keyword, the storage requirement will be reduced by 50%. The only known parameter that is adversely affected by using this keyword is `time`, which is a very large number. Example: `@SINGLE_PRECISION`

Keyword	Required	Computed Expression	Description
@LOG_DIGITAL_DESCP	No	No	This keyword will cause the LOGICAL_VARIABLE descriptions to be logged for all LOGICAL_VARIABLES in place of the values 0 or 1. The keyword may have an entry following it of either yes or no. If no entry follows this keyword the value of yes is assumed. Example: @LOG_DIGITAL_DESCP Yes
SCAN_LIST	Yes	No	The list of variables that are to be sampled. Each variable specified may have units specified, a C-language format statement for logging the value to the output file, and/or a statistical member. Statistical members are only valid if the @LOG_STATISTICS or @RUNNING_AVERAGE keywords have been specified before this keyword is processed. The statistics are computed internally and are not the values of any CyFlex Statistical variable. Example 1: @SCAN_LIST int_mnf_t %5.2f int_mnf_p[in_hg] The above will log int_mnf_t with a minimum field width of 5 and have 2 digits to right of the decimal point It will also log int_mnf_p in units of 'inches of mercury'. Note: When specifying a format, a space should be between the variable name and the format statement. However, when specifying the units, there should NOT be a space between the variable name and the specified units of [in_hg]. Example 2: @SCAN_LIST int_mnf_t int_mnf_t .MX If the @LOG_STATISTICS or @RUNNING_AVERAGE keyword have been specified, then the values logged are different than described above. If @LOG_STATISTICS keyword has been specified and the scan list

Keyword	Required	Computed Expression	Description

			is that shown in example 2, the log file would contain the average value of the parameter `int_mnf_t` as the first value and it would have the maximum value of the parameter `int_mnf_t` as the second value. The maximum value member was specified by entering the standard two-character CyFlex statistical variable member preceded by a period. Refer to `@RUNNING_AVERAGE` on page 15 for more details.
			Example 3:if the standard deviation for the variable in example 2 should be logged, the entry under the `@SCAN_LIST` would be:
			`int_mnf_t .SD`
			If the specified variable that is being logged is a `LOGICAL_VARIABLE` the logical description may be logged in place of the values `0` or `1`. This capability is available in versions 08Sep and later. For example, if the following channel specification is an entry under the `@SCAN_LIST` keyword,
			`enab_lwr_lmt LOG_DIGITAL_DESCP`
			the `LOGICAL_VARIABLE` description of `enab_lwr_lmt` will be logged in place of the values `0` or `1`. The entry must be exactly `LOG_DIGITAL_DESCP` or the logical description will not be logged.
			Format hints:
			• For floating point numbers (CyFlex `REAL_VARIABLES`), use `%W.Xf`:
			o `W` is the field width
			o `X` is the number of places to the right of the decimal point
			• For integers (ASSET `INTEGER_VARAIBLES`), use `%Wd`:
			o `W` is the field width
			• For strings (ASSET `STRING_VARAIBLES`), use %.s

Keyword	Required	Computed Expression	Description
@SUPPRESS_AV	No	No	Use this keyword in conjunction with the @LOG_STATISTICS and @RUNNING_AVERAGE keywords to determine if the label in the output file should be the value in the SCAN_LIST with .AV appended to it. To suppress the .AV extension use: @SUPPRESS_AV Yes or @SUPPRESS_AV The keyword need not be used in order for .AV to be appended to the label in the SCAN_LIST.
@START_EVENT	No	No	The name of an event that signals the start of a sampling interval. Example: @START_EVENT start_logging
@STOP_EVENT	No	No	The name of an event that signals the end of a sampling interval. Example: @START_EVENT start_logging
@RELEASE_EVENT	No	No	The name of the event that signals the end of a sampling interval and that the floger task should exit after the data file has been written. Example: @RELEASE_IT rels_it
@SYNC_EVENT	No	No	The name of an event that can be used to trigger a scan of all channels, usually as an alternative to sampling at a periodic interval. If both @SCAN_INTERVAL and @SYNC_EVENT are specified, the sync scans and interval scans will be interlaced. Example: @SYNC_EVENT log_now

