SSSG dsLog User Guide

Tim Thiel - 2017-08-12 - Rev 006

Scope of document

The dsLog software is used to log and redistribute various, RS232, RS422, RS485, and UDP data streams. The operation of the dsLog software is controlled by specifying a user-generated initialization file. This document will provide the information needed to create an appropriate initialization file.

INI File Tutorial

A configuration file is used to control the behavior of dsLog. Excerpts from the Armstrong dsLog.ini file will be used to help describe the configuration file syntax. A complete copy of the Armstrong configuration file is included in the appendices.

The configuration file is composed of a sequence of sections which are delimited at the top of each section with a specific header surrounded by square brackets ***("[HEADER]").*** The configuration file must contain a [***GENERAL***] section, a section for each UDP or serial input channel ***([INPUT_1], [INPUT_2],*** etc.), a ***[GUI_THREAD]*** section, and a ***[TCP_SERVER_THREAD]*** section. The order in which the sections appear is irrelevant to the code but adhering to the SSSG standard of having GENERAL at the top, followed by the INPUT sections in numerical order, followed by the GUI_THREAD and TCP_SERVER_THREAD sections is strongly encouraged.

The basic operation of the dsLog program is that it starts a thread for each specified input. The threads listen for data on the specified input and when a complete data record is received it generates a time-stamped data record which is then written to one or several "destinations". The destinations can be either disk files or UDP sockets.

Configuration file syntax and notes

- Specified destination directories for disk logging must be created by the user. dsLog will NOT create these directories.
- 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.
- Use the '**#**' character to enter comments in the configuration file. Any line starting with the '**#**' will be ignored when the configuration file is processed.
- A convenient mechanism for commenting out an entire square-bracket delimited section is to replace the text within the square brackets with an unrecognized string. Remember that valid, consecutively-numbered input sections must still be defined. A contrived example is shown below:

```
# CNAV DATA
[*** DISABLED *** INPUT_2]
DATA_TYPE="NAV"
DATA_SOURCE="CNAV"
SOURCE_TYPE=2
```

[GENERAL] Section

Here is the [GENERAL] section from the Armstrong dsLog.ini:

```
[GENERAL]
ROOT_DIR="/underway/raw"
FILENAME_PREFIX="ar"
```

- ***ROOT_DIR*** – this entry controls where logged data is stored when it is directed to the local harddrive. Example syntax: ***ROOT_DIR="/underway/raw".*** It is important to remember that this directory must exist. It will NOT be created by the dsLog program.
- FILENAME_PREFIX – this entry controls the filenames generated by dsLog. In some applications the user will choose to use input specific or prompt specific prefixes (details later), but it is important to specify a global value. Example syntax: ***FILENAME_PREFIX="ar".*** The filenames for the logged data files take the form: ***"xxYYYYMMDD_hhmm.zzz***" where:
  - ***xx*** is the FILENAME_PREFIX.
  - ***YYYY*** is the year of the time the file was first written to.
  - ***MM*** is the month of the time the file was first written to.
  - ***DD*** is the day of the time the file was first written to.
  - ***hh*** is the hour of the time the file was first written to.
  - ***mm*** is the minute of the time the file was first written to.
  - ***zzz*** is the FILENAME_EXTENSION (defined later in this document).

[INPUT_N] Section

An input is typically either a serial port or UDP port. Each serial or UDP input has a corresponding ini file section labeled ***[INPUT_N],*** where 'N' is an integer. The inputs must be numbered consecutively beginning with 1. Each input section will source information and destination information. Some of the inputs will have information specifying prompts to be sent on the serial port, for example the Vaisala instrument. The source information specifies the information needed to receive data from the specific serial port or UDP socket. The destination information specifies where to send this information either on the local hard drive or on a UDP socket. Each input can have several destinations.

An example for a serial port input

For the first example we will use the CNAV data input from the Armstrong configuration file:

```
# CNAV DATA
[INPUT_2]
DATA_TYPE="NAV"
DATA_SOURCE="CNAV"
SOURCE_TYPE=2
SERIAL_PORT_NAME="/dev/ttyUSB1"
BAUD=9600
TERM_CHAR="\n"
FILENAME_PREFIX="ar"

DESTINATION_1=1
DESTINATION_2=0
DESTINATION_2_IPADDRESS="192.147.41.60"
DESTINATION_2_SOCKET=55001
DESTINATION_3=0
```

```
DESTINATION_3_IPADDRESS="127.0.0.1"
DESTINATION_3_SOCKET=55001
# this is for make_dat file from UDP
DESTINATION_4=0
DESTINATION_4_IPADDRESS="127.0.0.1"
DESTINATION_4_SOCKET=55007
# for mailhost need to change IP number
DESTINATION_5=0
DESTINATION_5_IPADDRESS="192.147.41.2"
DESTINATION_5_SOCKET=55001
DESTINATION_5_RAWMODE=1

USE_DESTINATION_SPECIFIC_FILE_EXTENSION=1
DESTINATION_1_EXTENSION="CNAV_3050"
```

- The CNAV is the second input section in the configuration file so the section header is ***[INPUT_2]***.
- The ***"SOURCE_TYPE"*** line is a critical piece of each ***[INPUT_X]*** section. It specifies whether the data source is a serial port or a UDP socket.
    - Set ***SOURCE_TYPE*** to 0 for a UDP socket.
    - Set ***SOURCE_TYPE*** to 2 for a serial port.
- ***DATA_TYPE*** and ***DATA_SOURCE*** are used to control fields in the logged data records. The ***DATA_TYPE*** string appears before the data record timestamp and the ***DATA_SOURCE*** string appears after the timestamp and before the actual logged data. The brief sample of data below illustrates how these fields are included in the logged data records. Some of the longer lines were truncated for clarity.

```
NAV 2016/02/06 10:00:00.280 CNAV $GNGGA,100000.00,3251.655617,N,07957.5679,…
NAV 2016/02/06 10:00:01.071 CNAV $GNZDA,100001.00,06,02,2016,00,00*79
NAV 2016/02/06 10:00:01.151 CNAV $GNRMC,100001.00,A,3251.655615,N,…
NAV 2016/02/06 10:00:01.190 CNAV $GNVTG,180.0,T,,M,0.01,N,0.01,K,D*1F
```

- Configuring serial inputs:
    - ***SERIAL_PORT_NAME*** must be specified. This is the absolute path to the /dev entry for the device serial port.
    - Use ***BAUD*** to specify the baud rate of the serial data on the port. In this example the data is at 9600 baud.
    - There are also handles to control ***PARITY, DATABITS, STOPBITS,*** and ***FLOWCONTROL.*** Default values are used for each of these fields in this example. The use of these entries is specified later in this document.
    - For serial port data sources dsLog looks for a termination character to signal that a complete data record to be logged has been received. The ***TERM_CHAR*** entry in the configuration file ***INPUT*** section controls which character represents the end-of-record. In this example it is the line-feed character. The methods for specifying ***TERM_CHAR*** are fully specified later in this document.
- Configuring Destinations:
    - For each input one or several logging destinations can be specified.
    - Configuration information for destination number "X' will have the substring "***_X_"*** between the "***DESTINATION"*** substring and the string identifying the controlled parameter.
    - The ***"DESTNATION_X"*** specifier controls whether that particular destination is a disk file or a UDP socket.

