SSSG dsLogCSV User Guide

Tim Thiel/Laura Stolp

2015-03-09 - Rev 007

Corresponds to SVN Version 97 (2015-02-08)

Scope of document

The dsLogCSV software is used to repackage the output of a dsLog process and create a condensed data record at a user specified cadence. Typically, it will be used to generate a line of output each minute. The operation of the dsLogCSV software is controlled by specifying a user-generated initialization file. This document will provide the information needed to create an appropriate initialization file and run the dsLogCSV application.

Overview of dsLogCSV

The dsLogCSV application will launch separate threads to handle input data and output data. The input threads will wait for data on the appropriate channel and then attempt to parse out the user-defined data fields when that data is received. The output threads will wait for a specified number of seconds to elapse and will then create an output data string according to the configuration specified by the user. If no data has been received on a particular channel since the last time the output thread created an output then a string indicating there is no new data (the default string is “NAN”) will be placed in the output data string.

Each input corresponds to a specific UDP channel. There could be several different items of data that could be parsed out of the messages received on that UDP channel. At this point it is only possible to parse out predefined elements of certain predefined message types. For example mechanisms have been created to take a NMEA $GPRMC string and parse out latitude, longitude, SOG, COG, and other elements.

An array of data bins exist to store the data parsed by the input threads for use by the output threads in creating the output strings written out to the files or UDP sockets. Each bin contains the latest value of the parsed data and a timestamp indicating when the data was received. This array is referred to as the “ABSTRACT_DATA” in the code and the “ABSTRACT_INDEX” will provide the mapping of these items between the input and output threads. A table should be created to keep track of these mappings. The recommended practice is to keep the table in the header comments of the configuration file itself. An example of such a table:

ABSTRACT_INDEX	Value	String	RULE
1	Gyrocompass Heading from $HEHDT	$HEHDT	HEHDT_HDG
2	SCIGPS Latitude from $GPRMC	$GPRMC	GPRMC_LAT_DECDEG
3	SCIGPS Longitude from $GPRMC	$GPRMC	GPRMC_LON_DECDEG
4	SCIGPS COG from $GPRMC	$GPRMC	GPRMC_COG
5	SCIGPS SOG from $GPRMC	$GPRMC	GPRMC_SOG
6	Temperature from SBE48	SBE48	SBE48_TEMPERATURE

7	Speed from $VDVBW	$VDVBW	VDVBW_GROUND_SPEED_AHEAD

Notice that the sequencing in the abstract array is arbitrary and that values parsed from [INPUT 2] can appear at lower ABSTRACT_INDEX values than data parsed from [INPUT_1]. The ABSTRACT_INDEX values used do not need to be consecutive (it is acceptable for there to be gaps in the set of ABSTRACT_INDEX values used). It is possible to assign the same ABSTRACT_INDEX value to more than one parsed data value. In most cases this is not desired and will likely create very confusing results if done unintentionally. An example of a scenario where this could actually be useful is a situation where a user has two temperature sensors and wants an output that reflects the most recent value received from either of them and doesn't need to know which instrument the data is from. In contrast, using a single ABSTRACT_INDEX value in multiple places in the output fields is likely to be useful to many users.

Required dsLog Configuration changes

As noted above, the dsLogCSV process takes a set of UDP streams as its input. Typically, these UDP streams will be created by the dsLog process running on that ship, but dsLogCSV is also capable of receiving data over UDP directly from instruments. The dsLog generated timestamps in the incoming data are ignored and the dsLogCSV process assumes the data has been delivered in a timely manner and uses the current time from the system clock as the time to associate with data parsed from the incoming packet. The date and time strings contained in each output data line are generated from the computers system clock.

The initialization file for the dsLog process must include output specifiers that include mechanisms to create the UDP data streams needed by the dsLogCSV process.

Control of Input Data Parsing

A set of rules for parsing individual data elements from known data messages have been implemented. Future versions of the program will likely provide some more generic parsing abilities as well. The user needs to specify a single parsing rule for each field of interest in the input strings. The configuration file will have a section for each input with a section header of [INPUT_n]. Each input section will contain the parsing rules for the messages arriving on the UDP port associated with that input. For example an input receiving $GPRMC messages might have three parsing rules, one for latitude, one for longitude, and one for the date.

For example the string that is sent from dsLog for a Gyrocompass looks like:

```
NAV 2014/04/28 20:00:00.319 GYRO $HEDT,025.8,T*20
```

In order to specify that the 025.8 field be extracted from this string the user would specify the *PARSE_DATATYPE_n* to be *HEHDT_HDG.* This instructs the software to attempt to read a floating point value from the string and, if successful, to write the appropriate value to the ABSTRACT_DATA array and to update the associated timestamp.

There is a table in Appendix A that lists all of the parsing rules currently available.

Control of Output Data Streams

The user can specify a number of output streams. Each stream will have a corresponding initialization file section and can have multiple destinations. The destinations can be a mix of output files and UDP streams.

The heading for the initialization file section will be [OUTPUT_n] where n is an integer starting at one and increasing from there. The configuration file will contain some information that is common to all the output destinations the most important of which are the OUTPUT_FILE_FORMAT (controls whether the output files will be SAMOS style, NOAA style, or conventional DSLOGCSV style), the OUTPUT_INTERVAL (controls the number of seconds which elapse between generating output lines in the file), the NUMBER_OF_OUTPUT_FIELDS (which controls how many data fields are written on each output line, and finally a sequence of pairs of OUTFIELD_ABSTRACT_INDEX_n and OUTFIELD_HEADER_STRING_n which control the actual data written to the file. Additional information will allow control of file names and UDP destinations. All of these fields will be demonstrated in either the examples below or in the comprehensive list of initialization file elements below.

An Example Initialization File

Appendix B shows the dsLogCSV configuration file for the Atlantis as of 2014-10-22. This section will describe some of the basic dsLogCSV functions controlled by the configuration file and will reference the contents of Appendix B.

There is a [GENERAL] section. For the most basic use we need only to define the path where all data files will be written. It is possible to override this setting using other settings described later in the document.

```
# in the [GENERAL] section the ROOT_DIR is specified so that the dsLogCSV
# will know where to place output files.
[GENERAL]
ROOT_DIR="/underway/proc"
```

Following the GENERAL section are a series of [INPUT_n] sections each of which defines the processing for a specific UDP input to the dsLogCSV process. The first INPUT specifier is repeated immediately below. It defines the source of the SCIENCE-GPS data signal as a UDP stream generated by dsLog. So, we need to specify that it is a UDP source type and specify the UDP port. There are four specified data values to be parsed from the data on this INPUT. For each one we specify a sequence of parsing rules. Each parsing rule corresponds to a particular field of a particular message (for instance the COG field of the GPRMC message). Each parsing rule is minimally defined by a PARSE_DATATYPE_n and an associated ABSTRACT_INDEX_n. The 'n' field at the end of the parameter specifier is used to group the PARSE_DATATYPE and ABSTRACT_INDEX into the 'n-th' parsing rule for that input. The values for 'n' must start with 1 and be consecutive. The order of the rules is arbitrary and does not need to have any correlation with the order that the parsed values appear in the input sentences.

```
#------------------------------------------------
#--- Science GPS
#------------------------------------------------
[INPUT_1]
SOURCE_TYPE="UDP"
IN_SOCKET=55001
##switched source GPS - usually set to CNAV
```

```
NUMBER_OF_CSV_PARSE=7

# The minimum for each field is to specify what rule
# to use to parse the data and the ABSTRACT_INDEX
PARSE_DATATYPE_1="GPRMC_LAT_DECDEG"
PARSE_ABSTRACT_INDEX_1=2

PARSE_DATATYPE_2="GPRMC_LON_DECDEG"
PARSE_ABSTRACT_INDEX_2=3

PARSE_DATATYPE_3="GPRMC_COG"
PARSE_ABSTRACT_INDEX_3=4

PARSE_DATATYPE_4="GPRMC_SOG"
PARSE_ABSTRACT_INDEX_4=5

PARSE_DATATYPE_5="GPRMC_LAT_NOAA"
PARSE_ABSTRACT_INDEX_5=90

PARSE_DATATYPE_6="GPRMC_LON_NOAA"
PARSE_ABSTRACT_INDEX_6=91

PARSE_DATATYPE_7="GPRMC_COG_SAMOS"
PARSE_ABSTRACT_INDEX_7=94
```

An OUTPUT section defines the data that will be written to the data files. The OUTPUT_INTERVAL is in seconds and determines how often an output will be generated. For each desired data field in the output we need to specify the OUTFIELD_ABSTRACT_INDEX and the associated header string to write at the top of each data file (in OUTFIELD_HEADER_STRING_n).

Immediately below we see the specification for an output file containing PHINS data.

- OUTPUT_FILE_FORMAT – The OUTPUT_FILE_FORMAT is not specified for this input so the default DSLOGCSV style output file will be generated. (Other choices are SAMOS or NOAA).
- OUTPUT_ENABLE=1 – This instructs dsLogCSV to create and write to the specified file. It defaults to one, so if it is not specified then the file will be active.
- OUTPUT_INTERVAL=10 – This instructs dsLogCSV to create an output record every ten seconds on this output channel. The default value for OUTPUT_INTERVAL is 60.
- DESTINATION_PREFIX=”AT_IMU10_” – This specifies the prefix used to generate the data filenames for this output channel.
- DESTINATION_1_TYPE=”FILE” – This specifies that the first (and here the only) destination for this output channel is a file.
- NUMBER_OF_OUTPUT_FIELDS=10 – This specifies that each line of output will have ten data fields in addition to the GMT_DATE field and GMT_TIME field that are always present in DSLOGCSV style outputs.
- OUTFIELD_ABSTRACT_INDEX_1 – This specifies the ABSTRACT_INDEX for the data in the first data field of the output stream.
- OUTFIELD_HEADER_STRING_1 - This specifies the string placed in the HEADER line at the top of each file generated on this output stream.

```
#-----------------------------------------------
#  Create an IMU csv file for plotting
#----------------------------------------------
```

```
[OUTPUT_3]
OUTPUT_ENABLE=1

# specify the interval (in seconds) between output strings being created and output
#OUTPUT_INTERVAL=1
OUTPUT_INTERVAL=10

# specify the string to use in data file names
DESTINATION_PREFIX="AT_IMU10_"

# specify that the first output is to a file
DESTINATION_1_TYPE="FILE"

# specify the number of data fields in each output string
NUMBER_OF_OUTPUT_FIELDS=12

# For each output field specify the ABSTRACT_INDEX and
# a string to use for the header in data files

OUTFIELD_ABSTRACT_INDEX_1=70
OUTFIELD_HEADER_STRING_1="PHINS_HEADING"

OUTFIELD_ABSTRACT_INDEX_2=1
OUTFIELD_HEADER_STRING_2="HDT"

OUTFIELD_ABSTRACT_INDEX_3=71
OUTFIELD_HEADER_STRING_3="PHTRO_ROLL"

OUTFIELD_ABSTRACT_INDEX_4=80
OUTFIELD_HEADER_STRING_4="PHTRO_ROLL_FRAME"

OUTFIELD_ABSTRACT_INDEX_5=72
OUTFIELD_HEADER_STRING_5="PHTRO_PITCH"

OUTFIELD_ABSTRACT_INDEX_6=79
OUTFIELD_HEADER_STRING_6="PHTRO_PITCH_FRAME"

OUTFIELD_ABSTRACT_INDEX_7=73
OUTFIELD_HEADER_STRING_7="PHLIN_SURGE_LIN"

OUTFIELD_ABSTRACT_INDEX_8=74
OUTFIELD_HEADER_STRING_8="PHLIN_SWAY_LIN"

OUTFIELD_ABSTRACT_INDEX_9=74
OUTFIELD_HEADER_STRING_9="PHLIN_HEAVE_LIN"

OUTFIELD_ABSTRACT_INDEX_10=76
OUTFIELD_HEADER_STRING_10="PHSPD_SURGE_SPD"

OUTFIELD_ABSTRACT_INDEX_11=77
OUTFIELD_HEADER_STRING_11="PHSPD_SWAY_SPD"

OUTFIELD_ABSTRACT_INDEX_12=78
OUTFIELD_HEADER_STRING_12="PHSPD_HEAVE_SPD"
```

Calculated Values and Example Configuration File

The dsLogCSV application has the ability to derive calculated values such as SSV, BAROMETER, and TRUEWIND. The calculation of these values is part of the output process. When a line of data is written to the output port or file any defined calculated values in that output stream will be calculated and included in the output.

For each type of calculation (SSV, BAROMETER, TRUEWIND) the user can specify as many as eight unique sets of input parameters. The information for any single calculated value specification will be in its own initialization file section. This capability could be used to calculate TRUEWIND values for both a port and starboard sensor. For SSV calculations the header field for the initialization file section would be [SSV_1], [SSV_2], …. or [SSV_8]. There is a unique set of input specifications corresponding to each of the calculation types. The relevant sections of the example initialization file included in Appendix B are shown below.