Keyword	Required	Computed Expression	Description

@MAX_SCANS	No	Yes	The maximum number of samples in a sampling session. A zero value or the keyword being absent indicates a sampling session will continue until a stop_event is received.
			Example:
			`@MAX_SCANS`
			`1000`

@DONE_EVENT	No	No	The name of the event that is set when the data collection is complete. This is an output event and can be used to inform another process that the output file is now available.
			Example:
			`@DONE_EVENT`
			`logging_done`

@READ_SPEC_FILE_EVENT	No	No	The name of an event that, when it is received by `floger`, will cause `floger` to re-read the spec file.
			Example:
			`@READ_SPEC_FILE_EVENT`
			`read_it`

@FEDERAL_SMOKE	No	No	Generate a header file that is read by the federal smoke report program. The filename is optional and if missing, the spec file being processed will be used to generate the header file. Example: `/specs/logr_hdr.fsr`

@ENABLE	No	Yes	The enable variable is a logical variable that must be set to TRUE before logging will take place. Typically, this variable would be set in a `gp_test` procedure or manually by the user. It may be used to turn logging on and off at different modes of a test.
			Example:
			`@ENABLE`
			`logging_ok`

@LOGGING_ACTIVE LABEL	No	No	The name of an ASSET LOGICAL_VARIABLE that indicates the `floger` is actively collecting data and logging it.
			Example:
			`@LOGGING_ACTIVE_LABEL`
			`Loger_collecting`

Keyword	Required	Computed Expression	Description
@OUTPUT_PATH	No	No	The directory path specifying where the output file should be written. If this keyword is absent, then the default path of /data/PC_format will be used. Example: @OUTPUT_PATH /data/PC_format/logger/
@OVERWRITE_ OUTPUT	No	No	Indicates that an existing file should be overwritten. If this keyword is absent then the default action is to append the new data to an existing file. Example: @OVERWRITE_OUTPUT
@WRITE_HEADER	No	No	Indicates when the header (labels and units lines) should be written to the output file. Valid entries are: • Always to always write the header when the start event is received • None_on_append to write the header only when the output file is new so that the header will be the first entry in the file • Never to never write header lines • Enable_transition_true to write the header when the enable variable (as specified via the @ENABLE keyword) goes from FALSE to TRUE Entries must be on the same line and they are case insensitive. Example: @WRITE_HEADER None_on_append Enable_transition_true The above would cause floger to write the header if the output file didn't previously exist and whenever the enable variable becomes TRUE.
@WRITE_EXTERNAL_ HEADER	No	No	The name of the file that will be copied to the output file when the data is written to the output file. Example: @WRITE_EXTERNAL_HEADER tmp/my_header.info

Keyword	Required	Computed Expression	Description
@FAST_TIMESTAMP	No	No	Specifies that time should be written to the output file relative to when the time sampling began (fast timestamp). If not specified, the time format is a function of the sample interval Refer to @SCAN_INTERVAL on page 6. Example: @FAST_TMESTAMP
@TIME_FORMAT	No	No	Specifies the format of the date/time that is logged in the file. Valid entries are: • DEFAULT_FORMAT to format the time/date as MM/DD/YYY hh:mm:ss. For example: 11/23/06 12:03:04 Note: If no entry follows the keyword, then DEFAULT_FORMAT option is used. • OLD_FORMAT to format the date/time as MON YY hh:mm:ss. For example: Aug 21 12:12:34 • EURO to format the date/time as YYYYMMDD hh:mm:ss. For example: 20060813 13:14:15 • EURO2 to format the date/time as DD/MON/YYYY hh:mm:ss. For example: 13/Aug/2006 08:09:10 • HI_RES to format the date/time as DD/MM/YEAR HR:MIN:SEC.MSEC. For example: 22/08/2020 08:09:10.123 Note: If this keyword is not specified, then the time format used will be a function of the @SCAN_INTERVAL.
@FORCE_DIRECT_ FILE_WRITE	No	No	Indicates to write data directly to the output file when high data rates are used. Exercise care when using this command with very high data rates so that excessive CPU time is not used by the floger program. Example: @FORCE_DIRECT_FILE_WRITE NOTE: This option must be used in order to log string variables at the high data rates.