        - The value 0 indicates a UDP socket.
        - The value 1 indicates a disk file.
    - Disk file destinations:
        - Set ***DESTNATION_X=1***.
        - For disk files ***FILENAME_PREFIX*** controls the first part of the filename (preceding the timestamp). This value is not destination specific. It applies to all of the destinations associated with this input.
        - At this point each destination must have an associated filename extension (***DESTINATION_X_EXTENSION***) specified. For each input it is also necessary for the user to specify that ***"USE_DESTINATION_SPECIFIC_FILE_EXTENSION=1"*** or the destination specific extensions will be ignored. This will be addressed in an upcoming release.
    - UDP socket destinations:
        - Set ***DESTNATION_X=0***.
        - Set ***DESTINATION_X_IPADDRESS*** to the destination IP address. Note that this value is a string delimited by double-quotes.
        - Set ***DESTINATION_X_SOCKET*** to the destination UDP socket port number. Note that this value is an integer.

An example for a UDP socket input

The configuration of UDP based data sources is actually simpler than the serial case as with UDP sockets there is only an inward socket number to specify. The message boundaries do not need to be specified (a single UDP datagram is a single loggable data string. Here is an example configuration file entry:

```
[INPUT_20]
DATA_TYPE="NAV"
DATA_SOURCE="GYRO"

SOURCE_TYPE=0
IN_SOCKET=55110
### Define each of the different destinations
FILENAME_PREFIX="ar"

USE_DESTINATION_SPECIFIC_FILE_EXTENSION=1
DESTINATION_1=1
DESTINATION_1_EXTENSION="GYRO1_11424"
DESTINATION_2=0
DESTINATION_2_IPADDRESS="127.0.0.1"
DESTINATION_2_SOCKET=55105
```

Setting the ***SOURCE_TYPE*** to zero instructs dsLog to treat this input as UDP socket based. Then simply use the ***IN_SOCKET*** parameter to specify which port to bind the associated UDP socket to. The example shows the configuration of two destinations as well. These are similar to the previous example.

An example for a serial port input with transmitted prompt strings

Certain instruments transmit data only when prompted. There will be times when multiple instruments are on a single RS485 loop. The dsLog program provides facilities for generating and transmitting appropriate prompts for each of these situations. Additionally, the user can specify specific destinations and other handling for the responses to each of these prompts.

The configuration file segment for the 485 loop on the Armstrong is shown below. This section configures prompts for each of five different instruments, with two of them disabled by the syntax of the configuration file.

The configuration file controls both the generation and transmission of these prompts and the handling of the replies.

dsLog will transmit a sequence of prompts at a specified interval. To control the generation and transmission of these prompts the user must specify:

- The ASCII characters that make up each prompt string.
- The interval to wait between the transmissions of individual prompts in the prompt sequence.
- The interval to wait between transmitting the series of prompts.

Controlling the cadence of prompt transmissions

For example it is possible to specify a sequence of three prompts with five seconds between individual prompts and one minute between sequences of prompts. With this configuration three minutes of operation could result in prompts being transmitted at the following times:

- 00:00:03 – Transmit prompt 1
- 00:00:08 – Transmit prompt 2
- 00:00:13 – Transmit prompt 3
- 00:01:03 – Transmit prompt 1
- 00:01:08 – Transmit prompt 2
- 00:01:13 – Transmit prompt 3
- 00:02:03 – Transmit prompt 1
- 00:02:08 – Transmit prompt 2
- 00:02:13 – Transmit prompt 3

Note that when the first sequence of prompts starts is somewhat arbitrary and there is no reason to expect it to start at the top of the hour or at the top of the minute.

To configure the cadence of the prompt sequence use the following configuration file entries:

- ***PROMPT_SEQUENCE_NUMBER_OF_PROMPTS*** – The number of prompts in the sequence. Prompts are specified with consecutive integers starting with one.
- ***PROMPT_SEQUENCE_INTERVAL*** – The number of seconds that will elapse between the start of transmission for each SEQUENCE of prompts.
- ***PROMPT_SEQUENCE_INTERVAL_BETWEEN_PROMPTS*** – The number of seconds that will elapse between the start of transmission for each individual prompt in a sequence.

```
# time between sequences of prompts
PROMPT_SEQUENCE_INTERVAL=10

# time between prompts within the sequence
PROMPT_SEQUENCE_INTERVAL_BETWEEN_PROMPTS=1

# number of instruments to be prompted
PROMPT_SEQUENCE_NUMBER_OF_PROMPTS=3
```

In the example from the Armstrong five prompt strings are defined, but because ***PROMPT_SEQUENCE_NUMBER_OF_PROMPTS*** is set to three only the first three prompts will ever be transmitted.

Controlling the strings that are transmitted

To configure the string that will be transmitted for each prompt a series of entries is used to specify each individual character in the string. Hexadecimal notation is used to specify the individual characters. In the Armstrong example the ***PROMPT_1*** string is set to the string "<CR><LF>PR0<CR><LF>" (the number 0 follows the PR characters). The first <CR><LF> pair are sent to clear the Vaisala input handling.

- ***PROMPT_X_CHAR_Y*** specifies a single character of a prompt.
  - The ***X*** indicates which prompt string is being specified.
  - The ***Y*** indicates which character of the string is being specified.
  - So ***PROMPT_2_CHAR_5=33*** sets the fifth character of prompt number 2 to the character '3' (0x33).

```
# Port Vaisala  (PR0CRLF)
PROMPT_1_CHAR_1=0D
PROMPT_1_CHAR_2=0A
PROMPT_1_CHAR_3=50
PROMPT_1_CHAR_4=52
PROMPT_1_CHAR_5=30
PROMPT_1_CHAR_6=0D
PROMPT_1_CHAR_7=0A
```

Specifying PROMPT specific termination characters

There can be several different instruments on a single RS485 loop and it is possible (or likely) that they will not all use the same character to terminate their responses. Use the ***PROMPT_X_TERM_CHAR*** setting to specify a unique termination character for each prompt. The termination character signals to ***dsLog*** that a complete reply has been received and triggers the generation of a data record. For ***INPUT_7*** on the Armstrong the termination characters are specified by:

```
# Answers terms CHAR
USE_PROMPT_SPECIFIC_TERM_CHAR=1
PROMPT_1_TERM_CHAR=0A
PROMPT_2_TERM_CHAR=0A
PROMPT_3_TERM_CHAR=0D

#PROMPT_4_TERM_CHAR=0A
#PROMPT_5_TERM_CHAR=0A
```

This set the termination character for PROMPT #3 to <CR> (0x0D), and sets all the other prompt specific termination characters to <LF> (0x0A). Note that ***USE_PROMPT_SPECIFIC_TERMCHAR=1*** must be specified or the user-specified prompt specific termination characters will be ignored.