Typically, if any of the raw inputs associated with that output stream have not been updated since the previous output event then the “NAN” string will be placed in the output where the calculated value would have gone. To override this behavior specify an age limit (in seconds) for all the raw data values associated with a particular calculated value using the “OUTFIELD_ENABLE_n” specifier.

It is possible to specify the precision (number of digits to the right of the decimal point) for any calculated value using the “OUTPUT_PRECISION” specifier.

```
### Quantities Calculated by dsLogCsv ###

#-----------------------------------------------
# SSV
# SBE45S and SBE48T
[SSV_1]
TEMPERATURE_ABSTRACT_INDEX=6
SALINITY_ABSTRACT_INDEX=12

# SSV_2
# SBE45T and SBE45S
# I know it calculates its own, but curious how our calculations compare to the SBE45s
[SSV_2]
TEMPERATURE_ABSTRACT_INDEX=10
SALINITY_ABSTRACT_INDEX=12

#-----------------------------------------------
# PORT BAROMETER
[BAROMETER_1]
RAW_PRESSURE_ABSTRACT_INDEX=27
SENSOR_HEIGHT=20.6

# Specifies number of digits to right of decimal point (default is 3)
OUTPUT_PRECISION=2

#-----------------------------------------------
# STBD BAROMETER
[BAROMETER_2]
RAW_PRESSURE_ABSTRACT_INDEX=37
SENSOR_HEIGHT=20.6

# Specifies number of digits to right of decimal point (default is 3)
OUTPUT_PRECISION=2

#-----------------------------------------------
# Port TRUEWIND
[TRUEWIND_1]
WIND_DIR_ABSTRACT_INDEX=21
WIND_SPEED_ABSTRACT_INDEX=23
COG_ABSTRACT_INDEX=4
SOG_ABSTRACT_INDEX=5
HDG_ABSTRACT_INDEX=1

# Specifies number of digits to right of decimal point (default is 3)
OUTPUT_PRECISION=2

#-----------------------------------------------
# Stbd TRUEWIND
[TRUEWIND_2]
WIND_DIR_ABSTRACT_INDEX=31
WIND_SPEED_ABSTRACT_INDEX=33
```

```
COG_ABSTRACT_INDEX=4
SOG_ABSTRACT_INDEX=5
HDG_ABSTRACT_INDEX=1

# Specifies number of digits to right of decimal point (default is 3)
OUTPUT_PRECISION=2
```

Once these calculated values have been defined, it is an easy process to include them in the output stream. For fields where a calculated value will be shown, the user needs to specify an “OUTFIELD_TYPE_n” value instead of an “OUTFIELD_ABSTRACT_INDEX_n” value as shown in the example initialization file segment below.

```
OUTFIELD_TYPE_21="TRUEWIND_SPEED_1"
OUTFIELD_HEADER_STRING_21="WXTP_TS"

OUTFIELD_TYPE_22="TRUEWIND_SPEED_2"
OUTFIELD_HEADER_STRING_22="WXTS_TS"

OUTFIELD_TYPE_23="TRUEWIND_DIRECTION_1"
OUTFIELD_HEADER_STRING_23="WXTP_TD"

OUTFIELD_TYPE_24="TRUEWIND_DIRECTION_2"
OUTFIELD_HEADER_STRING_24="WXTS_TD"
```

Multiple Identical Datatypes on a single input channel

Examining the details of the initialization file in Appendix B shows that there is data from two different Vaisalla instruments arriving on a single input. For a parsing field of type VAI_WIND_SPEED_AVG_1 any input string containing the substring ‘WXT’ will be parsed for the wind speed data. However, typically, the wind speed value needs to be sent to only one of the ABSTRACT-DATA array elements, either the one for port wind speed or the one for starboard wind speed. The MUST_CONTAIN directive is used to control the parsing in this situation. In the case of the Vaisalla the full prefix string generated by dsLog (“WXTP” vs. “WXTS”) is used to identify the source of the message. If a MUST_CONTAIN directive is included as part of the definition of a parsing rule then only strings that contain the specified substring will be parsed for that particular rule. If the substring is contained anywhere in the input string then the rule is satisfied.

The MUST_CONTAIN directive is specific to each defined parsing field. So, in the example initialization file in Appendix B the values for parsing rules 1, 2, and 3 each come from the PORT Vaisalla and we must include a MUST_CONTAIN_1, a MUST_CONTAIN_2, and a MUST_CONTAIN_3 directive, one for each of the values form the PORT string.

Note that a MUST_NOT_CONTAIN directive and a MUST_CONTAIN2 directive are also available to help control the parsing process. An input string must meet ALL of the defined rules in order for its data to be submitted to the parser for that particular field.

To illustrate the use of the MUST_CONTAIN directive the parsing rules for the two wind speeds (port and starboard) are shown below.

```
# Port Vai Wind Speed Avg
PARSE_DATATYPE_4="VAI_WIND_SPEED_AVG"
PARSE_ABSTRACT_INDEX_4=23
```

```
MUST_CONTAIN_4="WXTP"
```

```
# Stbd Vai Wind Speed Avg
PARSE_DATATYPE_14="VAI_WIND_SPEED_AVG"
PARSE_ABSTRACT_INDEX_14=33
MUST_CONTAIN_14="WXTS"
```

SAMOS Output Format

The dsLogCSV application is capable of creating SAMOS formatted output files. A special “OUTPUT_FILE_FORMAT” setting (“SAMOS”) was created for this data format, which directs dsLogCSV to create each line of data output using the distinctive SAMOS format. The [OUTPUT_n] section in the initialization file must also specify values for “SHIP_CALL_SIGN” and “SHIP_NAME”. With these changes the usual tools can be used to form the desired line of data outputs. Note that two special data output formats (“GPRMC_COG_SAMOS” and “HEHDT_HDG_SAMOS”) were implemented to facilitate the needs of the SAMOS files. The initialization file in the appendix provides a complete example of creating a SAMOS output format file. A single line of an example SAMOS file is shown below:

```
$SAMOS:001, CS:KAQP, YMD:20150226, HMS:000041, LA:-54.103, LO:-89.661, SOG:1.480, COG:248.7, GY:289.8,
SWR:66.078, LWR:335.600, SST:7.8683, SSC:3.53541, SAL:34.1126, WSTD:288.62, WSTS:4.19, WSAT:7.1, WSBP:1017.44,
WSRH:90.6, WSRC:28.520, WSRI:0.0, WSRD:353, WSRS:4.8, WPTD:277.32, WPTS:5.22, WPAT:6.9, WPBP:1017.74, WPRH:91.3,
WPRC:30.960, WPRI:0.0, WPRD:344, WPRS:5.9,
```

NOAA Output Format

The dsLogCSV application is capable of creating NOAA formatted output files. A special “OUTPUT_FILE_FORMAT” setting (“NOAA”) was created for this data format, which directs dsLogCSV to create a NOAA output file immediately upon execution and then at the top of each hour. The NOAA output file is always called “noaa.tmp”. At the top of each hour the existing “noaa.tmp” file is overwritten with a single line of data output using the required NOAA format.

The [OUTPUT_n] section in the initialization file must also specify values for “NOAA_VERSION”, “SHIP_CALL_SIGN” and “SHIP_NAME”. “NOAA_VERSION” is typically set to “SCS v4.0.0”. With these changes the usual tools can be used to form the desired line of data outputs. Note that four special data output formats (“GPGGA_LAT_NOAA”, “GPGGA_LON_NOAA”, “GPRMC_LAT_NOAA”, and “GPRMC_LON_NOAA”) were implemented to facilitate the needs of the NOAA files. The initialization file in the appendix provides a complete example of creating a NOAA output format file.

```
<<< ATLANTIS, 02/27/2015,18:00:45 >>>
<,KAQP,none,ddmm.mmmh,ddmm.mmmh,deg,m/s,C,%,hPa,C,Version,>
<,,5426.148S,08917.908W,310.56,10.97,8.3,85.6,993.04,7.8252,SCS v4.0.0,>
```

Post Processing Mode – dsLogPostCSV

A separate application, dsLogPostCSV, is available that allows users to create CSV data files from an existing database of previously collected dsLog data.

The operation of dsLogPostCSV is controlled by an initialization file which is similar to the configuration files used by dsLogCSV. There are some additional fields that must be specified by the user for post processing mode and a few of the settings specified for the real-time program (e.g. “SOURCE_TYPE” and “IN_SOCKET”) will not be relevant for post processing. Some of the additional settings must be specified in the “[POSTCSV]” section of the initialization file. These settings will specify the beginning and end times for the period that will be covered by the output data from the dsLogPostCSV.   All of these settings are described in detail below.

In addition the data inputs must be specified. These [INPUT_n] sections will be very similar to those used with real-time dsLogCSV program with “DATATYPE” and “ABSTRACT_INDEX” settings specified.   With post processing, the input data will be dsLog output files rather than UDP streams. So the data directory, filename prefix, and filename extension must be specified for each data stream. A ‘global’ input directory can be specified in the [GLOBAL] section with the “INPUT_DATA_DIR” setting. The specified directory will be used for all the inputs except those where the user chooses to override the global input directory by specifying an “INFILE_DIRECTORY” within the [INPUT_n] section. Use “INFILE_PREFIX” and “INFILE_EXTENSION” to specify the filenames used for each input. All these parameters are fully described in the sections below.

The user can control how often new output files are opened. Global settings can be set in the [GENERAL] section using the “SINGLE_LOGFILE_FLAG” and “”DAILY_LOGFILES_FLAG” settings. These can be overridden in each OUTPUT_n section with identically named setting specifiers there. It is important to note that the SINGLE_LOGFILE_FLAG takes precedence. If this flag is set to one, then a single logfile will be output no matter what DAILY_LOGFILES_FLAG is set to. So, if the global SINGLE_LOGFILE_FLAG is set to one and daily logfiles are desired for a particular output then the user must include both of the following two lines in the definition of that output:

```
SINGLE_LOGFILE_FLAG=0
DAILY_LOGFILES_FLAG=1
```

Overview of dsLogCSV initialization file

A configuration file is used to control the behavior of dsLogCSV. The initialization file consists of several sections. Each of these sections is delimited at the top of the section by a header line which contains the name of the section surrounded by square brackets (e.g. ***[INPUT_1]***). The main components of the initialization file are a general section for global information and then a section for each input and for each output.

An input will correspond to a specific UDP port. Each input has a corresponding initialization file section labeled [INPUT_N], where ‘N’ is an integer. The inputs must be numbered consecutively beginning with 1.

Initialization file syntax is strictly interpreted. For instance, if spaces exist before or after the ‘=’ in a line of the file it will be ignored.

The sections are enumerated below with brief descriptions of the individual parameters.

[GENERAL] section

- **ROOT_DIR**
    - Type – string
    - Stored as – startup.rootDirectory
    - Valid values – A string representing a valid path on the host machine.

    - Description – All logged data files will be placed in this directory.
    - Example syntax: ***ROOT_DIR="/data/csv"***
- **DAILY_LOGFILES_FLAG**
    - Type – integer
    - Stored as – globalDailyLogfilesFlag
    - Valid values – 0 or 1.
    - Default value – 1.
    - Description – if 0 then new output files are opened every hour. If 1 then new output files are opened at the start of each calendar day. This value can be overwritten for each individual output if needed.
- **SINGLE_LOGFILE_FLAG**
    - Type – integer
    - Stored as – globalSingleLogfileFlag
    - Valid values – 0 or 1.
    - Default value – 1.
    - Description – Used only in post-processing mode. If 0 then new output files are opened every day or every hour (depending upon value of DAILY_LOGFILES_FLAG). If 1 then a single output file will be created, containing all of the output data for the specified date range. This value can be overwritten for each individual output if needed.
- **INPUT_DATA_DIR**
    - Type – string
    - Stored as – postGlobalInputDirectory
    - Valid values – A string representing a valid path on the host machine.
    - Description – Used only in post-processing mode. Unless overridden, the post-processing code will look for data files in this directory.
    - Example syntax: ***INPUT_DATA_DIR="/data/csv"***

[POSTCSV] section

- **START_YEAR/END_YEAR**
    - Type – int
    - Stored as – postCsv.iniStartTime.year/postCsv.iniEndTime.year
    - Valid values – 1960 < year < 2200.
    - Description – Specifies the YEAR of the date and time to begin/end outputting CSV data.
- **START_MONTH/END_MONTH**
    - Type – int
    - Stored as – postCsv.iniStartTime.month/postCsv.iniEndTime.month
    - Valid values – 1 <= month <= 12.
    - Description – Specifies the MONTH of the date and time to begin/end outputting CSV data.
- **START_DAY/END_DAY**
    - Type – int
    - Stored as – postCsv.iniStartTime.day/postCsv.iniEndTime.day
    - Valid values – 1 <= day <= lastDayOfSpecifiedMonth.