Keyword	Required	Computed Expression	Description
@DOS_NFS_DRIVE	No	No	Specifies that output should be written on the NFS DOS mounted (World) drive. The location of the World drive is contained in the file /data/cell_map_location on each CyFlex computer. Example drive: //2/imports/world/. Consult your system administrator to see if there is a World drive mounted at your location. Example: @DOS_NFS_DRIVE Default
@MAX_STATISTICAL_SCANS	No	No	Specifies the maximum number of scans when the @LOG_STATISTICS keyword has been specified. Example: @MAX_STATISTICAL_SCANS 1000 This causes statistics to be calculated when 1000 scans have been made.
@CLEAR_STATISTICS_EVENT	No	No	The event that will cause the statistical buffers within floger to be reset to 0. Example: @CLEAR_STATISTICS_EVENT clear_stats This might be used at the start of a test mode so that statistics only apply to data taken in that mode.
@LOG_STATISTICS	No	No	Specifies that statistics should be computed for the variables specified via the @SCAN_LIST keyword. The statistical variables are created locally and are not available to any other process. Data collection begins when the start event (@START_EVENT keyword) and is collected at the rate specified by the @INTERVAL keyword. The statistical values are logged to the output file when a stop event (@STOP_EVENT) is received or the maximum number of samples (@MAX_STATISTICAL_SCANS keyword) have been collected. This also stops the data collection.

Keyword	Required	Computed Expression	Description
			By default, the average value, AV member of the statistical variable, is logged; however, additional members may be logged. Refer to the @SCAN_LIST keyword on page 8. When another start event is received, all statistical buffers are reset and the data collection process begins again. NOTE: If another start event is received before a stop event has been received or the maximum number of scans is reached, then no output will be produced. The variables will be reset and the data collection will begin again. Example: `@LOG_STATISTICS`

@RUNNING_AVERAGE	No	Yes	Specify the window width of a running average and the event that will cause the data to be logged.
			This keyword causes statistics to be computed for the variables specified via the @SCAN_LIST keyword. The variables are created locally and are not available to any other process.
			The statistics are computed for the specified window width and continue to be computed as long as data collection is active. The total number of data points making up the running average is a function of the window width and the scan interval as specified by the @SCAN_INTERVAL keyword.
			Computed expressions are allowed for the window width specification.
			The values will be logged to the output file whenever the specified 'log data event' is received. By default, the average (AV) member is logged; however, additional members may be logged. . Refer to the @SCAN_LIST keyword on page 8.
			Example:
			`@RUNNING_AVERAGE`
			`# window width log data event`
			`30[sec]`
			`log_average_data`

Keyword	Required	Computed Expression	Description
@FTP_PATH	No	No	Specify the directory on the remote destination that will receive the data file. This is pre-pended to the @FTP_DIRECTORY entry. Default is /group/mech_dev/. Example: @FTP_PATH /group/perf_dev/
@FTP_DIRECTORY	No	No	Specify the directory to be appended to the @FTP_PATH entry. NOTE. This entry is required to activate FTP transfers. There is no default. Example: @FTP_DIRECTORY isx2/
@FTP_EVENT	No	No	Specify the ASSET event that will be set to cause the logged data file to be transferred to the remote destination. Default is FTP_write. Example: @FTP_EVENT ftp_log_data
@FTP_HOST	No	No	Specify the name of the remote destination (IP address or hostname) that should receive a copy of the logged data file. Default is cidcssetdl1. Example: @FTP_HOST ctcqnx2
@FTP_ACCOUNT	No	No	Specify the account name to be logged into on the remote destination. Default is datasend. Example: @FTP_ACCOUNT cell411 Note, the default account is a public account and data in it may be viewed by anyone inside the firewall. If you have data that you want to protect, you should specify an account with limited access.