Specifying PROMPT specific destination info

It is likely that users will need to control which destinations receive data associated with specific prompts on the RS485 loop. For an input with defined prompts the mechanism to do this consists of specifying a common set of destinations and then using the ***PROMPT_X_DESTINATION_ENABLE_Y*** setting to control which destinations receive data associated with each of the various prompts. To use this feature the user must first set "***USE_PROMPT_SPECIFIC_DESTINATIONS_FLAG=1***" for the input. If ***USE_PROMPT_SPECIFIC_DESTINATIONS_FLAG*** is unspecified or set to one then all replies on this input will be sent to ALL of the defined destinations for the input.

Then for each prompt the user must specify which destinations should receive the data associated with that prompt. All of the prompt specific destination enable flags default to zero. So the user only needs to set the flags that should be one. Use the syntax:

- ***PROMPT_X_DESTINATION_ENABLE_Y=1***
- This enables DESTINATION #Y for PROMPT #X.
-

```
# Enable destinations for each prompt
USE_PROMPT_SPECIFIC_DESTINATIONS_FLAG=1
# PROMPT 1 Port Vaisala
PROMPT_1_DESTINATION_ENABLE_1=1
PROMPT_1_DESTINATION_ENABLE_2=1
PROMPT_1_DESTINATION_ENABLE_3=1
# PROMPT 2 STBD Vaisala
PROMPT_2_DESTINATION_ENABLE_4=1
PROMPT_2_DESTINATION_ENABLE_5=1
PROMPT_2_DESTINATION_ENABLE_6=1
#PROMPT 3 FLR
PROMPT_3_DESTINATION_ENABLE_7=1
PROMPT_3_DESTINATION_ENABLE_8=1
PROMPT_3_DESTINATION_ENABLE_9=1

# PROMPT 4 SWR
#PROMPT_4_DESTINATION_ENABLE_10=1
#PROMPT_4_DESTINATION_ENABLE_11=1
#PROMPT_4_DESTINATION_ENABLE_12=1
## PROMPT 5 LWR
#PROMPT_5_DESTINATION_ENABLE_13=1
#PROMPT_5_DESTINATION_ENABLE_14=1
#PROMPT_5_DESTINATION_ENABLE_15=1
```

Specifying PROMPT specific source names

As each logfile entry is generated it contains a ***SOURCE_NAME***. This ***SOURCE_NAME*** string appears immediately following the time string and immediately before the actual logged data. It is possible to specify unique ***SOURCE_NAME*** strings for each prompt.

To use this feature the user must first set "***USE_PROMPT_SPECIFIC_SOURCE_NAME=1***" for the input. Then for each prompt the user must specify the source name to be associated with that prompt. All of the prompt specific source names default to the source name that was specified for the input. So the user only needs to set the source names that need to be unique. Use the syntax:

```
USE_PROMPT_SPECIFIC_SOURCE_NAME=1
PROMPT_1_SOURCE_NAME="WXTP"
PROMPT_2_SOURCE_NAME="WXTS"
PROMPT_3_SOURCE_NAME="FLR"
#PROMPT_4_SOURCE_NAME="SWR"
#PROMPT_5_SOURCE_NAME="LWR"
```

Specifying PROMPT specific data types

As each logfile entry is generated it contains a ***DATA_TYPE***. This ***DATA_TYPE*** string appears at the start of the logged data record immediately before the time string. It is possible to specify unique ***DATA_TYPE*** strings for each prompt.

To use this feature the user must first set "***USE_PROMPT_SPECIFIC_DATA_TYPE=1***" for the input. Then for each prompt the user must specify the data type to be associated with that prompt. All of the prompt specific data types default to the data type that was specified for the input. So the user only needs to set the data types that need to be unique. Use the syntax:

```
USE_PROMPT_SPECIFIC_DATA_TYPE=1
PROMPT_1_DATA_TYPE="MET"
PROMPT_2_DATA_TYPE="MET"
PROMPT_3_DATA_TYPE="SSW"
#PROMPT_4_DATA_TYPE="MET"
#PROMPT_5_DATA_TYPE="MET"
```

dsLog Status Messages

The dsLog process creates a file with information about the status of the data logging process. This file has a suffix of LGS and is placed in the root logging directory.

LGS messages are added to the file regularly to give basic statistics about each of the input threads. The syntax for these messages is:

LGS timestamp LGR INPUT_1_INFO INPUT_2_INFO … INPUT_N_INFO

For each defined data Input there is a section of data defined as:

Sn TimeSinceLastData NumberOfMessagesReceived NumberOfBytesReceived NumberOfReadErrors

Here is an example:

LGS 2013/01/11 10:46:28.190 LGR S1 1.97 52 2460 0 S2 4.32 3 69 0

In this example, for the [INPUT_1] stream it has been 1.97 seconds since data has been received; 52 messages have been received; 2460 bytes of data have been received; and there have been zero read errors. For the [INPUT_2] stream it has been 4.32 seconds since data has been received; 3 messages have been received; 69 bytes of data have been received; and there have been zero read errors.

DSR messages are added to the file which give information on the disk space available for each disk based destination. The syntax for these messages is:

DSR timestamp LGR INPUT_1_INFO INPUT_2_INFO … INPUT_N_INFO

For each defined data Input there is a section of data defined as:

Sn numberOfDestinations DEST_1_INFO DEST_2_INFO … DEST_N_INFO

For each destination there will be three associated fields. If the destination is a disk type destination then the fields will be:

- Error_Count – at this time this is unimplemented and will always be zero.
- Percentage of free space remaining
- Remaining space in bytes

If the destination is not a disk type destination then the three associated fields will be "NOT_DISK 0.0 0".

Here is an example:

DSR 2013/01/11 10:46:48.189 LGR S1 4 0 92.6 1362468093952 NOT_DISK 0.0 0 NOT_DISK 0.0 0 NOT_DISK 0.0 0 NOT_DISK 0.0 0 S2 2 0 92.6 1362468093952 NOT_DISK 0.0 0

The 'S1' field starts the output related to the '[INPUT_1]' section. The next field indicates that there are four defined destinations for that input. The first defined INPUT_1 destination is a disk so there are fields indicating the error count, percentage of space remaining, and amount of available space. The remaining three destinations associated with INPUT_1 are UDP type, so they each show the NOT_DISK field and zeroes. Finally there are similar outputs for the two destinations associated with '[INPUT_2]' in the fields following the 'S2' string.