    - Description – Specifies the DAY of the date and time to begin/end outputting CSV data.
- **START_HOUR/END_HOUR**
    - Type – int
    - Stored as – postCsv.iniStartTime.hour/postCsv.iniEndTime.hour
    - Valid values – 0 <= hour <= 23.
    - Description – Specifies the HOUR of the date and time to begin/end outputting CSV data.
- **START_MINUTE/END_MINUTE**
    - Type – int
    - Stored as – postCsv.iniStartTime.minute/postCsv.iniEndTime.minute
    - Valid values – 0 <= hour <= 59.
    - Description – Specifies the MINUTE of the date and time to begin/end outputting CSV data.
- **START_SECOND/END_SECOND**
    - Type – int
    - Stored as – postCsv.iniStartTime.second/postCsv.iniEndTime.second
    - Valid values – 0 <= hour <= 59.
    - Description – Specifies the SECOND of the date and time to begin/end outputting CSV data.

[INPUT_n] section

- **SOURCE_TYPE**
    - Type – string
    - Stored as – loggingList[x].sourceType (stored as an integer)
    - Valid values – "UDP" or "FILE".
    - Not used in post-processing mode.
    - Description – Used to specify whether this data comes from a UDP port or a logged file. Presently only UDP input is supported.
- **IN_SOCKET**
    - Type – int
    - Stored as – loggingList[x].inSocketNumber
    - Valid values – Any valid UDP socket number.
    - Not used in post-processing mode.
    - Description – Specifies the UDP socket upon which the UDP data will be received.
- **INFILE_DIRECTORY**
    - Type – string
    - Stored as – loggingList[x]. infileDirectory
    - Valid values – A string representing a valid path on the host machine.
    - Used only in post-processing mode.
    - Description – Specifies the path to a directory where the dsLog data files for this input are stored. Can be omitted if the global input directory is valid and points to the proper directory.
- **INFILE_PREFIX**
    - Type – string
    - Stored as – loggingList[x].infilePrefix

  - Valid values – A string composed of characters valid in Unix filenames.
  - Used only in post-processing mode.
  - Description – Used to specify the prefix portion of the data filenames.
- **INFILE_EXTENSION**
  - Type – string
  - Stored as – loggingList[x].infileExtension
  - Valid values – A string composed of characters valid in Unix filenames.
  - Used only in post-processing mode.
  - Description – Used to specify the extension portion of the data filenames.
- **INPUT_ENABLE**
  - Type – integer
  - Stored as – not stored
  - Valid values – 0, 1.
  - Default value – 1.
  - Description – If a zero value is specified then this input specifier will be ignored.
- **NUMBER_OF_CSV_PARSE**
  - Type – integer
  - Stored as – loggingList[x].numberOfCsvParse
  - Valid values – 1 thru 255.
  - Description – for this input the dsLogCSV will use this number of parsing rules to attempt to extract data from the received strings.
- **PARSE_ABSTRACT_INDEX_n**
  - Type – integer
  - Stored as – loggingList[x]. csvParse[n].parseAbstractIndex
  - Valid values – 1 thru 511.
  - Description – This value controls where parsed data is stored in the ABSTRACT_DATA array.
- **PARSE_DATATYPE_n**
  - Type – Specified as string, stored as integer (based on table).
  - Stored as – loggingList[x]. csvParse[n].parseDataType
  - Valid values – “GPRMC_DECLAT”, “GPRMC_DECLON”, or any other entry from the RULE column in the table in the Appendix.
  - Description – This value specifies which parsing rule to use to attempt to extract data from received strings.
- **MUST_CONTAIN_n**
  - Type – string
  - Stored as – loggingList[x].csvParse[n].mustContain
  - Valid values – Any string.
  - Description – This parsing rule will only be applied to input lines containing this string. If a **MUST_CONTAIN2_n** string is also specified then only input lines containing both strings will be searched. If no **MUST_CONTAIN_n** or **MUST_CONTAIN2_n** strings are specified then all input lines will be searched.

- **MUST_NOT_CONTAIN_n**
    - Type – string
    - Stored as – loggingList[x].csvParse[n].mustContain
    - Valid values – Any string.
    - Description – This parsing rule will only be applied to input lines containing this string. If no **MUST_NOT_CONTAIN_n** string is specified then all input lines will be searched.

[OUTPUT_n] section

- **OUTPUT_ENABLE**
    - Type – integer
    - Stored as – not stored
    - Valid values – 0, 1.
    - Default value – 1.
    - Description – If a zero value is specified then this output specifier will be ignored.
- **OUTPUT_INTERVAL**
    - Type – integer
    - Stored as – outputList[x].outputInterval
    - Valid values – Any integer greater than zero.
    - Default value – 60.
    - Description – This value specifies the number of seconds that will elapse between output records being generated for this output channel.
- **DATE_HEADER_STRING**
    - Type – string
    - Stored as – outputList[x]. dateHeaderString
    - Valid values – Any string.
    - Default value – DATE_GMT.
    - Description – A double quote delimited string specifying the string that will appear in the date column of the header row of default format (Not SAMOS or NOAA) output files.
- **TIME_HEADER_STRING**
    - Type – string
    - Stored as – outputList[x]. timeHeaderString
    - Valid values – Any string.
    - Default value – TIME_GMT.
    - Description – A double quote delimited string specifying the string that will appear in the time column of the header row of default format (Not SAMOS or NOAA) output files.
- **OUTPUT_PREFIX_1**
    - Type – string
    - Stored as – outputList[x]. outputPrefix1
    - Valid values – Any string.
    - Default value – null string.

- Description – A double quote delimited string specifying the string that will appear as the leftmost field in each output record.

- **OUTPUT_PREFIX_2**
  - Type – string
  - Stored as – outputList[x]. outputPrefix2
  - Valid values – Any string.
  - Default value – null string.
  - Description – A double quote delimited string specifying the string that will appear in each output record after the timestamp but before the data fields.
- **OUTPUT_DELIMITER**
  - Type – string
  - Stored as – outputList[x]. outputDelimiter
  - Valid values – Any string.
  - Default value – “, “ (a comma followed by a space).
  - Description – A double quote delimited string specifying the string that will appear between data elements in the output records.
- **BAD_DATA_STRING**
  - Type – string
  - Stored as – outputList[x]. badDataString
  - Valid values – Any string.
  - Default value – “NAN“.
  - Description – A double quote delimited string specifying the string that will appear in the output records for any field where data has not been received since the last output.
- **NUMBER_OF_OUTPUT_FIELDS**
  - Type – int
  - Stored as – outputList[x].numberOfOutputFields
  - Valid values – An integer between 1 and 511.
  - Description – The number of output fields that will be included in this output stream.
- **OUTFIELD_ABSTRACT_INDEX_n**
  - Type – integer
  - Stored as – outputList[x].outfieldAbstractIndex[n]
  - Valid values – Any unique integer value. These values do not need to be consecutive.
  - Description – This unique identifier will specify which element in the ABSTRACT_DATA array will be placed in this field of the output string.
- **OUTFIELD_HEADER_STRING_n**
  - Type – string
  - Stored as – outputList[x].outfieldHeaderString[n]
  - Valid values – Any string.
  - Default value – null string.
  - Description – This string will be used as the column heading for this data element in output files.
- **OUTPUT_FILE_FORMAT**

  - Type – Specified as string. Stored as integer.
  - Stored as – outputList[x].outputFileFormat
  - Valid values – DSLOGCSV, SAMOS, NOAA.
  - Default value – DSLOGCSV.
  - Description – This value controls the basic output file structure. SAMOS style, NOAA style, or DSLOGCSV style file structures can be specified.
- **SHIP_CALL_SIGN**
  - Type – string
  - Stored as – outputList[x].shipCallSign
  - Valid values – Any string.
  - Default value – BADINI_SHIP_CALL_SIGN.
  - Description – This string will be used to fill in the appropriate fields in NOAA and SAMOS style output files.
- **SHIP_NAME**
  - Type – string
  - Stored as – outputList[x].shipName
  - Valid values – Any string.
  - Default value – BADINI_SHIP_NAME.
  - Description – This string will be used to fill in the appropriate fields in NOAA style output files.
- **NOAA_VERSION**
  - Type – string
  - Stored as – outputList[x].noaaVersionString
  - Valid values – Any string.
  - Default value – BADINI_NOAA_VERSION.
  - Description – This string will be used to fill in the appropriate fields in NOAA style output files.
- **DESTINATION_PREFIX**
  - Type – string
  - Stored as – outputList[x].logging.filenamePrefix
  - Valid values – Any string.
  - Default value – "CSV".
  - Description – This string will be used when generating filenames for output files.
- **USE_FILE_HEADER**
  - Type – integer
  - Stored as – outputList[x].logging.useFileHeaderFlag
  - Valid values – 0 or 1.
  - Default value – 1.
  - Description – If this value is one (the default) then each data file will start with a line showing the column headers. Set to zero to suppress these headers.
- **DAILY_LOGFILES_FLAG**
  - Type – integer
  - Stored as – outputList[x].logging.dailyLogfilesFlag

    - Valid values – 0 or 1
    - Default value – the value from the [GENERAL] section.
    - Description – if 0 then new output files are opened every hour. If 1 then new output files are opened at the start of each calendar day.
- **DESTINATION_n_ENABLE**
    - Type – integer.
    - Stored as – outputList[x].logging.destinations[n].destinationEnable
    - Valid values – 0 or 1.
    - Default value – 1.
    - Description – Set to zero to disable the destination.
- **DESTINATION_n_TYPE**
    - Type – Specified as string. Stored as integer.
    - Stored as – outputList[x].logging.destinations[n].destinationType
    - Valid values
        - "UDP" - 0 (UDP_SOCKET)
        - "FILE" - 1 (DISK)
    - Description – Specifies the type of the logging destination (disk vs. network).
- **DESTINATION_n_IPADDRESS**
    - Type – string
    - Stored as – outputList[x]. logging.destinations[n].networkDestination.ipAddress
    - Valid values – A double quote delimited string specifying the IP_ADDRESS of the UDP destination (e.g. "192.168.1.44").
    - Description – For network destination types this variable specifies the IP_ADDRESS of the destination.
- **DESTINATION_n_SOCKET**
    - Type – int
    - Stored as – outputList[x]. logging.destinations[n].networkDestination.toSocketNumber
    - Valid values – A valid socket number.
    - Description – For network destination types this variable specifies the SOCKET_NUMBER of the destination.
- **DESTINATION_n _PATHNAME**
    - Type – string
    - Stored as – outputList[x].logging. destinations[n].loggingDestination.loggingDirectory
    - Valid values –
    - Description – For disk type destinations, this value will specify the base directory for logged data. It will default to the directory specified in [GENERAL]->ROOT_DIR.
- **DESTINATION_n _FILENAME_PREFIX**
    - Type – string
    - Stored as – outputList[x]. destinations[n].loggingDestination.filenamePrefix
    - Valid values –
    - Description – For disk type destinations, filenames will consist of the PREFIX_STRING, then the date/time stamp, then a '.', and finally, the EXTENSION_STRING.

- **DESTINATION_n _EXTENSION**
  - Type – string
  - Stored as – outputList[x]. destinations[n].loggingDestination.fileExtension
  - Valid values –
  - Description – For disk type destinations, filenames will consist of the PREFIX_STRING, then the date/time stamp, then a '.', and finally, the EXTENSION_STRING.

Appendix A – Table of Parsing Rules

In the table below the target data fields in the example input strings are shown in red. Note that in some cases multiple fields are parsed and combined to yield the desired quantity.