Keyword	Required	Computed Expression	Description
@FTP_PASSWORD	No	No	Specify the password for account name to be logged into on the remote destination. If not specified, the default value (data123) will be used. Example: @FTP_PASSWORD Mypasswd

5 logr_spec.def Example

```
# ----------------------------------------------------------------
#
#       The floger program now requires keywords in the specification
#       file.  If floger encounters a file of the old format, it
#       automatically converts the file and saves the old file in
#       /data/errors
#
#        NOTE:
#           !! Keywords that are marked with !! will accept
#              any valid computed expression
#              See COMPUTED EXPRESSION FORMAT at end of document
#
#           ** Keywords that are marked with ** are required
#              All other keywords are optional
#
#
# ----------------------------------------------------------------
#
#  @DESCRIPTION   !! **
#     My_Test            <A title to be written to the output file>
#
#  @FILENAME      !! **
#     file_name          <the contents of the CyFlex variable
#                         will be used as the name of file that
#                         will contain the logged data.>
#
#  @START_EVENT
#     start_it           <the name of the event that will start
#                         the data acquisition. The event will be
#                         created if it doesn't exist. If not>
#                         specified the sampling will start
#                         immediately
#
#  @STOP_EVENT
#     stop_it            <the name of the event that will stop
#                         the data acquisition. The event will be
#                         created if it doesn't exist>
#
#  @RELEASE_EVENT
#     all_done           <the name of the event that will cause
#                         the loger task to exit. The event will be
#                         created if it doesn't exist>
#
#  @READ_SPEC_FILE_EVENT
#     readit             <the name of the event that will cause
#                         the loger task to re-read the spec file.
#                         The event will be created if it doesn't exist>
#
#  @DONE_EVENT
#     all_done           <the name of the event that will be set
#                         when the data acquisition is complete.
#                         The event will be created if it doesn't
#                         exist>
```

```
#
#  @SCAN_INTERVAL !! **
#     10[sec]            <the time between data samples.  Beginning with
#                         cyflex.6.3.exp this value can be a standard
#                         CyFlex interval (FAS, MED, SLO, etc.)>
#
#  @ENABLE        !!
#     loggin_ok          <the CyFlex logical variable that is used
#                         to enable/disable logging data.>
#
#  @LOGGING_ACTIVE_LABEL
#     loggin_active      <the CyFlex logical variable that will be set
#                         TRUE when data is being logged.>
#
#  @SYNC_EVENT
#    log_it_now          <the name of the event that will cause
#                         data to be logged. The event will be
#                         created if it doesn't exist>
#
#  @MAX_SCANS     !!
#     1000               <The maximum number of times to sample
#                         the channels and write their values to
#                         the output file.>
#
#  @SCAN_LIST         **
#    label_1             < the CyFlex variable to be logged, can also be
#                          a CSAR variable label. If the CSAR label
#                          is duplicated with multiple ECMs then the ECM
#                          prefix and an '_' should be added.
#                          ex. ECM2_ccsar_variable_name>
#    label_2  %10.4f     < An optional format may be specified>
#    label_2[units]      < Optional units may be specified for
#                          each label>
#    label_2 .MX         < Statistical member may be specified>
#                          when either @LOG_STATISTICS or
#                          @RUNNING_AVERAGE keywords are specified>
#
#    label_3 LOG_DIGITAL_DESCP < 'label_3 is a LOGICAL_VARIABLE and
#                                description should be logged
#                                instead of 0 or 1.
#
#                        < The optional format, statistical member
#                          and LOG_DIGITAL_DESCP may be specified
#                          in any order.  If units are specified
#                          they must immediately follow the label
#                          name and be enclosed in []>
#
#
#  @FORCE_DIRECT_FILE_WRITE
#    yes                 < the data will be written directly to the
#                          output file even though the sample rate
#                          implies that it should be written to
#                          memory.
#
```

```