Initialization File Sections

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 sections are enumerated below with brief descriptions:

- [GENERAL]
- [INPUT_n]
- [GUI_THREAD]
- [TCP_SERVER_THREAD]

[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.
- **READ_FROM_INI**
  - Type – int
  - Stored as – Not applicable
  - Valid values – Not applicable
  - Description – Apparently an unimplemented function. Appears in template ini files but not in source. Will check with Jon Howland about intentions.
- **INI_SERVER_IP_ADDRESS**
  - Type – string
  - Stored as – Not applicable
  - Valid values – Not applicable
  - Description – Apparently an unimplemented function. Appears in template ini files but not in source. Will check with Jon Howland about intentions.
- **INI_SERVER_SOCKET**
  - Type – int
  - Stored as – Not applicable
  - Valid values – Not applicable

- Description – Apparently an unimplemented function. Appears in template ini files but not in source. Will check with Jon Howland about intentions. [INPUT_x] section

[INPUT_n] section

- **DATA_TYPE**
  - Type – string
  - Stored as – loggingList[x].dataType
  - Valid values – Any unique three character string.
  - Description – Typically this is a three character string. It is appended to the ASCII log file names after a dot ('.') character. For each received string the corresponding log file entry will contain the dataType string as the leftmost field.
- **DATA_SOURCE**
  - Type – string
  - Stored as – loggingList[x].sourceName
  - Valid values – An ASCII string.
  - Description – For each received string the corresponding log file entry will contain the sourceName string after the timestamp and before the received data.
- **SOURCE_TYPE**
  - Type – int
  - Stored as – loggingList[x].sourceType
  - Valid values – 0 (UDP_ASCII), 1 (UDP_BINARY), 2 (SERIAL_ASCII), 3 (SERIAL_BINARY)
  - Description – Used to specify whether data is received via a serial port or a UDP connection and whether the data is ASCII or binary.
- **TIMESTAMP**
  - Type – int
  - Stored as – loggingList[x].timeStampFlag
  - Valid values – 0, 1
  - Description – Specifies whether logged data will have a timestamp included in the records.
- **IN_SOCKET**
  - Type – int
  - Stored as – loggingList[x].inSocketNumber
  - Valid values – Any valid UDP socket number.
  - Description – Specifies the UDP socket upon which the UDP data will be received.
- **SERIAL_PORT_NAME**
  - Type – string
  - Stored as – loggingList[x].theSerialPort.sio_com_port_name
  - Valid values – A valid serial port on the host machine. (e.g. "/dev/ttyS0", "/dev/ttyS1").
  - Description – Specifies the serial port to use to recive logged data and (if specified) send queries.
- **PARITY**
  - Type – int
  - Stored as – loggingList[x]. theSerialPort.parity
  - Valid values – 0 (NONE); 1 (ODD); 2 (EVEN).
  - Description – Specifies the parity setting for the associated serial port.
- **BAUD**

- Type – int
- Stored as – loggingList[x]. theSerialPort.baud
- Valid values – Any baud rate supported by the hardware serial port and system software.
- Description – Specifies the baud rate for the associated serial port.

- **DATA_BITS**
  - Type – int
  - Stored as – loggingList[x]. theSerialPort.data_bits
  - Valid values – Any number of data bits supported by the hardware serial port and system software. Typically 8 which is the default value.
  - Description – Specifies the number of data bits for the associated serial port.
- **STOP_BITS**
  - Type – int
  - Stored as – loggingList[x]. theSerialPort.stop_bits
  - Valid values – Any number of stop bits supported by the hardware serial port and system software.
  - Description – Specifies the number of stop bits for the associated serial port.
- **ECHO**
  - Type – int (Boolean)
  - Stored as – loggingList[x]. theSerialPort.echo
  - Valid values – 0 and 1.
  - Description – If set to a non-zero value then received characters will be echoed out on the associated serial port.
- **FLOWCONTROL**
  - Type – int
  - Stored as – loggingList[x]. theSerialPort.flow_control
  - Valid values – 0 (NONE); 1 (XONXOFF); 2 (RTSCTS); 3 (XONXOFF_RTSCTS)
  - Description – Specifies the flow_control setting for the associated serial port.
- **TERM_CHAR**
  - Type – string
  - Stored as – loggingList[x]. theSerialPort.auto_mux_enabled
  - Stored as – loggingList[x]. theSerialPort.auto_mux_term_char
  - Valid values – “none”; “\r”; “\n”; the hexadecimal representation of a single character (“0xnn”); a single character enclosed in double-quotes.
  - Description – the auto_mux_enabled value is set to true unless “none” is specified. The auto_mux_term_char is set the specified ASCII character. The default value for auto_mux_term_char is ‘\r’.
- **PROMPT_SEQUENCE_INTERVAL**
  - Type – double
  - Stored as – loggingList[x].promptSequence.secondsBetweenSequences
  - Valid values –
  - Description – This value specifies the number of seconds that will elapse between sequences of specified prompt transmissions.
- **PROMPT_SEQUENCE_INTERVAL_BETWEEN_PROMPTS**
  - Type – double
  - Stored as – loggingList[x]. promptSequence.secondsBetweenPrompts

- Valid values –
- Description – This value specifies the number of seconds that will elapse between individual prompts in a prompt sequence, unless 'USE_PROMPT_SPECIFIC_REPLY_WINDOWS' is defined and set to 1.

- **PROMPT_SEQUENCE_NUMBER_OF_PROMPTS**
  - Type – int
  - Stored as – loggingList[x]. promptSequence.numberOfPrompts
  - Valid values –
  - Description – This value specifies the number of user specified prompts contained in the prompt sequence for this input.
- **PROMPT_n_CHAR_xx**
  - Type – Hexadecimal
  - Stored as – loggingList[x].promptSequence.prompts[n].promptChars[xx]
  - Valid values –
  - Description – Using hexadecimal, this value specifies the xx-th character of the n-th prompt in the prompt sequence. For example if the fourth character of the second prompt in the sequence is the digit 6 then the following line would be used:
    - PROMPT_2_CHAR_4=36
- **USE_PROMPT_SPECIFIC_TERM_CHAR**
  - Type – int
  - Stored as – loggingList[x].usePromptSpecificTermChar
  - Valid values – 0 or 1.
  - Description – If set to 0 then incoming characters will be written to the logger destinations every time the character specified by the "TERM_CHAR" is received. If set to one then the user can specify a specific term_char to be associated with each prompt string. This allows the use of devices with different response syntaxes to be used on a single input channel.
- **PROMPT_n_TERM_CHAR**
  - Type – Hexadecimal
  - Stored as – loggingList[x].promptTermChar[n]
  - Valid values – Any ASCII character.
  - Description – Ignored unless USE_PROMPT_SPECIFIC_TERM_CHAR is set to 1. After the nth prompt is sent on the interface the logger will buffer characters until the specified term_char is received. The reception of this term_char will trigger the logging of a data record.
- **USE_PROMPT_SPECIFIC_DATA_TYPE**
  - Type – int
  - Stored as – loggingList[x]. usePromptSpecificDataType
  - Valid values – 0 or 1.
  - Description – If set to 0 then incoming records will be written to the logger destinations with the INPUT_x 'DATA_TYPE'. If set to one then the user can specify a specific DATA_TYPE to be associated with each prompt string.
- **PROMPT_n_DATA_TYPE**
  - Type – string
  - Stored as – loggingList[x].promptDataType[n]
  - Valid values – Any unique three character string.