RULE	DESCRIPTION
GPGGA_UTC_TIME	Time from a NMEA GPGGA string. $GPGGA,**150001.06**,2445.0499,S,00642.6807,E,1,10,00.9,28.4,M,23.1,M,7.0,0138*56 Time will be written out in format: hh:mm:ss.ss
GPGGA_UTC_HOUR	Hours of time from a NMEA GPGGA string. $GPGGA,**15**0001.06,2445.0499,S,00642.6807,E,1,10,00.9,28.4,M,23.1,M,7.0,0138*56
GPGGA_UTC_MINUTE	Minutes of time from a NMEA GPGGA string. $GPGGA,15**00**01.06,2445.0499,S,00642.6807,E,1,10,00.9,28.4,M,23.1,M,7.0,0138*56
GPGGA_UTC_SECOND	Seconds of time from a NMEA GPGGA string. $GPGGA,1500**01**.06,2445.0499,S,00642.6807,E,1,10,00.9,28.4,M,23.1,M,7.0,0138*56
GPGGA_UTC_FRACTION	Fractions of seconds from a NMEA GPGGA string. $GPGGA,150001.**06**,2445.0499,S,00642.6807,E,1,10,00.9,28.4,M,23.1,M,7.0,0138*56
GPGGA_LAT_DECDEG	Latitude in decimal degrees from a NMEA GPGGA string. $GPGGA,150001.06,**2445.0499,S**,00642.6807,E,1,10,00.9,28.4,M,23.1,M,7.0,0138*56 Latitude is written in format: dd.ddd (d is degrees)
GPGGA_LAT_DECMIN	Latitude in decimal degrees from a NMEA GPGGA string. $GPGGA,150001.06,**2445.0499,S**,00642.6807,E,1,10,00.9,28.4,M,23.1,M,7.0,0138*56 Latitude is written in format: dd mm.mm (d is degrees, m is minutes)
GPGGA_LAT_DECSEC	Latitude in decimal degrees from a NMEA GPGGA string. $GPGGA,150001.06,**2445.0499,S**,00642.6807,E,1,10,00.9,28.4,M,23.1,M,7.0,0138*56 Latitude is written in format: dd mm ss.sss (d is degrees, m is minutes, s is seconds)
GPGGA_LAT_NOAA	Latitude in decimal degrees from a NMEA GPGGA string. $GPGGA,150001.06,**2445.0499,S**,00642.6807,E,1,10,00.9,28.4,M,23.1,M,7.0,0138*56 Latitude is written in format: ddmm.mmmh (d is degrees, m is minutes, h is hemisphere)
GPGGA_LON_DECDEG	Longitude in decimal degrees from a NMEA GPGGA string. $GPGGA,150001.06,2445.0499,S,**00642.6807,E**,1,10,00.9,28.4,M,23.1,M,7.0,0138*56 Longitude is written in format: dd.ddd (d is degrees)
GPGGA_LON_DECMIN	Longitude in decimal degrees from a NMEA GPGGA string. $GPGGA,150001.06,2445.0499,S,**00642.6807,E**,1,10,00.9,28.4,M,23.1,M,7.0,0138*56 Longitude is written in format: dd mm.mm (d is degrees, m is minutes)
GPGGA_LON_DECSEC	Longitude in decimal degrees from a NMEA GPGGA string. $GPGGA,150001.06,2445.0499,S,**00642.6807,E**,1,10,00.9,28.4,M,23.1,M,7.0,0138*56 Longitude is written in format: dd mm ss.sss (d is degrees, m is minutes, s is seconds)
GPGGA_LON_NOAA	Longitude in decimal degrees from a NMEA GPGGA string. $GPGGA,150001.06,2445.0499,S,**00642.6807,E**,1,10,00.9,28.4,M,23.1,M,7.0,0138*56 Longitude is written in format: dddmm.mmmh (d is degrees, m is minutes, h is hemisphere)
GPGGA_QUALITY	Fix quality from a NMEA GPGGA string. $GPGGA,150001.06,2445.0499,S,00642.6807,E,**1**,10,00.9,28.4,M,23.1,M,7.0,0138*56
GPGGA_NUM_SAT	Number of Satellites from a NMEA GPGGA string. $GPGGA,150001.06,2445.0499,S,00642.6807,E,1,**10**,00.9,28.4,M,23.1,M,7.0,0138*56
GPGGA_DILUTION	Horizontal dilution from a NMEA GPGGA string. $GPGGA,150001.06,2445.0499,S,00642.6807,E,1,10,**00.9**,28.4,M,23.1,M,7.0,0138*56
GPGGA_ALTITUDE	Altitude (meters above mean sea level) from a NMEA GPGGA string. $GPGGA,150001.06,2445.0499,S,00642.6807,E,1,10,00.9,**28.4**,M,23.1,M,7.0,0138*56
GPGGA_GEOID_HEIGHT	Height of Geoid from a NMEA GPGGA string. $GPGGA,150001.06,2445.0499,S,00642.6807,E,1,10,00.9,28.4,M,**23.1**,M,7.0,0138*56
GPGGA_FIX_AGE	Time in seconds since last DGPS update from a NMEA GPGGA string. $GPGGA,150001.06,2445.0499,S,00642.6807,E,1,10,00.9,28.4,M,23.1,M,**7.0**,0138*56
GPGGA_STATION_ID	DGPS Station ID number from a NMEA GPGGA string. $GPGGA,150001.06,2445.0499,S,00642.6807,E,1,10,00.9,28.4,M,23.1,M,7.0,**0138***56

GPRMB_STATUS	Data status from a NMEA GPRMB string. $GPRMB,**A**,4.08,L,003,004,5130.02,N,00046.34,W,004.6,213.9,122.9,A*3D
GPRMB_CTE_MAG	Cross track error magnitude from a NMEA GPRMB string. $GPRMB,A,**4.08**,L,003,004,5130.02,N,00046.34,W,004.6,213.9,122.9,A*3D
GPRMB_CTE_DIR	Cross track error direction from a NMEA GPRMB string. $GPRMB,A,4.08,**L**,003,004,5130.02,N,00046.34,W,004.6,213.9,122.9,A*3D
GPRMB_ORIGIN_ID	Origin waypoint ID from a NMEA GPRMB string. $GPRMB,A,4.08,L,**003**,004,5130.02,N,00046.34,W,004.6,213.9,122.9,A*3D
GPRMB_DEST_ID	Destination ID from a NMEA GPRMB string. $GPRMB,A,4.08,L,003,**004**,5130.02,N,00046.34,W,004.6,213.9,122.9,A*3D
GPRMB_DEST_LAT	Destination waypoint latitude from a NMEA GPRMB string. $GPRMB,A,4.08,L,003,004,**5130.02,N**,00046.34,W,004.6,213.9,122.9,A*3D
GPRMB_DEST_LON	Destination waypoint longitude from a NMEA GPRMB string. $GPRMB,A,4.08,L,003,004,5130.02,N,**00046.34,W**,004.6,213.9,122.9,A*3D
GPRMB_RANGE	Range (in nautical miles) to destination from a NMEA GPRMB string. $GPRMB,A,4.08,L,003,004,5130.02,N,00046.34,W,**004.6**,213.9,122.9,A*3D
GPRMB_BEARING	Bearing to destination from a NMEA GPRMB string. $GPRMB,A,4.08,L,003,004,5130.02,N,00046.34,W,004.6,**213.9**,122.9,A*3D
GPRMB_DEST_SPEED	Velocity (in knots) towards destination from a NMEA GPRMB string. $GPRMB,A,4.08,L,003,004,5130.02,N,00046.34,W,004.6,213.9,**122.9**,A*3D
GPRMB_ARRIVAL_ALARM	Arrival alarm from a NMEA GPRMB string. $GPRMB,A,4.08,L,003,004,5130.02,N,00046.34,W,004.6,213.9,122.9,**A***3D
GPRMC_UTC_TIME	Time from a NMEA GPRMC string. $GPRMC,**200000.00**,A,4131.43576,N,07040.33630,W,0.01,270.0,280414,0.0,E*75 Time will be written out in format: hh:mm:ss.ss
GPRMC_ UTC_HOUR	Hours of time from a NMEA GPRMC string. $GPRMC,**20**0000.00,A,4131.43576,N,07040.33630,W,0.01,270.0,280414,0.0,E*75
GPRMC_ UTC_MINUTE	Minutes of time from a NMEA GPRMC string. $GPRMC,20**00**00.00,A,4131.43576,N,07040.33630,W,0.01,270.0,280414,0.0,E*75
GPRMC_ UTC_SECOND	Seconds of time from a NMEA GPRMC string. $GPRMC,2000**00**.00,A,4131.43576,N,07040.33630,W,0.01,270.0,280414,0.0,E*75
GPRMC_ UTC_FRACTION	Fractions of seconds of time from a NMEA GPRMC string. $GPRMC,200000.**00**,A,4131.43576,N,07040.33630,W,0.01,270.0,280414,0.0,E*75
GPRMC_RCVR_STATUS	Receiver status from a NMEA GPRMC string. $GPRMC,200000.00,**A**,4131.43576,N,07040.33630,W,0.01,270.0,280414,0.0,E*75
GPRMC_LAT_DECDEG	Decimal latitude from a NMEA GPRMC string. $GPRMC,200000.00,A,**4131.43576,N**,07040.33630,W,0.01,270.0,280414,0.0,E*75 Latitude is written in format: dd.ddd (d is degrees)
GPRMC_LAT_DECMIN	Decimal latitude from a NMEA GPRMC string. $GPRMC,200000.00,A,**4131.43576,N**,07040.33630,W,0.01,270.0,280414,0.0,E*75 Latitude is written in format: dd mm.mm (d is degrees, m is minutes)
GPRMC_LAT_DECSEC	Decimal latitude from a NMEA GPRMC string. $GPRMC,200000.00,A,**4131.43576,N**,07040.33630,W,0.01,270.0,280414,0.0,E*75 Latitude is written in format: dd mm ss.sss (d is degrees, m is minutes, s is seconds)
GPRMC_LAT_NOAA	Decimal latitude from a NMEA GPRMC string. $GPRMC,200000.00,A,**4131.43576,N**,07040.33630,W,0.01,270.0,280414,0.0,E*75 Latitude is written in format: ddmm.mmmh (d is degrees, m is minutes, h is hemisphere)
GPRMC_LON_DECDEG	Decimal longitude from a NMEA GPRMC string. $GPRMC,200000.00,A,4131.43576,N,**07040.33630,W**,0.01,270.0,280414,0.0,E*75 Longitude is written in format: dd.ddd (d is degrees)
GPRMC_LON_DECMIN	Decimal longitude from a NMEA GPRMC string. $GPRMC,200000.00,A,4131.43576,N,**07040.33630,W**,0.01,270.0,280414,0.0,E*75 Longitude is written in format: dd mm.mm (d is degrees, m is minutes)
GPRMC_LON_DECSEC	Decimal longitude from a NMEA GPRMC string. $GPRMC,200000.00,A,4131.43576,N,**07040.33630,W**,0.01,270.0,280414,0.0,E*75 Longitude is written in format: dd mm ss.sss (d is degrees, m is minutes, s is seconds)
GPRMC_LON_NOAA	Decimal longitude from a NMEA GPRMC string.

	$GPRMC,200000.00,A,4131.43576,N,**07040.33630,W**,0.01,270.0,280414,0.0,E*75 Longitude is written in format: dddmm.mmmh (d is degrees, m is minutes, h is hemisphere)
GPRMC_SOG	Speed-Over-Ground from a NMEA GPRMC string. $GPRMC,200000.00,A,4131.43576,N,07040.33630,W,**0.01**,270.0,280414,0.0,E*75
GPRMC_COG	Course-Over-Ground from a NMEA GPRMC string. $GPRMC,200000.00,A,4131.43576,N,07040.33630,W,0.01,**270.0**,280414,0.0,E*75 COG is written in format: ddd.ddd (d is degrees. Without leading zeros on integer part.)
GPRMC_COG_SAMOS	Course-Over-Ground from a NMEA GPRMC string. $GPRMC,200000.00,A,4131.43576,N,07040.33630,W,0.01,**270.0**,280414,0.0,E*75 COG is written in format: ddd.d (d is degrees. With leading zeros on integer part.)
GPRMC_ UTC_DATE	Date from a NMEA GPRMC string. $GPRMC,200000.00,A,4131.43576,N,07040.33630,W,0.01,270.0,**280414**,0.0,E*75 Date will be written in the format: DDMMYY
GPRMC_ UTC_YEAR	Year of date from a NMEA GPRMC string. $GPRMC,200000.00,A,4131.43576,N,07040.33630,W,0.01,270.0,2804**14**,0.0,E*75
GPRMC_ UTC_MONTH	Month of date from a NMEA GPRMC string. $GPRMC,200000.00,A,4131.43576,N,07040.33630,W,0.01,270.0,28**04**14,0.0,E*75
GPRMC_ UTC_DAY	Day of date from a NMEA GPRMC string. $GPRMC,200000.00,A,4131.43576,N,07040.33630,W,0.01,270.0,**28**0414,0.0,E*75
GPRMC_MAGVAR	Magnetic variation from a NMEA GPRMC string. $GPRMC,200000.00,A,4131.43576,N,07040.33630,W,0.01,270.0,280414,**0.0,E***75
GPVTG_TRUE_TRACK	Track made good from a NMEA GPVTG string. $GPVTG,**3.8**,T,19.5,M,9.7,N,18.0,K*4F
GPVTG_MAGNETIC_TRACK	Track from a NMEA GPVTG string. $GPVTG,3.8,T,**19.5**,M,9.7,N,18.0,K*4F
GPVTG_SPEED_KNOTS	Speed (in knots) from a NMEA GPVTG string. $GPVTG,3.8,T,19.5,M,**9.7**,N,18.0,K*4F
GPVTG_SPEED_KPH	Speed (in km/hour) from a NMEA GPVTG string. $GPVTG,3.8,T,19.5,M,9.7,N,**18.0**,K*4F
GPZDA_UTC_TIME	UTC time from a NMEA GPZDA string. $GPZDA,**110023.20**,24,3,2013,+0,+0*53 Time will be written out in format: hh:mm:ss.ss
GPZDA_UTC_HOUR	Hours of UTC time from a NMEA GPZDA string. $GPZDA,**11**0023.20,24,3,2013,+0,+0*53
GPZDA_UTC_MINUTE	Minutes of UTC time from a NMEA GPZDA string. $GPZDA,11**00**23.20,24,3,2013,+0,+0*53
GPZDA_UTC_SECOND	Seconds of UTC time from a NMEA GPZDA string. $GPZDA,1100**23**.20,24,3,2013,+0,+0*53
GPZDA_UTC_FRACTION	Fractions of seconds of UTC time from a NMEA GPZDA string. $GPZDA,110023.**20**,24,3,2013,+0,+0*53
GPZDA_UTC_DATE	Date from a NMEA GPZDA string. $GPZDA,110023.20,**24,3,2013**,+0,+0*53 Date will be written in the format: YYYY-MM-DD
GPZDA_UTC_YEAR	Year of date from a NMEA GPZDA string. $GPZDA,110023.20,24,3,**2013**,+0,+0*53
GPZDA_UTC_MONTH	Month of date from a NMEA GPZDA string. $GPZDA,110023.20,24,**3**,2013,+0,+0*53
GPZDA_UTC_DAY	Day of date from a NMEA GPZDA string. $GPZDA,110023.20,**24**,3,2013,+0,+0*53
GPZDA_LOCALZONE_HOURS	Localzone hours from a NMEA GPZDA string. $GPZDA,110023.20,24,3,2013,**+0**,+0*53
GPZDA_LOCALZONE_MINUTES	Localzone minutes from a NMEA GPZDA string. $GPZDA,110023.20,24,3,2013,+0,**+0***53
HEHDT_HDG	Heading from an $HEHDT string. $HEHDT,**025.8**, T*20 Heading is written in format: ddd.ddd (d is degrees. Without leading zeros on integer part.)
HEHDT_HDG_SAMOS	Heading from an $HEHDT string.