#  @LOG_DIGITAL_DESCP
#    yes                < When logging a LOGICAL_VARIABLE log the
#                         TRUE or FALSE description instead of 1 or
#                         0. The entry 'yes' is optional. If no entry
#                         is found, 'yes' is assumed.
#
#  @REG_NAME
#    mylogr             < the name that should be used to register
#                         this instance of floger with the operating
#                         system. If this entry is not found the
#                         name of the spec file being read will be
#                         used as the registered name.
#  @OUTPUT_PATH
#     /data/mydata      < the directory path specifies where to
#                         write the output file>
#
#  @PACKED
#    Yes                < Should the output file be a packed comma
#                         separated format? The entry
#                         'yes' is optional. If only the keyword is
#                         entered, then the output is packed. Entry
#                         is case insensitive>
#
#  @SINGLE_PRECISON     < For sample rates faster than 1Hz, the data
#                         is stored in RAM.  The default storage method
#                         is double precision floating point.  By using
#                         the @SINGLE_PRECISION keyword, the storage
#                         requirement will be reduced by 50%.  The only
#                         parameter which is normally affected adversely
#                         by this is a 'time' parameter, which is a
#                         very large number.
#
#  @OVERWRITE_OUTPUT
#    Yes                < Should an existing output file be over
#                         written on startup? The entry
#                         'yes' is optional. If only the keyword is
#                         entered, then the output is overwritten.
#                         Entry is case insensitive>
#
#  @WRITE_EXTERNAL_HEADER
#    filename           < the name of a file that will be copied to the
#                         output file when the output file is written to.
#
#  @FAST_TIMESTAMP
#    Yes                < Should the time stamp in the output file
#                         be a relative to start time? The entry
#                         'yes' is optional. If only the keyword is
#                         entered, then the fast timestamp is
#                         activated. Entry is case insensitive>
#
#  @TIME_FORMAT
#                        NOTE: versions of floger that are new than
#                        January 31, 2009 now support the @TIME_FORMAT
#                        when sampling at rates faster than 1 Hz.
#                           The default time format (format not specified)
```

```
#                           changes, depending on the sample rate:
#
#                             1Hz or slower     - DEFAULT_FORMAT
#
#                             faster than 1Hz   - RELATIVE
#
#
#    DEFAULT_FORMAT     < the format for the date/time will be
#                         MM/DD/YYYY hh:mm:ss. For example,
#                         11/11/2004 11:10:11
#
#                         NOTE: If no entry follows the keyword the format
#                         will be DEFAULT_FORMAT.
#
#    OLD_FORMAT         < the format of the time in the output file
#                         should be "MMM DD hh:mm:ss". For example,
#                         Jan 04 11:01:03
#
#    EURO               < the format of the time in the output file
#                         should be "YYYYMMDD hhmmss". For example,
#                         20050123 110103
#
#    EURO2              < the format of the time in the output file
#                         should be "DD/MON/YYYY hh:mm:ss". For example,
#                         03/Jan/2005 11:01:03
#
#    RELATIVE           < the format of the time in the output file
#                         will be a floating point number exactly the
#                         same as when the @FAST_TIMESTAMP option is
#                         used or in older versions of 'floger' when
#                         the sampling rate was faster than 1Hz
#
#    HI_RES             < the format for the date/time will be
#                         MM/DD/YYYY hh:mm:ss.XXX, where XXX is milliseconds
#                         For example,
#                         11/11/2004 11:10:11.161
#
#  @WRITE_HEADER
#     Always Never None_on_Append Enable_Transition_True
#
#                       < Always - will write headers on startup
#                             This is the default
#
#                         Never  - Don't write headers on startup
#
#                         None_on_Append - Don't write headers when
#                             appending to file
#
#                         Enable_Transition_True - Write headers
#                             when the enable variable goes true
#
#                         All entries must be on the same line.
#                         They can be in any order and are case
#                         insensitive>
#
```

```