- Description – Typically this is a three character string. If USE_PROMPT_SPECIFIC_DATA_TYPE is unspecified or set to zero each log file entry for data on this input will contain the 'DATA_TYPE' string as the leftmost field. If set to one then the data records received in reply to prompt n will contain the 'PROMPT_n_DATA_TYPE' string as the leftmost field.

- **USE_PROMPT_SPECIFIC_SOURCE_NAME**
  - Type – int
  - Stored as – loggingList[x]. usePromptSpecificSourceName
  - Valid values – 0 or 1.
  - Description – If set to 0 then incoming records will be written to the logger destinations with the INPUT_x 'SOURCE_NAME'. If set to one then the user can specify a specific SOURCE_NAME to be associated with each prompt string.
- **PROMPT_n_SOURCE_NAME (change to DATA_SOURCE???)**
  - Type – string
  - Stored as – loggingList[x].promptSourceName[n]
  - Valid values – Any unique string.
  - Description –If USE_PROMPT_SPECIFIC_SOURCE_NAME is unspecified or set to zero each log file entry for data on this input will contain the 'DATA_SOURCE' string as the field after the timestamp and before the received data. If set to one then the data records received in reply to prompt n will contain the 'PROMPT_n_SOURCE_NAME' string as the field after the timestamp and before the received data.
- **USE_PROMPT_SPECIFIC_SYNCOPATION**
  - Type – int
  - Stored as – loggingList[x]. usePromptSpecificSyncopation
  - Valid values – 0 or 1.
  - Description – If unspecified or set to 0 then all of the prompts in the sequence will be sent each time the prompt sequence is triggered. If set to one then the user must specify the associated syncopation for each prompt in the sequence.
- **PROMPT_n_SYNCOPATION**
  - Type – int
  - Stored as – loggingList[x].promptSyncopation[n]
  - Valid values – Any integer.
  - Description – Ignored unless USE_PROMPT_SPECIFIC_SYNCOPATION is set to 1. This setting allows the user to specify that certain prompts in the sequence will not be transmitted every time the prompt sequence is triggered. The transmission of the associated prompt will be skipped PROMPT_N_SYNCOPATION times after each time the prompt is actually sent. So to send the prompt every cycle set PROMPT_N_SYNCOPATION to zero; to send it every other cycle set it to one; to send it every tenth cycle set PROMPT_N_SYNCOPATION to 9.
- **USE_PROMPT_SPECIFIC_REPLY_WINDOWS**
  - Type – int
  - Stored as – loggingList[x]. usePromptSpecificReplyWindows
  - Valid values – 0 or 1.
  - Description – If unspecified or set to 0 then the prompts in the sequence will be sent at a fixed cadence specified by 'PROMPT_SEQUENCE_INTERVAL_BETWEEN_PROMPTS'. If set to one then the user must specify the associated reply_window for each prompt in the sequence.
- **PROMPT_n_REPLY_WINDOW**

- Type – int
- Stored as – loggingList[x].promptReplyWindows[n]
- Valid values – Any integer.
- Description – If USE_PROMPT_SPECIFIC_REPLY_WINDOWS is unspecified or set to zero then the individual prompts in the prompt sequence will be sent at a fixed cadence specified by 'PROMPT_SEQUENCE_INTERVAL_BETWEEN_PROMPTS'. If set to one then the n+1th prompt will be sent PROMPT_n_REPLY_WINDOW seconds after the nth prompt.

- **USE_PROMPT_SPECIFIC_DESTINATIONS_FLAG**
  - Type – int
  - Stored as – loggingList[x]. usePromptSpecificDestinationsFlag
  - Valid values – 0 or 1.
  - Description – Specifies whether the logged data on this channel will be sent to all of the associated destinations or if the user can specify destinations to be associated with each prompt.
- **PROMPT_n_DESTINATION_ENABLE_nn**
  - Type – int
  - Stored as – loggingList[x].promptDestinationEnable[n][nn]
  - Valid values – 0 or 1.
  - Description – If USE_PROMPT_SPECIFIC_DESTINATIONS_FLAG is unspecified or set to zero then all data records on this input will be sent to all of the defined destinations for this input. If set to one then the data records received in reply to prompt n will be sent only to the destinations with the PROMPT_n_DESTINATION_EANBLE_nn set to one.
- **DESTINATION_n**
  - Type – int
  - Stored as – loggingList[x].destinations[n].destinationType
  - Valid values – 0 (UDP_SOCKET); 1 (DISK)
  - Description – Specifies the type of the logging destination (disk vs. network).
- **DESTINATION_n_IPADDRESS**
  - Type – string
  - Stored as – loggingList[x]. 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 – loggingList[x]. 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_SOCKET_STARTS_WITH**
  - Type – string
  - Stored as – loggingList[x]. destinations[n].destinationFilter.startsWith
  - Valid values –
  - Description -

- **DESTINATION_n_SOCKET_STARTS_CONTAINS**
    - Type – string
    - Stored as – loggingList[x]. destinations[n]. destinationFilter.greps
    - Valid values –
    - Description -
- **DESTINATION_n_SOCKET_STARTS_DOESNT_CONTAIN**
    - Type – string
    - Stored as – loggingList[x]. destinations[n]. destinationFilter.grepMinusVs
    - Valid values –
    - Description -
- **DESTINATION_n _PATHNAME**
    - Type – string
    - Stored as – loggingList[x]. 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.

[GUI_THREAD] section (TBD)

- **PORT**
    - Type – int
    - Stored as –
    - Valid values –
    - Description -
- **IPADDRESS**
- **TO_IP_ADDRESS**
- **TO_PORT**
- **DESTINATION_N**
    - Description -

[TCP_SERVER_THREAD] section (TBD)

- **TCP_SERVER_THREAD_SOCKET**
    - Type – int
    - Stored as –
    - Valid values –
    - Description –