	$HEHDT,**025.8**, T*20 Heading is written in format: ddd.d (d is degrees. With leading zeros on integer part.)
GPHDT_HDG	Heading from an $GPHDT string. $GPHDT,**025.8**, T*20
SBE45_TEMPERATURE	Temperature from SBE45 string. SBE45 **19.3599**, 4.28013, 31.3199, 1515.556
SBE45_CONDUCTIVITY	Conductivity from SBE45 string. SBE45 19.3599, **4.28013**, 31.3199, 1515.556
SBE45_SALINITY	Salinity from SBE45 string. SBE45 19.3599, 4.28013, **31.3199**, 1515.556
SBE45_SOUNDVELOCITY	Sound velocity from SBE45 string. SBE45 19.3599, 4.28013, 31.3199, **1515.556**
SBE48_TEMPERATURE	Temperature from SBE48 string. SBE48 # **8.8622**, 28 Apr 2014, 20:05:56
FLR_MILLIVOLTS	Millivolts from FLR string. FLR ***+02156.80**
VAI_ADDRESS	Instrument address from WXT* string. **P**R0,Dm=079D,Sn=0.2M,Sm=0.6M,Sx=0.9M,Ta=11.0C,Ua=60.3P,Pa=1017.2H,Rc=45.81M,Ri=0.0M
VAI_WIND_DIRECTION_AVG	Average wind direction from WXT* string. PR0,Dm=**079**D,Sn=0.2M,Sm=0.6M,Sx=0.9M,Ta=11.0C,Ua=60.3P,Pa=1017.2H,Rc=45.81M,Ri=0.0M
VAI_WIND_SPEED_MIN	Minimum wind speed from WXT* string. PR0,Dm=079D,Sn=**0.2**M,Sm=0.6M,Sx=0.9M,Ta=11.0C,Ua=60.3P,Pa=1017.2H,Rc=45.81M,Ri=0.0M
VAI_WIND_SPEED_AVG	Average wind speed from WXT* string. PR0,Dm=079D,Sn=0.2M,Sm=**0.6**M,Sx=0.9M,Ta=11.0C,Ua=60.3P,Pa=1017.2H,Rc=45.81M,Ri=0.0M
VAI_WIND_SPEED_MAX	Maximum wind speed from WXT* string. PR0,Dm=079D,Sn=0.2M,Sm=0.6M,Sx=**0.9**M,Ta=11.0C,Ua=60.3P,Pa=1017.2H,Rc=45.81M,Ri=0.0M
VAI_TEMPERATURE	Temperature from WXT* string. PR0,Dm=079D,Sn=0.2M,Sm=0.6M,Sx=0.9M,Ta=**11.0**C,Ua=60.3P,Pa=1017.2H,Rc=45.81M,Ri=0.0M
VAI_HUMIDITY	Humidity from WXT* string. PR0,Dm=079D,Sn=0.2M,Sm=0.6M,Sx=0.9M,Ta=11.0C,Ua=**60.3**P,Pa=1017.2H,Rc=45.81M,Ri=0.0M
VAI_PRESSURE	Pressure from WXT* string. PR0,Dm=079D,Sn=0.2M,Sm=0.6M,Sx=0.9M,Ta=11.0C,Ua=60.3P,Pa=**1017.2**H,Rc=45.81M,Ri=0.0M
VAI_RAIN_ACCUMULATION	Rain accumulation from WXT* string. PR0,Dm=079D,Sn=0.2M,Sm=0.6M,Sx=0.9M,Ta=11.0C,Ua=60.3P,Pa=1017.2H,Rc=**45.81**M,Ri=0.0M
VAI_RAIN_INTENSITY	Rain intensity from WXT* string. PR0,Dm=079D,Sn=0.2M,Sm=0.6M,Sx=0.9M,Ta=11.0C,Ua=60.3P,Pa=1017.2H,Rc=45.81M,Ri=**0.0**M
VDVBW_WATER_SPEED_AHEAD	Speed ahead thru the water from $VDVBW string. $VDVBW,-00.04,-00.03,A,000.03,000.06,A,-00.03,V,000.06,V*4B
VDVBW_ WATER _SPEED_STBD	Speed starboard thru the water from $VDVBW string. $VDVBW,-00.04,-00.03,A,000.03,000.06,A,-00.03,V,000.06,V*4B
VDVBW_WATER_SPEED_STATUS	Status of speed thru the water from $VDVBW string. $VDVBW,-00.04,-00.03,A,000.03,000.06,A,-00.03,V,000.06,V*4B
VDVBW_GROUND_SPEED_AHEAD	Speed ahead over ground from $VDVBW string. $VDVBW,-00.04,-00.03,A,000.03,000.06,A,-00.03,V,000.06,V*4B
VDVBW_GROUND_SPEED_STBD	Speed starboard over ground from $VDVBW string. $VDVBW,-00.04,-00.03,A,000.03,000.06,A,-00.03,V,000.06,V*4B
VDVBW_GROUND_SPEED_STATUS	Status of speed over ground from $VDVBW string. $VDVBW,-00.04,-00.03,A,000.03,000.06,A,-00.03,V,000.06,V*4B
SWR_RADIATION	Radiation in SWR string. SWR **134.273**
LWR_DOME_TEMPERATURE	Dome temperature in LWR string. LWR **285.46** 282.78 -12.2 372.3
LWR_BODY_TEMPERATURE	Body temperature in LWR string. LWR 285.46 **282.78** -12.2 372.3
LWR_THERMOPILE_VOLTAGE	Thermopile voltage in LWR string. LWR 285.46 282.78 **-12.2** 372.3
LWR_RADIATION	Radiation in LWR string.

	LWR 285.46 282.78 -12.2 **372.3**
PASHR_TIME	UTC time from $PASHR string. $PASHR,160000.000,197.56,T,+01.01,+03.29,-00.01,0.063,0.058,0.073,2,0*01
PASHR_HEADING	Heading in degrees from $PASHR string. $PASHR,160000.000,197.56,T,+01.01,+03.29,-00.01,0.063,0.058,0.073,2,0*01
PASHR_ROLL	Roll angle in degrees from $PASHR string. $PASHR,160000.000,197.56,T,+01.01,+03.29,-00.01,0.063,0.058,0.073,2,0*01
PASHR_PITCH	Pitch angle in degrees from $PASHR string. $PASHR,160000.000,197.56,T,+01.01,+03.29,-00.01,0.063,0.058,0.073,2,0*01
PASHR_HEAVE	Heave from $PASHR string. $PASHR,160000.000,197.56,T,+01.01,+03.29,-00.01,0.063,0.058,0.073,2,0*01
PASHR_ROLL_ACCURACY	Roll angle accuracy (Std. Dev.) in degrees from $PASHR string. $PASHR,160000.000,197.56,T,+01.01,+03.29,-00.01,0.063,0.058,0.073,2,0*01
PASHR_PITCH_ACCURACY	Pitch angle accuracy (Std. Dev.) in degrees from $PASHR string. $PASHR,160000.000,197.56,T,+01.01,+03.29,-00.01,0.063,0.058,0.073,2,0*01
PASHR_HEADING_ACCURACY	Heading angle accuracy (Std. Dev.) in degrees from $PASHR string. $PASHR,160000.000,197.56,T,+01.01,+03.29,-00.01,0.063,0.058,0.073,2,0*01
PASHR_GPS_FLAG	Aiding status from $PASHR string. $PASHR,160000.000,197.56,T,+01.01,+03.29,-00.01,0.063,0.058,0.073,2,0*01
PASHR_IMU_FLAG	IMU status from $PASHR string. $PASHR,160000.000,197.56,T,+01.01,+03.29,-00.01,0.063,0.058,0.073,2,0*01
PASHR_ATT_SECONDS	Seconds in the week from $PASHR,ATT string. $PASHR,ATT,74432.0000,35.3421,0.0000,0.0000,0.0013,19150.0205,1*15
PASHR_ATT_HEADING	Heading from $PASHR,ATT string. $PASHR,ATT,74432.0000,35.3421,0.0000,0.0000,0.0013,19150.0205,1*15
PASHR_ATT_PITCH	Pitch from $PASHR,ATT string. $PASHR,ATT,74432.0000,35.3421,0.0000,0.0000,0.0013,19150.0205,1*15
PASHR_ATT_ROLL	Roll from $PASHR,ATT string. $PASHR,ATT,74432.0000,35.3421,0.0000,0.0000,0.0013,19150.0205,1*15
PASHR_ATT_MRMS	MRMS from $PASHR,ATT string. $PASHR,ATT,74432.0000,35.3421,0.0000,0.0000,0.0013,19150.0205,1*15
PASHR_ATT_BRMS	BRMS from $PASHR,ATT string. $PASHR,ATT,74432.0000,35.3421,0.0000,0.0000,0.0013,19150.0205,1*15
PASHR_ATT_REAQUISITION	Reacquisition from $PASHR,ATT string. $PASHR,ATT,74432.0000,35.3421,0.0000,0.0000,0.0013,19150.0205,1*15
TSS1_ACCEL_HOR	Horizontal acceleration from MRU TSS1 string. :090048 0013F-0067 0035 The two characters indicated are read as hexadecimal and then scaled by 0.0383 to produce the final output.
TSS1_ACCEL_VER	Vertical acceleration from MRU TSS1 string. :090048 0013F-0067 0035 The four characters indicated are read as hexadecimal and then scaled by 0.000625 to produce the final output.
TSS1_HEAVE	Heave acceleration from MRU TSS1 string. :090048 0013F-0067 0035 The four characters indicated are read as hexadecimal and then scaled by 0.01 to produce the final output.
TSS1_PITCH	Pitch from MRU TSS1 string. :090048 0013F-0067 0035 The four characters indicated are read as hexadecimal and then scaled by 0.01 to produce the final output.
TSS1_ROLL	Roll from MRU TSS1 string. :090048 0013F-0067 0035 The four characters indicated are read as hexadecimal and then scaled by 0.01 to produce the final output.
TSS1_STATUS	Status from MRU TSS1 string.