#  @FEDERAL_SMOKE
#    filename           < generate a header that is read by the federal
#                         smoke report program. 'filename' is optional and
#                         if not present the spec file being processed
#                         will be read to generate the header.
#
#  @DOS_NFS_DRIVE
#      /nfs/dir/  or
#       Default          < Specifies the NFS DOS mounted drive or
#                         if the word 'Default' is found the
#                         default NFS directory will be used
#                         At CTC this is "//2/imports/world". The
#                         word 'Default' is case insensitive>
#
#  @LOG_STATISTICS
#         Yes             < Should the statistical averages be
#                         logged?  The entry 'yes' is optional. If
#                         only the keyword is entered, then
#                         statistical logging is activated. Entry
#                         is case insensitive.>
#
#  @CLEAR_STATISTICS_EVENT
#     clear_stat         <the name of the event that will be set
#                         to cause the statistical variables to be
#                         initialized to 0. The event will be
#                         created if it doesn't exist.>
#
#  @MAX_STATISTICAL SCANS     !!
#     max_stat_scans     <The maximum number of times to sample
#                         the channels. When the channels have been
#                         sampled this number of times, the
#                         satistics are computed and written to
#                         the file.>
#
#  @RUNNING_AVERAGE
#     window_width[time_units] !!    log_event_name
#                        <window_width = the width of the running
#                         average window in units of time. The number
#                         of samples making up the window is a
#                         function of the specified interval.
#
#                         log_event_name = the name of the event
#                         that will cause the averages to be
#                         written to the output file. The .AV
#                         member is written by default.>
#  @FTP_HOST
#     hostname           <the name of the remote destination that
#                         receives a copy of the logged data
#                         file. Default value = ctcxssetdl1.>
#
#  @FTP_ACCOUNT
#     account_name       <the account name to be logged into on
#                         hostname. Default value = datasend.>
#
#  @FTP_PASSWORD
```

```
#     password          <the password for account 'account_name'
#                        Default value = whatever is the default
#                        value for account datasend.>
#
#  @FTP_DIRECTORY
#     remote_dir        <the directory path be appended to the
#                        @FTP_PATH entry. NOTE: This entry is
#                        required to activate FTP transfers.
#                        There is no default>
#
#  @FTP_PATH
#     root_dir          <the directory on the remote destination
#                        that will receive the data file.
#                        Default value = /group/mech_dev/. This
#                        is prepended to the @FTP_DIRECTORY entry.>
#
#  @FTP_EVENT
#     ftp_it            <the name of the event that will be set
#                        to cause the logged data file to be
#                        ftp'd to '@FTP_HOST'. The event will be
#                        created if it doesn't exist.
#                        Default value = FTP_write.>
#
#  @GET_NEW_SCAN_INTERVAL
#     my_event          <the name of an event that will cause floger
#                        to re-evaluate the expression that was provided
#                        as an interval in the original specification.
#                        Use of this feature for intervals that were
#                        not entered as an expression may have unwanted
#                        results.
#
#  @CSAR_LABEL
#     csar              <Use the CSAR variable label in the the header
#                        of the data output file.>
#
#     prefix            <Use the CSAR variable lebel including the ECM
#                        prefix on the label in the header of the data
#                        output file.>
#
#     ext               <Use the CSAR external label in the header
#                        of the data output file.>
# ------------------------------------------------------------------
#
#  COMPUTED EXPRESSION FORMAT
#
#     Computed expressions must be enclosed with double
#     quotes ("). A literal string must be enclosed with a
#     single quote ('). Strings maybe concatenated by using
#     the plus (+) sign.
#
#  For example:
#
#      Assume the following computed expression was entered for
#      the test description
#
```

```
#     "'Engine model = ' + model + ' S/N = ' + serial"
#
#      Given that the CyFlex string variable <model> had a value
#      of <Enforcer> and the string variable <serial> had
#      a value <14014957> the test description for the loger file
#      would be equal to
#
#      Engine model = Enforcer S/N = 14014957
#
#
#
# ---------------------------------------------------------------
#End_spec_file_description

@DESCRIPTION
'example logger description'

@FILENAME
" 'log_' + @year_month_dayL() + '.' + @hour_of_dayL() + '.csv'"

@SCAN_INTERVAL
100[msec]

@START_EVENT
e_strt_example_log

@STOP_EVENT
e_stop_example_log

@RELEASE_EVENT
e_rels_example_log

@LOGGING_ACTIVE_LABEL
example_logr_sts

@PACKED
YES

@OUTPUT_PATH
/data/PC_format/my_new_data

@SCAN_LIST
     # ASSET variable label[optional units]  [optional format or statistical
member]
air_mtr0_p
air_mtr0_p_cmp
oilrfl_p[kpa]

ctl_torq %10.4f


$
```