Appendix – Configuration file section for a serial port input with transmitted prompt strings

```
## RS485
[INPUT_7]
## DISABLED PROMPT 4 AND 5, REPLACED WITH RAD SENSOR
# These are needed
DATA_TYPE="MET"
DATA_SOURCE="RS485"
##

FILENAME_PREFIX="ar"

SOURCE_TYPE=2
SERIAL_PORT_NAME="/dev/ttyUSB6"
BAUD=9600

# time between sequences of prompts
PROMPT_SEQUENCE_INTERVAL=10

# time between prompts within the sequence
PROMPT_SEQUENCE_INTERVAL_BETWEEN_PROMPTS=1

# number of instruments to be prompted
PROMPT_SEQUENCE_NUMBER_OF_PROMPTS=3

# Port Vaisala  (PR0CRLF)
PROMPT_1_CHAR_1=0D
PROMPT_1_CHAR_2=0A
PROMPT_1_CHAR_3=50
PROMPT_1_CHAR_4=52
PROMPT_1_CHAR_5=30
PROMPT_1_CHAR_6=0D
PROMPT_1_CHAR_7=0A

# Starboard Vaisala (SR0CRLF)
PROMPT_2_CHAR_1=0D
PROMPT_2_CHAR_2=0A
PROMPT_2_CHAR_3=53
PROMPT_2_CHAR_4=52
PROMPT_2_CHAR_5=30
PROMPT_2_CHAR_6=0D
PROMPT_2_CHAR_7=0A


# SSW Fluorometer ($3RDCR)
PROMPT_3_CHAR_1=24
PROMPT_3_CHAR_2=33
PROMPT_3_CHAR_3=52
PROMPT_3_CHAR_4=44
PROMPT_3_CHAR_5=0D

## ASMET Shortwave Radiation (##SWR01C)
#PROMPT_4_CHAR_1=23
#PROMPT_4_CHAR_2=23
#PROMPT_4_CHAR_3=53
#PROMPT_4_CHAR_4=57
#PROMPT_4_CHAR_5=52
#PROMPT_4_CHAR_6=30
#PROMPT_4_CHAR_7=31
#PROMPT_4_CHAR_8=43
```

```
## ASMET Longwave Radiation (##LWR01C)
#PROMPT_5_CHAR_1=23
#PROMPT_5_CHAR_2=23
#PROMPT_5_CHAR_3=4C
#PROMPT_5_CHAR_4=57
#PROMPT_5_CHAR_5=52
#PROMPT_5_CHAR_6=30
#PROMPT_5_CHAR_7=31
#PROMPT_5_CHAR_8=43

### Define each of the different destinations

USE_DESTINATION_SPECIFIC_FILE_EXTENSION=1

## Port Vaisala destination options
# File
DESTINATION_1=1
DESTINATION_1_EXTENSION="XTP"
#UDP
DESTINATION_2=0
DESTINATION_2_IPADDRESS="192.147.41.60"
DESTINATION_2_SOCKET=55401
#UDP to CSV file
DESTINATION_3=0
DESTINATION_3_IPADDRESS="127.0.0.1"
DESTINATION_3_SOCKET=55400

## STB Vaisala destination options
# File
DESTINATION_4=1
DESTINATION_4_EXTENSION="XTS"
#UDP
DESTINATION_5=0
DESTINATION_5_IPADDRESS="192.147.41.60"
DESTINATION_5_SOCKET=55402
DESTINATION_6=0
DESTINATION_6_IPADDRESS="127.0.0.1"
DESTINATION_6_SOCKET=55400

## FLR destination options
# File
DESTINATION_7=1
DESTINATION_7_EXTENSION="FLR"
# UDP
DESTINATION_8=0
DESTINATION_8_IPADDRESS="192.147.41.60"
DESTINATION_8_SOCKET=55503
DESTINATION_9=0
DESTINATION_9_IPADDRESS="127.0.0.1"
DESTINATION_9_SOCKET=55503

### These have been replaced by RAD, but they maybe needed again
### since they are around lstolp 20170324
### SWR destination options
##Destination_10=1
##DESTINATION_10_EXTENSION="SWR"
## UDP
##DESTINATION_11=0
#DESTINATION_11_IPADDRESS="10.10.10.1"
#DESTINATION_11_SOCKET=55403
#DESTINATION_12=0
```

```
#DESTINATION_12_IPADDRESS="127.0.0.1"
#DESTINATION_12_SOCKET=55403

## LWR destination options
#Destination_13=1
#DESTINATION_13_EXTENSION="LWR"
## UDP
#DESTINATION_14=0
#DESTINATION_14_IPADDRESS="10.10.10.1"
#DESTINATION_14_SOCKET=55404
#DESTINATION_15=0
#DESTINATION_15_IPADDRESS="127.0.0.1"
#DESTINATION_15_SOCKET=55404

# Enable destinations for each prompt
USE_PROMPT_SPECIFIC_DESTINATIONS_FLAG=1
# PROMPT 1 Port Vaisala
PROMPT_1_DESTINATION_ENABLE_1=1
PROMPT_1_DESTINATION_ENABLE_2=1
PROMPT_1_DESTINATION_ENABLE_3=1
# PROMPT 2 STBD Vaisala
PROMPT_2_DESTINATION_ENABLE_4=1
PROMPT_2_DESTINATION_ENABLE_5=1
PROMPT_2_DESTINATION_ENABLE_6=1
#PROMPT 3 FLR
PROMPT_3_DESTINATION_ENABLE_7=1
PROMPT_3_DESTINATION_ENABLE_8=1
PROMPT_3_DESTINATION_ENABLE_9=1

# PROMPT 4 SWR
#PROMPT_4_DESTINATION_ENABLE_10=1
#PROMPT_4_DESTINATION_ENABLE_11=1
#PROMPT_4_DESTINATION_ENABLE_12=1
## PROMPT 5 LWR
#PROMPT_5_DESTINATION_ENABLE_13=1
#PROMPT_5_DESTINATION_ENABLE_14=1
#PROMPT_5_DESTINATION_ENABLE_15=1

# Answers terms CHAR
USE_PROMPT_SPECIFIC_TERM_CHAR=1
PROMPT_1_TERM_CHAR=0A
PROMPT_2_TERM_CHAR=0A
PROMPT_3_TERM_CHAR=0D

#PROMPT_4_TERM_CHAR=0A
#PROMPT_5_TERM_CHAR=0A

USE_PROMPT_SPECIFIC_DATA_TYPE=1
PROMPT_1_DATA_TYPE="MET"
PROMPT_2_DATA_TYPE="MET"
PROMPT_3_DATA_TYPE="SSW"
#PROMPT_4_DATA_TYPE="MET"
#PROMPT_5_DATA_TYPE="MET"

USE_PROMPT_SPECIFIC_SOURCE_NAME=1
PROMPT_1_SOURCE_NAME="WXTP"
PROMPT_2_SOURCE_NAME="WXTS"
PROMPT_3_SOURCE_NAME="FLR"
#PROMPT_4_SOURCE_NAME="SWR"
#PROMPT_5_SOURCE_NAME="LWR"
```