	:090048 0013F-0067 0035
PKEL_DEPTH_1	First depth field in $PKEL string. $PKEL99,2828.,1,+005.00,2832.,1,+005.00,1500,22 37.033158N,110 01.688307W
PKEL_STATUS_1	First status field in $PKEL string. $PKEL99,2828.,1,+005.00,2832.,1,+005.00,1500,22 37.033158N,110 01.688307W
PKEL_DUCER_DEPTH_1	First transducer depth field in $PKEL string. $PKEL99,2828.,1,+005.00,2832.,1,+005.00,1500,22 37.033158N,110 01.688307W
PKEL_DEPTH_2	Second depth field in $PKEL string. $PKEL99,2828.,1,+005.00,2832.,1,+005.00,1500,22 37.033158N,110 01.688307W
PKEL_STATUS_2	Second status field in $PKEL string. $PKEL99,2828.,1,+005.00,2832.,1,+005.00,1500,22 37.033158N,110 01.688307W
PKEL_DUCER_DEPTH_2	Second transducer depth field in $PKEL string. $PKEL99,2828.,1,+005.00,2832.,1,+005.00,1500,22 37.033158N,110 01.688307W
PHINS_HEADING	Heading from PHINS $HEHDT string. $HEHDT,5.021,T*29
PHTRO_PITCH	Pitch from PHINS $PHTRO string. $PHTRO,1.72,M,0.34,T*4B
PHTRO_PITCH_FRAME	Pitch frame from PHINS $PHTRO string. $PHTRO,1.72,M,0.34,T*4B
PHTRO_ROLL	Roll from PHINS $PHTRO string. $PHTRO,1.72,M,0.34,T*4B
PHTRO_ROLL_FRAME	Roll frame from PHINS $PHTRO string. $PHTRO,1.72,M,0.34,T*4B
PHLIN_SURGE_LIN	Surge displacement in meters from PHINS $PHLIN string. $PHLIN,0.335,0.407,0.542*54
PHLIN_SWAY_LIN	Sway displacement in meters from PHINS $PHLIN string. $PHLIN,0.335,0.407,0.542*54
PHLIN_HEAVE_LIN	Heave displacement in meters from PHINS $PHLIN string. $PHLIN,0.335,0.407,0.542*54
PHSPD_SURGE_SPD	Surge speed in meters/second from PHINS $PHSPD string. $PHSPD,0.016,0.309,0.001*51
PHSPD_SWAY_SPD	Sway speed in meters/second from PHINS $PHSPD string. $PHSPD,0.016,0.309,0.001*51
PHSPD_HEAVE_SPD	Heave speed in meters/second from PHINS $PHSPD string. $PHSPD,0.016,0.309,0.001*51
KIDPT_DEPTH	Depth from KIDPT string. KIDPT,2812.42,6.17,12000.0*7c
KIDPT_BEAMS_GOOD	Beams_Good from KIDPT string. KIDPT,2812.42,6.17,12000.0*7c
KIDPT_FREQUENCY	Frequency from KIDPT string. KIDPT,2812.42,6.17,12000.0*7c

Appendix B – Example Configuration File 1

```
#Date, Time_GMT, Dec_Lat, Dec_Lon, SPD, HDT, COG, SOG, WXTP_Ta, WXTS_Ta, WXTP_Pa, WXTS_Pa, WXTP_Ri, WXTS_Ri,
WXTP_Rc, WXTS_Rc, WXTP_Dm, WXTS_Dm, WXTP_Sm, WXTS_Sm, WXTP_Ua, WXTS_Ua, WXTP_TS, WXTS_TS, WXTP_TD, WXTS_TD, SWR,
LWR, PAR, SBE45S, SBE48T, FLR, SSV, Depth12, Depth35, HdChkSum=00
#2013/03/23,00:00:01,-34.900096,-
56.202505,000.00,068.7,071.6,0.04,18.8,18.7,1017.4,1017.4,0.0,0.0,83.82,63.73,277,304,0.4,0.6,40.1,40.0,0.40,0.5
9,342.75,10.99,-10.591,296.5,7.38,28.7496,20.4439,00091.80,1524.37,nan,nan
#
#
#-----------------------------------------------
#
# created for running dsLogCsv on the R/V Atlantis
# t. thiel
# 2014-04-29
# update 20141015
# updated 20150130  lstolp
#-----------------------------------------------
#--- Abstract Index Table
#   1   Gyrocompass Heading from $HEHDT
#   2   SCIGPS Latitude from $GPRMC
#   3   SCIGPS Longitude from $GPRMC
#   4   SCIGPS COG from $GPRMC
#   5   SCIGPS SOG from $GPRMC
#   6   Temperature from SBE48
#   7   Speed from $VDVBW
#   8   LWR
#   9   SWR
#  10   Temperature from SBE45
#  11   Conductivity from SBE45
#  12   Salinity from SBE45
#  13   Sound velocity from SBE45
#  14
#  15   Fluorometer
#  16   Knudsen Depth12
#  17   Knudsen Depth 3.5
#  18   EM122  Center Depth KIDPT_DEPTH
#  19
#  20   Port Vaisala Address
#  21   Port Vaisala Wind Direction
#  22   Port Vaisala Wind Speed Min
#  23   Port Vaisala Wind Speed Avg
#  24   Port Vaisala Wind Speed Max
#  25   Port Vaisala Temperature
#  26   Port Vaisala Humidity
#  27   Port Vaisala Pressure
#  28   Port Vaisala Rain Accumulation
#  29   Port Vaisala Rain Intensity
#  30   Stbd Vaisala Address
#  31   Stbd Vaisala Wind Direction
#  32   Stbd Vaisala Wind Speed Min
#  33   Stbd Vaisala Wind Speed Avg
#  34   Stbd Vaisala Wind Speed Max
#  35   Stbd Vaisala Temperature
#  36   Stbd Vaisala Humidity
#  37   Stbd Vaisala Pressure
#  38   Stbd Vaisala Rain Accumulation
#  39   Stbd Vaisala Rain Intensity
#  40
#  41
#  42   CNAV GPS Latitude from $GPRMC
#  43   CNAV GPS Longitude from $GPRMC
#  44   CNAV GPS COG from $GPRMC
#  45   CNAV GPS SOG from $GPRMC
#  46   MX512 GPS Latitude from $GPRMC
#  47   MX512 GPS Longitude from $GPRMC
#  48   MX512 GPS COG from $GPRMC
#  49   MX512 GPS SOG from $GPRMC
#  50
#  51
#  52
#  53
#  54
#  60  Coda PASHR  PASHR_HEADING
#  61  Coda PASHR  PASHR_PITCH
#  62  Coda PASHR  PASHR_ROLL
#  63  Coda PASHR  PASHR_HEAVE
#  64  Coda PASHR  PASHR_ROLL_ACCURACY
#  65  Coda PASHR  PASHR_PITCH_ACCURACY
#  66  Coda PASHR  PASHR_HEADING_ACCURACY
#  67  Coda PASHR  PASHR_IMU_FLAG
#
#  70  PHINS HEHDT  PHINS_HEADING
#  71  PHINS PHTRO  PHTRO_ROLL
#  72  PHINS PHTRO  PHTRO_PITCH
#  73  PHINS PHLIN  PHLIN_SURGE_LIN
#  74  PHINS PHLIN  PHLIN_SWAY_LIN
#  75  PHINS PHLIN  PHLIN_HEAVE_LIN
#  76  PHINS PHSPD  PHSPD_SURGE_SPD
#  77  PHINS PHSPD  PHSPD_SWAY_SPD
#  78  PHINS PHSPD  PHSPD_HEAVE_SPD
#  79  PHINS PHTRO  PHTRO_PITCH_FRAME
#  80  PHINS PHTRO  PHTRO_ROLL_FRAME
#
#  90  SCIGPS GPRMC GPRMC_LAT_NOAA
#  91  SCIGPS GPRMC GPRMC_LON_NOAA
#  92  SCIGPS GPGGA GPGGA_LAT_NOAA
#  93  SCIGPS GPGGA GPGGA_LON_NOAA
#  94  SCIGPS GPRMC GPRMC_COG_SAMOS
#  95  GYRO   HEHDT HEHDT_GYROHDG_SAMOS
#
#
#
#
#
#
#-----------------------------------------------
#-----------------------------------------------

[GENERAL]

ROOT_DIR="/underway/proc"

#-----------------------------------------------
#--- Science GPS
#-----------------------------------------------
[INPUT_1]
SOURCE_TYPE="UDP"
IN_SOCKET=55001
##switched source GPS - usually set to CNAV

NUMBER_OF_CSV_PARSE=7

# The minimum for each field is to specify what rule
# to use to parse the data and the ABSTRACT_INDEX
PARSE_DATATYPE_1="GPRMC_LAT_DECDEG"
PARSE_ABSTRACT_INDEX_1=2

PARSE_DATATYPE_2="GPRMC_LON_DECDEG"
PARSE_ABSTRACT_INDEX_2=3

PARSE_DATATYPE_3="GPRMC_COG"
PARSE_ABSTRACT_INDEX_3=4

PARSE_DATATYPE_4="GPRMC_SOG"
PARSE_ABSTRACT_INDEX_4=5

PARSE_DATATYPE_5="GPRMC_LAT_NOAA"
PARSE_ABSTRACT_INDEX_5=90

PARSE_DATATYPE_6="GPRMC_LON_NOAA"
PARSE_ABSTRACT_INDEX_6=91
```

```
PARSE_DATATYPE_7="GPRMC_COG_SAMOS"
PARSE_ABSTRACT_INDEX_7=94

#-----------------------------------------------
#--- Gyrocompass
#-----------------------------------------------
[INPUT_2]
SOURCE_TYPE="UDP"
IN_SOCKET=55103

NUMBER_OF_CSV_PARSE=2

PARSE_DATATYPE_1="HEHDT_HDG"
PARSE_ABSTRACT_INDEX_1=1

PARSE_DATATYPE_2="HEHDT_HDG_SAMOS"
PARSE_ABSTRACT_INDEX_2=95

#-----------------------------------------------
#--- PHINS String
#-----------------------------------------------
[INPUT_3]
SOURCE_TYPE="UDP"
IN_SOCKET=55102

NUMBER_OF_CSV_PARSE=11

PARSE_DATATYPE_1="PHINS_HEADING"
PARSE_ABSTRACT_INDEX_1=70

PARSE_DATATYPE_2="PHTRO_ROLL"
PARSE_ABSTRACT_INDEX_2=71

PARSE_DATATYPE_3="PHTRO_PITCH"
PARSE_ABSTRACT_INDEX_3=72

PARSE_DATATYPE_4="PHLIN_SURGE_LIN"
PARSE_ABSTRACT_INDEX_4=73

PARSE_DATATYPE_5="PHLIN_SWAY_LIN"
PARSE_ABSTRACT_INDEX_5=74

PARSE_DATATYPE_6="PHLIN_HEAVE_LIN"
PARSE_ABSTRACT_INDEX_6=75

PARSE_DATATYPE_7="PHSPD_SURGE_SPD"
PARSE_ABSTRACT_INDEX_7=76

PARSE_DATATYPE_8="PHSPD_SWAY_SPD"
PARSE_ABSTRACT_INDEX_8=77

PARSE_DATATYPE_9="PHSPD_HEAVE_SPD"
PARSE_ABSTRACT_INDEX_9=78

PARSE_DATATYPE_10="PHTRO_PITCH_FRAME"
PARSE_ABSTRACT_INDEX_10=79

PARSE_DATATYPE_11="PHTRO_ROLL_FRAME"
PARSE_ABSTRACT_INDEX_11=80

#-----------------------------------------------
# MET Data on 485 loop
#-----------------------------------------------
[INPUT_4]
SOURCE_TYPE="UDP"
IN_SOCKET=55400

NUMBER_OF_CSV_PARSE=20

# Port Vai Address
```

```
PARSE_DATATYPE_1="VAI_ADDRESS"
PARSE_ABSTRACT_INDEX_1=20
MUST_CONTAIN_1="WXTP"

# Port Vai Wind Direction
PARSE_DATATYPE_2="VAI_WIND_DIRECTION_AVG"
PARSE_ABSTRACT_INDEX_2=21
MUST_CONTAIN_2="WXTP"

# Port Wind Speed Min
PARSE_DATATYPE_3="VAI_WIND_SPEED_MIN"
PARSE_ABSTRACT_INDEX_3=22
MUST_CONTAIN_3="WXTP"

# Port Vai Wind Speed Avg
PARSE_DATATYPE_4="VAI_WIND_SPEED_AVG"
PARSE_ABSTRACT_INDEX_4=23
MUST_CONTAIN_4="WXTP"

# Port Vai Wind Speed Max
PARSE_DATATYPE_5="VAI_WIND_SPEED_MAX"
PARSE_ABSTRACT_INDEX_5=24
MUST_CONTAIN_5="WXTP"

# Port Vai Temperature
PARSE_DATATYPE_6="VAI_TEMPERATURE"
PARSE_ABSTRACT_INDEX_6=25
MUST_CONTAIN_6="WXTP"

# Port Vai Humidity
PARSE_DATATYPE_7="VAI_HUMIDITY"
PARSE_ABSTRACT_INDEX_7=26
MUST_CONTAIN_7="WXTP"

# Port Vai Pressure
PARSE_DATATYPE_8="VAI_PRESSURE"
PARSE_ABSTRACT_INDEX_8=27
MUST_CONTAIN_8="WXTP"

# Port Vai Rain Accumulation
PARSE_DATATYPE_9="VAI_RAIN_ACCUMULATION"
PARSE_ABSTRACT_INDEX_9=28
MUST_CONTAIN_9="WXTP"

# Port Vai Rain Intensity
PARSE_DATATYPE_10="VAI_RAIN_INTENSITY"
PARSE_ABSTRACT_INDEX_10=29
MUST_CONTAIN_10="WXTP"

#----------------------
# Stbd Vai Address
PARSE_DATATYPE_11="VAI_ADDRESS"
PARSE_ABSTRACT_INDEX_11=30
MUST_CONTAIN_11="WXTS"

# Stbd Vai Wind Direction
PARSE_DATATYPE_12="VAI_WIND_DIRECTION_AVG"
PARSE_ABSTRACT_INDEX_12=31
MUST_CONTAIN_12="WXTS"

# Stbd Wind Speed Min
PARSE_DATATYPE_13="VAI_WIND_SPEED_MIN"
PARSE_ABSTRACT_INDEX_13=32
MUST_CONTAIN_13="WXTS"

# Stbd Vai Wind Speed Avg
PARSE_DATATYPE_14="VAI_WIND_SPEED_AVG"
PARSE_ABSTRACT_INDEX_14=33
MUST_CONTAIN_14="WXTS"

# Stbd Vai Wind Speed Max
```

```
PARSE_DATATYPE_15="VAI_WIND_SPEED_MAX"
PARSE_ABSTRACT_INDEX_15=34
MUST_CONTAIN_15="WXTS"

# Stbd Vai Temperature
PARSE_DATATYPE_16="VAI_TEMPERATURE"
PARSE_ABSTRACT_INDEX_16=35
MUST_CONTAIN_16="WXTS"

# Stbd Vai Humidity
PARSE_DATATYPE_17="VAI_HUMIDITY"
PARSE_ABSTRACT_INDEX_17=36
MUST_CONTAIN_17="WXTS"

# Stbd Vai Pressure
PARSE_DATATYPE_18="VAI_PRESSURE"
PARSE_ABSTRACT_INDEX_18=37
MUST_CONTAIN_18="WXTS"

# Stbd Vai Rain Accumulation
PARSE_DATATYPE_19="VAI_RAIN_ACCUMULATION"
PARSE_ABSTRACT_INDEX_19=38
MUST_CONTAIN_19="WXTS"

# Stbd Vai Rain Intensity
PARSE_DATATYPE_20="VAI_RAIN_INTENSITY"
PARSE_ABSTRACT_INDEX_20=39
MUST_CONTAIN_20="WXTS"

#-----------------------------------------------
# $VDVBW
#-----------------------------------------------
[INPUT_5]
SOURCE_TYPE="UDP"
IN_SOCKET=55202

NUMBER_OF_CSV_PARSE=1

# to use to parse the data and the ABSTRACT_INDEX
PARSE_DATATYPE_1="VDVBW_GROUND_SPEED_AHEAD"
PARSE_ABSTRACT_INDEX_1=7

#-----------------------------------------------
# SBE45
#-----------------------------------------------
[INPUT_6]
SOURCE_TYPE="UDP"
IN_SOCKET=55501

NUMBER_OF_CSV_PARSE=4

# The minimum for each field is to specify what rule
# to use to parse the data and the ABSTRACT_INDEX
PARSE_DATATYPE_1="SBE45_TEMPERATURE"
PARSE_ABSTRACT_INDEX_1=10

PARSE_DATATYPE_2="SBE45_CONDUCTIVITY"
PARSE_ABSTRACT_INDEX_2=11

PARSE_DATATYPE_3="SBE45_SALINITY"
PARSE_ABSTRACT_INDEX_3=12

PARSE_DATATYPE_4="SBE45_SOUNDVELOCITY"
PARSE_ABSTRACT_INDEX_4=13

#-----------------------------------------------
# SBE48
#-----------------------------------------------
[INPUT_7]
SOURCE_TYPE="UDP"
IN_SOCKET=55502
```

```
NUMBER_OF_CSV_PARSE=1

# The minimum for each field is to specify what rule
# to use to parse the data and the ABSTRACT_INDEX
PARSE_DATATYPE_1="SBE48_TEMPERATURE"
PARSE_ABSTRACT_INDEX_1=6

#-----------------------------------------------
# LWR
#-----------------------------------------------
[INPUT_8]
SOURCE_TYPE="UDP"
IN_SOCKET=55404

NUMBER_OF_CSV_PARSE=1

# The minimum for each field is to specify what rule
PARSE_DATATYPE_1="LWR_RADIATION"
PARSE_ABSTRACT_INDEX_1=8

#-----------------------------------------------
# SWR
#-----------------------------------------------
[INPUT_9]
SOURCE_TYPE="UDP"
IN_SOCKET=55403

NUMBER_OF_CSV_PARSE=1

# The minimum for each field is to specify what rule
# to use to parse the data and the ABSTRACT_INDEX
PARSE_DATATYPE_1="SWR_RADIATION"
PARSE_ABSTRACT_INDEX_1=9

#-----------------------------------------------
# Knudsen Depth
#-----------------------------------------------
[INPUT_10]
SOURCE_TYPE="UDP"

#pick one set of input parameters - either 320B/R or 3260

#320
#IN_SOCKET=55603
#3260
IN_SOCKET=55604

NUMBER_OF_CSV_PARSE=2

# The minimum for each field is to specify what rule
# to use to parse the data and the ABSTRACT_INDEX
PARSE_DATATYPE_1="PKEL_DEPTH_1"

#Select this when 320B/R is in use: 12kHz
#PARSE_ABSTRACT_INDEX_1=16
#Select this when 3260 is in use: 12 kHz
PARSE_ABSTRACT_INDEX_1=17

PARSE_DATATYPE_2="PKEL_DEPTH_2"
#Select this when 320B/R is in use: 3.5kHz
#PARSE_ABSTRACT_INDEX_2=17
#Select this when 3260 is in use: 3.5kHz
PARSE_ABSTRACT_INDEX_2=16

#-----------------------------------------------
# FLR
#-----------------------------------------------
[INPUT_11]
SOURCE_TYPE="UDP"
IN_SOCKET=55503
```

```
NUMBER_OF_CSV_PARSE=1

PARSE_DATATYPE_1="FLR_MILLIVOLTS"
PARSE_ABSTRACT_INDEX_1=15

#-----------------------------------------------
# EM122 Depth
#-----------------------------------------------
[INPUT_12]
SOURCE_TYPE="UDP"
IN_SOCKET=55602

NUMBER_OF_CSV_PARSE=1

# The minimum for each field is to specify what rule
# to use to parse the data and the ABSTRACT_INDEX
PARSE_DATATYPE_1="KIDPT_DEPTH"
PARSE_ABSTRACT_INDEX_1=18

### Quanties Calculated by dsLogCsv ###

#-----------------------------------------------
# SSV
# SBE45S and SBE48T
[SSV_1]
TEMPERATURE_ABSTRACT_INDEX=6
SALINITY_ABSTRACT_INDEX=12

# SSV_2
# SBE45T and SBE45S
# I know it calculates its own, but curious how our calculations compare to the SBE45s
[SSV_2]
TEMPERATURE_ABSTRACT_INDEX=10
SALINITY_ABSTRACT_INDEX=12

#-----------------------------------------------
# PORT BAROMETER
[BAROMETER_1]
RAW_PRESSURE_ABSTRACT_INDEX=27
SENSOR_HEIGHT=20.6

# Specifies number of digits to right of decimal point (default is 3)
OUTPUT_PRECISION=2

#-----------------------------------------------
# STBD BAROMETER
[BAROMETER_2]
RAW_PRESSURE_ABSTRACT_INDEX=37
SENSOR_HEIGHT=20.6

# Specifies number of digits to right of decimal point (default is 3)
OUTPUT_PRECISION=2

#-----------------------------------------------
# Port TRUEWIND
[TRUEWIND_1]
WIND_DIR_ABSTRACT_INDEX=21
WIND_SPEED_ABSTRACT_INDEX=23
COG_ABSTRACT_INDEX=4
SOG_ABSTRACT_INDEX=5
HDG_ABSTRACT_INDEX=1

# Specifies number of digits to right of decimal point (default is 3)
OUTPUT_PRECISION=2

#-----------------------------------------------