Appendix – Complete R/V Armstrong Configuration file (2017-08-14)

```
[GENERAL]
ROOT_DIR="/underway/raw"
FILENAME_PREFIX="ar"

# Science GPS data possible switched source
[INPUT_1]
DATA_TYPE="NAV"
DATA_SOURCE="SWITCHED"
SOURCE_TYPE=2
SERIAL_PORT_NAME="/dev/ttyUSB0"
BAUD=9600
TERM_CHAR="\n"
FILENAME_PREFIX="ar"

DESTINATION_1=1
DESTINATION_2=0
#DESTINATION_2_IPADDRESS="10.10.10.1"
#DESTINATION_2_SOCKET=55006
#DESTINATION_3=0
#DESTINATION_3_IPADDRESS="127.0.0.1"
#DESTINATION_3_SOCKET=55001
#DESTINATION_2=0
#DESTINATION_2_IPADDRESS="192.147.41.60"
#DESTINATION_2_SOCKET=55001

USE_DESTINATION_SPECIFIC_FILE_EXTENSION=1
DESTINATION_1_EXTENSION="SCI_GPS"

# CNAV DATA
[INPUT_2]
DATA_TYPE="NAV"
DATA_SOURCE="CNAV"
SOURCE_TYPE=2
SERIAL_PORT_NAME="/dev/ttyUSB1"
BAUD=9600
TERM_CHAR="\n"
FILENAME_PREFIX="ar"

DESTINATION_1=1
DESTINATION_2=0
DESTINATION_2_IPADDRESS="192.147.41.60"
DESTINATION_2_SOCKET=55001
DESTINATION_3=0
DESTINATION_3_IPADDRESS="127.0.0.1"
DESTINATION_3_SOCKET=55001
# this is for make_dat file from UDP
DESTINATION_4=0
DESTINATION_4_IPADDRESS="127.0.0.1"
DESTINATION_4_SOCKET=55007
# for mailhost need to change IP number
DESTINATION_5=0
DESTINATION_5_IPADDRESS="192.147.41.2"
DESTINATION_5_SOCKET=55001
DESTINATION_5_RAWMODE=1

USE_DESTINATION_SPECIFIC_FILE_EXTENSION=1
DESTINATION_1_EXTENSION="CNAV_3050"

# Kongsberg DPS112 Nav
```

```
[INPUT_3]
DATA_TYPE="NAV"
DATA_SOURCE="DPS112"
SOURCE_TYPE=2
SERIAL_PORT_NAME="/dev/ttyUSB2"
BAUD=9600
TERM_CHAR="\n"
FILENAME_PREFIX="ar"

DESTINATION_1=1
DESTINATION_2=0
DESTINATION_2_IPADDRESS="127.0.0.1"
DESTINATION_2_SOCKET=55006
DESTINATION_3=0
DESTINATION_3_IPADDRESS="192.147.41.60"
DESTINATION_3_SOCKET=55006

USE_DESTINATION_SPECIFIC_FILE_EXTENSION=1
DESTINATION_1_EXTENSION="DPS_112"

## HDT
[INPUT_4]
DATA_TYPE="NAV"
DATA_SOURCE="GYRO"
SOURCE_TYPE=2
SERIAL_PORT_NAME="/dev/ttyUSB3"
BAUD=4800
TERM_CHAR="\n"
FILENAME_PREFIX="ar"

DESTINATION_1=1
DESTINATION_2=0
DESTINATION_2_IPADDRESS="192.147.41.60"
DESTINATION_2_SOCKET=55103
DESTINATION_3=0
DESTINATION_3_IPADDRESS="127.0.0.1"
DESTINATION_3_SOCKET=55103
DESTINATION_4=0
DESTINATION_4_IPADDRESS="192.147.41.2"
DESTINATION_4_SOCKET=55103

USE_DESTINATION_SPECIFIC_FILE_EXTENSION=1
DESTINATION_1_EXTENSION="HDT"

# IMU POSMV PASHR string
[INPUT_5]
DATA_TYPE="IMU"
DATA_SOURCE="POSMV"
SOURCE_TYPE=2
SERIAL_PORT_NAME="/dev/ttyUSB4"
BAUD=9600
TERM_CHAR="\n"
FILENAME_PREFIX="ar"

DESTINATION_1=1
DESTINATION_2=0
DESTINATION_2_IPADDRESS="10.10.10.1"
DESTINATION_2_SOCKET=55102
DESTINATION_3=0
DESTINATION_3_IPADDRESS="127.0.0.1"
DESTINATION_3_SOCKET=55102
DESTINATION_4=0
```

```
USE_DESTINATION_SPECIFIC_FILE_EXTENSION=1

DESTINATION_1_EXTENSION="PASHR"

# speedlog
[INPUT_6]
DATA_TYPE="NAV"
DATA_SOURCE="SPD"
SOURCE_TYPE=2
SERIAL_PORT_NAME="/dev/ttyUSB5"
BAUD=4800
TERM_CHAR="\n"
FILENAME_PREFIX="ar"

DESTINATION_1=1
DESTINATION_2=0
DESTINATION_2_IPADDRESS="192.147.41.60"
DESTINATION_2_SOCKET=55202
DESTINATION_3=0
DESTINATION_3_IPADDRESS="127.0.0.1"
DESTINATION_3_SOCKET=55202
DESTINATION_4=0

USE_DESTINATION_SPECIFIC_FILE_EXTENSION=1

DESTINATION_1_EXTENSION="SPD"

## RS485
[INPUT_7]
## DISABLED PROMPT 4 AND 5, REPLACED WITH RAD SENSOR
# These are needed
DATA_TYPE="MET"
DATA_SOURCE="RS485"
##

FILENAME_PREFIX="ar"

SOURCE_TYPE=2
SERIAL_PORT_NAME="/dev/ttyUSB6"
BAUD=9600

# time between sequences of prompts
PROMPT_SEQUENCE_INTERVAL=10

# time between prompts within the sequence
PROMPT_SEQUENCE_INTERVAL_BETWEEN_PROMPTS=1

# number of instruments to be prompted
PROMPT_SEQUENCE_NUMBER_OF_PROMPTS=3

# Port Vaisala  (PR0CRLF)
PROMPT_1_CHAR_1=0D
PROMPT_1_CHAR_2=0A
PROMPT_1_CHAR_3=50
PROMPT_1_CHAR_4=52
PROMPT_1_CHAR_5=30
PROMPT_1_CHAR_6=0D
PROMPT_1_CHAR_7=0A

# Starboard Vaisala (SR0CRLF)
PROMPT_2_CHAR_1=0D
```

```
PROMPT_2_CHAR_2=0A
PROMPT_2_CHAR_3=53
PROMPT_2_CHAR_4=52
PROMPT_2_CHAR_5=30
PROMPT_2_CHAR_6=0D
PROMPT_2_CHAR_7=0A

# SSW Fluorometer ($3RDCR)
PROMPT_3_CHAR_1=24
PROMPT_3_CHAR_2=33
PROMPT_3_CHAR_3=52
PROMPT_3_CHAR_4=44
PROMPT_3_CHAR_5=0D

## ASMET Shortwave Radiation (##SWR01C)
#PROMPT_4_CHAR_1=23
#PROMPT_4_CHAR_2=23
#PROMPT_4_CHAR_3=53
#PROMPT_4_CHAR_4=57
#PROMPT_4_CHAR_5=52
#PROMPT_4_CHAR_6=30
#PROMPT_4_CHAR_7=31
#PROMPT_4_CHAR_8=43