# Stbd TRUEWIND
[TRUEWIND_2]
WIND_DIR_ABSTRACT_INDEX=31
WIND_SPEED_ABSTRACT_INDEX=33
```

```
COG_ABSTRACT_INDEX=4
SOG_ABSTRACT_INDEX=5
HDG_ABSTRACT_INDEX=1

# Specifies number of digits to right of decimal point (default is 3)
OUTPUT_PRECISION=2

#-----------------------------------------------
[OUTPUT_1]
OUTPUT_ENABLE=1

# specify the interval (in seconds) between output strings being created and output
OUTPUT_INTERVAL=60

# specify the string to use in data file names
DESTINATION_PREFIX="AT"

# specify that the first output is to a file
DESTINATION_1_TYPE="FILE"

# Specify that the output is UDP
# This would go to ftp as the entire string of data
DESTINATION_2_TYPE="UDP"
DESTINATION_2_IPADDRESS="192.147.41.60"
DESTINATION_2_SOCKET=55900

# For each output field specify the ABSTRACT_INDEX and
# a string to use for the header in data files

OUTFIELD_ABSTRACT_INDEX_1=2
OUTFIELD_HEADER_STRING_1="Dec_LAT"

OUTFIELD_ABSTRACT_INDEX_2=3
OUTFIELD_HEADER_STRING_2="Dec_LON"

OUTFIELD_ABSTRACT_INDEX_3=7
OUTFIELD_HEADER_STRING_3="SPD"

OUTFIELD_ABSTRACT_INDEX_4=1
OUTFIELD_HEADER_STRING_4="HDT"

OUTFIELD_ABSTRACT_INDEX_5=4
OUTFIELD_HEADER_STRING_5="COG"

FIELD_ABSTRACT_INDEX_6=5
OUTFIELD_HEADER_STRING_6="SOG"

OUTFIELD_ABSTRACT_INDEX_7=25
OUTFIELD_HEADER_STRING_7="WXTP_Ta"

OUTFIELD_ABSTRACT_INDEX_8=35
OUTFIELD_HEADER_STRING_8="WXTS_Ta"

OUTFIELD_ABSTRACT_INDEX_9=27
OUTFIELD_HEADER_STRING_9="WXTP_Pa"

OUTFIELD_ABSTRACT_INDEX_10=37
OUTFIELD_HEADER_STRING_10="WXTS_Pa"

OUTFIELD_ABSTRACT_INDEX_11=29
OUTFIELD_HEADER_STRING_11="WXTP_Ri"

OUTFIELD_ABSTRACT_INDEX_12=39
OUTFIELD_HEADER_STRING_12="WXTS_Ri"

OUTFIELD_ABSTRACT_INDEX_13=28
OUTFIELD_HEADER_STRING_13="WXTP_Rc"

OUTFIELD_ABSTRACT_INDEX_14=38
OUTFIELD_HEADER_STRING_14="WXTS_Rc"
```

```
OUTFIELD_ABSTRACT_INDEX_15=21
OUTFIELD_HEADER_STRING_15="WXTP_Dm"

OUTFIELD_ABSTRACT_INDEX_16=31
OUTFIELD_HEADER_STRING_16="WXTS_Dm"

OUTFIELD_ABSTRACT_INDEX_17=23
OUTFIELD_HEADER_STRING_17="WXTP_Sm"

OUTFIELD_ABSTRACT_INDEX_18=33
OUTFIELD_HEADER_STRING_18="WXTS_Sm"

OUTFIELD_ABSTRACT_INDEX_19=26
OUTFIELD_HEADER_STRING_19="WXTP_Ua"

OUTFIELD_ABSTRACT_INDEX_20=36
OUTFIELD_HEADER_STRING_20="WXTS_Ua"

OUTFIELD_TYPE_21="TRUEWIND_SPEED_1"
OUTFIELD_HEADER_STRING_21="WXTP_TS"

OUTFIELD_TYPE_22="TRUEWIND_SPEED_2"
OUTFIELD_HEADER_STRING_22="WXTS_TS"

OUTFIELD_TYPE_23="TRUEWIND_DIRECTION_1"
OUTFIELD_HEADER_STRING_23="WXTP_TD"

OUTFIELD_TYPE_24="TRUEWIND_DIRECTION_2"
OUTFIELD_HEADER_STRING_24="WXTS_TD"

OUTFIELD_ABSTRACT_INDEX_25=9
OUTFIELD_HEADER_STRING_25="SWR"

OUTFIELD_ABSTRACT_INDEX_26=8
OUTFIELD_HEADER_STRING_26="LWR"

OUTFIELD_ABSTRACT_INDEX_27=12
OUTFIELD_HEADER_STRING_27="SBE45S"

OUTFIELD_ABSTRACT_INDEX_28=6
OUTFIELD_HEADER_STRING_28="SBE48T"

OUTFIELD_TYPE_29="BAROMETER_1"
OUTFIELD_HEADER_STRING_29="BAROM_P"

OUTFIELD_TYPE_30="BAROMETER_2"
OUTFIELD_HEADER_STRING_30="BAROM_S"

OUTFIELD_ABSTRACT_INDEX_31=15
OUTFIELD_HEADER_STRING_31="FLR"

OUTFIELD_TYPE_32="SSV_1"
OUTFIELD_HEADER_STRING_32="SSVdslog"

OUTFIELD_ABSTRACT_INDEX_33=16
OUTFIELD_HEADER_STRING_33="Depth12"

OUTFIELD_ABSTRACT_INDEX_34=17
OUTFIELD_HEADER_STRING_34="Depth35"

OUTFIELD_ABSTRACT_INDEX_35=18
OUTFIELD_HEADER_STRING_35="EM122"
# specify the nuber of data fields in each output string
NUMBER_OF_OUTPUT_FIELDS=35

#------------------------------------------------------
# Compare SBE45, SBE48, and SSV (SBE45SSV and dsLogSSV
#------------------------------------------------------

[OUTPUT_2]
OUTPUT_ENABLE=1
```

```
# specify the interval (in seconds) between output strings being created and output
OUTPUT_INTERVAL=20

# specify the string to use in data file names
DESTINATION_PREFIX="AT_SSW20_"

# specify that the first output is to a file
DESTINATION_1_TYPE="FILE"

# specify the nuber of data fields in each output string
NUMBER_OF_OUTPUT_FIELDS=7

# For each output field specify the ABSTRACT_INDEX and
# a string to use for the header in data files

OUTFIELD_ABSTRACT_INDEX_1=6
OUTFIELD_HEADER_STRING_1="SBE48T"

OUTFIELD_ABSTRACT_INDEX_2=10
OUTFIELD_HEADER_STRING_2="SBE45T"

OUTFIELD_ABSTRACT_INDEX_3=11
OUTFIELD_HEADER_STRING_3="SBE45C"

OUTFIELD_ABSTRACT_INDEX_4=12
OUTFIELD_HEADER_STRING_4="SBE45S"

OUTFIELD_ABSTRACT_INDEX_5=13
OUTFIELD_HEADER_STRING_5="SSV_SBE45"

# SBE48T and SBE45S
OUTFIELD_TYPE_6="SSV_1"
OUTFIELD_HEADER_STRING_6="SSV_dslog"

# SBE45T and SBE45S
OUTFIELD_TYPE_7="SSV_2"
OUTFIELD_HEADER_STRING_7="SSV_sbe45_calc"

#-----------------------------------------------
#  Create an IMU csv file for plotting
#----------------------------------------------
[OUTPUT_3]
OUTPUT_ENABLE=1

# specify the interval (in seconds) between output strings being created and output
#OUTPUT_INTERVAL=1
OUTPUT_INTERVAL=10

# specify the string to use in data file names
DESTINATION_PREFIX="AT_IMU10_"

# specify that the first output is to a file
DESTINATION_1_TYPE="FILE"

# specify the number of data fields in each output string
NUMBER_OF_OUTPUT_FIELDS=12

# For each output field specify the ABSTRACT_INDEX and
# a string to use for the header in data files

OUTFIELD_ABSTRACT_INDEX_1=70
OUTFIELD_HEADER_STRING_1="PHINS_HEADING"

OUTFIELD_ABSTRACT_INDEX_2=1
OUTFIELD_HEADER_STRING_2="HDT"

OUTFIELD_ABSTRACT_INDEX_3=71
OUTFIELD_HEADER_STRING_3="PHTRO_ROLL"

OUTFIELD_ABSTRACT_INDEX_4=80
```

```
OUTFIELD_HEADER_STRING_4="PHTRO_ROLL_FRAME"

OUTFIELD_ABSTRACT_INDEX_5=72
OUTFIELD_HEADER_STRING_5="PHTRO_PITCH"

OUTFIELD_ABSTRACT_INDEX_6=79
OUTFIELD_HEADER_STRING_6="PHTRO_PITCH_FRAME"

OUTFIELD_ABSTRACT_INDEX_7=73
OUTFIELD_HEADER_STRING_7="PHLIN_SURGE_LIN"

OUTFIELD_ABSTRACT_INDEX_8=74
OUTFIELD_HEADER_STRING_8="PHLIN_SWAY_LIN"

OUTFIELD_ABSTRACT_INDEX_9=74
OUTFIELD_HEADER_STRING_9="PHLIN_HEAVE_LIN"

OUTFIELD_ABSTRACT_INDEX_10=76
OUTFIELD_HEADER_STRING_10="PHSPD_SURGE_SPD"

OUTFIELD_ABSTRACT_INDEX_11=77
OUTFIELD_HEADER_STRING_11="PHSPD_SWAY_SPD"

OUTFIELD_ABSTRACT_INDEX_12=78
OUTFIELD_HEADER_STRING_12="PHSPD_HEAVE_SPD"

#-------------------------------------------------
#  SSV from flow thru to UDP port for use with em122
#  format is space delimited:     SST   SSV
#  ftp will read this udp port and redirect the string out
#  to a serial port
#-----------------------------------------------

[OUTPUT_4]
OUTPUT_ENABLE=1

# specify the interval (in seconds) between output strings being created and output
OUTPUT_INTERVAL=1
TIMESTAMP_ENABLE=0
## options for string output
#OUTPUT_PREFIX_1="SMUDGE"
OUTPUT_DELIMITER=" "

#specify the nuber of data fields in each output string
NUMBER_OF_OUTPUT_FIELDS=2

# For parsed data specify the ABSTRACT_INDEX and
# a string to use for the header in data files
# For calculated values use OUTFIELD_TYPE
# Set HEADER_STRING to null to prevent UDP transmit of headers at points new files
# would be opened.   Just a note - I thought the header default is supposed to be null

# For EM122 wants Sea Temperature space SSV
# Using SBE48 and SBE48 needs to be set to output data 1Hz
# INTERVAL=0  use DS command to show current setup
#OUTFIELD_ABSTRACT_INDEX_1=6
#ABSTRACT INDEX VALUES -    6=SBE48T, 10=SBE45T, 13= SBE45V
OUTFIELD_HEADER_STRING_1=""
OUTFIELD_ABSTRACT_INDEX_1=10

# This is using SSV from SBE45
OUTFIELD_ABSTRACT_INDEX_2=13
OUTFIELD_AGE_LIMIT_2=60
OUTFIELD_HEADER_STRING_2=""

#specify that the output is UDP
DESTINATION_1_TYPE="UDP"
DESTINATION_1_IPADDRESS="199.92.161.3"
DESTINATION_1_SOCKET=55504

#------------------------------------------------------
```

```
# OUTPUT NOAA data
#------------------------------------------------------

[OUTPUT_5]
OUTPUT_ENABLE=1

OUTPUT_FILE_FORMAT="NOAA"

# DO NOT need to specify the interval (in seconds) between output strings
# this is done at the top of the hour and over writes noaa.tmp
OUTPUT_INTERVAL=60

# Do NOT need to specify the string to use in data file names
# DESTINATION_PREFIX="noaa_"

# specify ship call sign
SHIP_CALL_SIGN="KAQP"

# specify ship name
SHIP_NAME="ATLANTIS"

# specify VERSION number
NOAA_VERSION="SCS v4.0.0"

# specify that the first output is to a file
DESTINATION_1_TYPE="FILE"

# specify the nuber of data fields in each output string
NUMBER_OF_OUTPUT_FIELDS=8

# For each output field specify the ABSTRACT_INDEX and
# a string to use for the header in data files

OUTFIELD_ABSTRACT_INDEX_1=90
OUTFIELD_HEADER_STRING_1="ddmm.mmmh"

OUTFIELD_ABSTRACT_INDEX_2=91
OUTFIELD_HEADER_STRING_2="ddmm.mmmh"

OUTFIELD_TYPE_3="TRUEWIND_DIRECTION_1"
OUTFIELD_HEADER_STRING_3="deg"

OUTFIELD_TYPE_4="TRUEWIND_SPEED_1"
OUTFIELD_HEADER_STRING_4="m/s"

OUTFIELD_ABSTRACT_INDEX_5=25
OUTFIELD_HEADER_STRING_5="C"

OUTFIELD_ABSTRACT_INDEX_6=26
OUTFIELD_HEADER_STRING_6="%"

OUTFIELD_TYPE_7="BAROMETER_1"
OUTFIELD_HEADER_STRING_7="hPa"

OUTFIELD_ABSTRACT_INDEX_8=6
OUTFIELD_HEADER_STRING_8="C"

#-------------------------------------------------
#  Output 6 SAMOS file
#-------------------------------------------------

[OUTPUT_6]

# ENABLE OUTPUT
OUTPUT_ENABLE=1

# specify OUTPUT FORMAT for SAMOS
OUTPUT_FILE_FORMAT="SAMOS"

# specify ship call sign
SHIP_CALL_SIGN="KAQP"
```

```
#specify ship name
SHIP_NAME="ATLANTIS"

# specify output interval
OUTPUT_INTERVAL=60

# specify the string to use in data file names
DESTINATION_PREFIX="samos_"

# specify that the first output is to a file
DESTINATION_1_TYPE="FILE"

# specify the number of data fields in each output string
NUMBER_OF_OUTPUT_FIELDS=28

# For each output field specify the ABSTRACT_INDEX and
# a string to use for the header in data files

OUTFIELD_ABSTRACT_INDEX_1=2
OUTFIELD_HEADER_STRING_1="LA"

OUTFIELD_ABSTRACT_INDEX_2=3
OUTFIELD_HEADER_STRING_2="LO"

OUTFIELD_ABSTRACT_INDEX_3=5
OUTFIELD_HEADER_STRING_3="SOG"

OUTFIELD_ABSTRACT_INDEX_4=94
OUTFIELD_HEADER_STRING_4="COG"

OUTFIELD_ABSTRACT_INDEX_5=95
OUTFIELD_HEADER_STRING_5="GY"

OUTFIELD_ABSTRACT_INDEX_6=9
OUTFIELD_HEADER_STRING_6="SWR"

OUTFIELD_ABSTRACT_INDEX_7=8
OUTFIELD_HEADER_STRING_7="LWR"

OUTFIELD_ABSTRACT_INDEX_8=6
OUTFIELD_HEADER_STRING_8="SST"

OUTFIELD_ABSTRACT_INDEX_9=11
OUTFIELD_HEADER_STRING_9="SSC"

OUTFIELD_ABSTRACT_INDEX_10=12
OUTFIELD_HEADER_STRING_10="SAL"

OUTFIELD_TYPE_11="TRUEWIND_DIRECTION_2"
OUTFIELD_HEADER_STRING_11="WSTD"

OUTFIELD_TYPE_12="TRUEWIND_SPEED_2"
OUTFIELD_HEADER_STRING_12="WSTS"

OUTFIELD_ABSTRACT_INDEX_13=35
OUTFIELD_HEADER_STRING_13="WSAT"

OUTFIELD_TYPE_14="BAROMETER_2"
OUTFIELD_HEADER_STRING_14="WSBP"

OUTFIELD_ABSTRACT_INDEX_15=36
OUTFIELD_HEADER_STRING_15="WSRH"

OUTFIELD_ABSTRACT_INDEX_16=38
OUTFIELD_HEADER_STRING_16="WSRC"

OUTFIELD_ABSTRACT_INDEX_17=39
OUTFIELD_HEADER_STRING_17="WSRI"

OUTFIELD_ABSTRACT_INDEX_18=31
```

```
OUTFIELD_HEADER_STRING_18="WSRD"

OUTFIELD_ABSTRACT_INDEX_19=33
OUTFIELD_HEADER_STRING_19="WSRS"

OUTFIELD_TYPE_20="TRUEWIND_DIRECTION_1"
OUTFIELD_HEADER_STRING_20="WPTD"

OUTFIELD_TYPE_21="TRUEWIND_SPEED_1"
OUTFIELD_HEADER_STRING_21="WPTS"

OUTFIELD_ABSTRACT_INDEX_22=25
OUTFIELD_HEADER_STRING_22="WPAT"

OUTFIELD_TYPE_23="BAROMETER_1"
OUTFIELD_HEADER_STRING_23="WPBP"

OUTFIELD_ABSTRACT_INDEX_24=26
OUTFIELD_HEADER_STRING_24="WPRH"

OUTFIELD_ABSTRACT_INDEX_25=28
OUTFIELD_HEADER_STRING_25="WPRC"

OUTFIELD_ABSTRACT_INDEX_26=29
OUTFIELD_HEADER_STRING_26="WPRI"

OUTFIELD_ABSTRACT_INDEX_27=21
OUTFIELD_HEADER_STRING_27="WPRD"

OUTFIELD_ABSTRACT_INDEX_28=23
OUTFIELD_HEADER_STRING_28="WPRS"
```

Document History

- R000 – 2014-01-04
    - This “unreleased” version of documentation created during development to facilitate discussion of requirements.
- R001 – 2014-05-07
    - Continued unreleased progress.
- R002 – 2014-05-08
    - Continued unreleased progress.
- R006 – 2014-11-13
    - Update table of parsing rules to reflect changes in code.
    - Update document to reflect code changes to support NOAA and SAMOS output formats.
    - Update example sections.
- R007 – 2015-03-09
    - Document use of dsLogPostCSV.
    - Document NOAA and SAMOS output file formats.
    - Improve and extend documentation of calculated values.
    - Updated and reformatted all sample data and initialization file samples.
    - Corresponds to SVN Version 97 (2015-02-08)