## ASMET Longwave Radiation (##LWR01C)
#PROMPT_5_CHAR_1=23
#PROMPT_5_CHAR_2=23
#PROMPT_5_CHAR_3=4C
#PROMPT_5_CHAR_4=57
#PROMPT_5_CHAR_5=52
#PROMPT_5_CHAR_6=30
#PROMPT_5_CHAR_7=31
#PROMPT_5_CHAR_8=43

### Define each of the different destinations

USE_DESTINATION_SPECIFIC_FILE_EXTENSION=1

## Port Vaisala destination options
# File
DESTINATION_1=1
DESTINATION_1_EXTENSION="XTP"
#UDP
DESTINATION_2=0
DESTINATION_2_IPADDRESS="192.147.41.60"
DESTINATION_2_SOCKET=55401
#UDP to CSV file
DESTINATION_3=0
DESTINATION_3_IPADDRESS="127.0.0.1"
DESTINATION_3_SOCKET=55400

## STB Vaisala destination options
# File
DESTINATION_4=1
DESTINATION_4_EXTENSION="XTS"
#UDP
DESTINATION_5=0
DESTINATION_5_IPADDRESS="192.147.41.60"
DESTINATION_5_SOCKET=55402
DESTINATION_6=0
DESTINATION_6_IPADDRESS="127.0.0.1"
DESTINATION_6_SOCKET=55400
```

```
## FLR destination options
# File
DESTINATION_7=1
DESTINATION_7_EXTENSION="FLR"
# UDP
DESTINATION_8=0
DESTINATION_8_IPADDRESS="192.147.41.60"
DESTINATION_8_SOCKET=55503
DESTINATION_9=0
DESTINATION_9_IPADDRESS="127.0.0.1"
DESTINATION_9_SOCKET=55503

### These have been replaced by RAD, but they maybe needed again
### since they are around lstolp 20170324
### SWR destination options
##Destination_10=1
##DESTINATION_10_EXTENSION="SWR"
## UDP
##DESTINATION_11=0
#DESTINATION_11_IPADDRESS="10.10.10.1"
#DESTINATION_11_SOCKET=55403
#DESTINATION_12=0
#DESTINATION_12_IPADDRESS="127.0.0.1"
#DESTINATION_12_SOCKET=55403

## LWR destination options
#Destination_13=1
#DESTINATION_13_EXTENSION="LWR"
## UDP
#DESTINATION_14=0
#DESTINATION_14_IPADDRESS="10.10.10.1"
#DESTINATION_14_SOCKET=55404
#DESTINATION_15=0
#DESTINATION_15_IPADDRESS="127.0.0.1"
#DESTINATION_15_SOCKET=55404

# Enable destinations for each prompt
USE_PROMPT_SPECIFIC_DESTINATIONS_FLAG=1
# PROMPT 1 Port Vaisala
PROMPT_1_DESTINATION_ENABLE_1=1
PROMPT_1_DESTINATION_ENABLE_2=1
PROMPT_1_DESTINATION_ENABLE_3=1
# PROMPT 2 STBD Vaisala
PROMPT_2_DESTINATION_ENABLE_4=1
PROMPT_2_DESTINATION_ENABLE_5=1
PROMPT_2_DESTINATION_ENABLE_6=1
#PROMPT 3 FLR
PROMPT_3_DESTINATION_ENABLE_7=1
PROMPT_3_DESTINATION_ENABLE_8=1
PROMPT_3_DESTINATION_ENABLE_9=1

# PROMPT 4 SWR
#PROMPT_4_DESTINATION_ENABLE_10=1
#PROMPT_4_DESTINATION_ENABLE_11=1
#PROMPT_4_DESTINATION_ENABLE_12=1
## PROMPT 5 LWR
#PROMPT_5_DESTINATION_ENABLE_13=1
#PROMPT_5_DESTINATION_ENABLE_14=1
#PROMPT_5_DESTINATION_ENABLE_15=1
```

```
# Answers terms CHAR
USE_PROMPT_SPECIFIC_TERM_CHAR=1
PROMPT_1_TERM_CHAR=0A
PROMPT_2_TERM_CHAR=0A
PROMPT_3_TERM_CHAR=0D

#PROMPT_4_TERM_CHAR=0A
#PROMPT_5_TERM_CHAR=0A

USE_PROMPT_SPECIFIC_DATA_TYPE=1
PROMPT_1_DATA_TYPE="MET"
PROMPT_2_DATA_TYPE="MET"
PROMPT_3_DATA_TYPE="SSW"
#PROMPT_4_DATA_TYPE="MET"
#PROMPT_5_DATA_TYPE="MET"

USE_PROMPT_SPECIFIC_SOURCE_NAME=1
PROMPT_1_SOURCE_NAME="WXTP"
PROMPT_2_SOURCE_NAME="WXTS"
PROMPT_3_SOURCE_NAME="FLR"
#PROMPT_4_SOURCE_NAME="SWR"
#PROMPT_5_SOURCE_NAME="LWR"

## AML Sound Velocity from em122
[INPUT_8]
DATA_TYPE="SSV"
DATA_SOURCE="AML_SSV"

FILENAME_PREFIX="ar"

SOURCE_TYPE=2
SERIAL_PORT_NAME="/dev/ttyUSB7"
BAUD=9600
TERM_CHAR="\n"

USE_DESTINATION_SPECIFIC_FILE_EXTENSION=1
DESTINATION_1=1
DESTINATION_1_EXTENSION="AML_SSV"
DESTINATION_2=0
DESTINATION_2_IPADDRESS="192.147.41.60"
DESTINATION_2_SOCKET=55505
DESTINATION_3=0
DESTINATION_3_IPADDRESS="127.0.0.1"
DESTINATION_3_SOCKET=55505

## SBE48
[INPUT_9]
DATA_TYPE="SSW"
DATA_SOURCE="SBE48"
TERM_CHAR="\n"

FILENAME_PREFIX="ar"

SOURCE_TYPE=2
SERIAL_PORT_NAME="/dev/ttyUSB8"
BAUD=9600

USE_DESTINATION_SPECIFIC_FILE_EXTENSION=1
DESTINATION_1=1
DESTINATION_1_EXTENSION="SBE48"
DESTINATION_2=0
```

```
DESTINATION_2_IPADDRESS="192.147.41.60"
DESTINATION_2_SOCKET=55502
DESTINATION_3=0
DESTINATION_3_IPADDRESS="127.0.0.1"
DESTINATION_3_SOCKET=55502

## SBE45
[INPUT_10]
DATA_TYPE="SSW"
DATA_SOURCE="SBE45"

FILENAME_PREFIX="ar"

SOURCE_TYPE=2
SERIAL_PORT_NAME="/dev/ttyUSB9"
BAUD=9600
TERM_CHAR="\n"

USE_DESTINATION_SPECIFIC_FILE_EXTENSION=1
DESTINATION_1=1
DESTINATION_1_EXTENSION="SBE45"
DESTINATION_2=0
DESTINATION_2_IPADDRESS="192.147.41.60"
DESTINATION_2_SOCKET=55501
DESTINATION_3=0
DESTINATION_3_IPADDRESS="127.0.0.1"
DESTINATION_3_SOCKET=55501
#DESTINATION_4=0
#DESTINATION_4_IPADDRESS="10.10.10.1"
#DESTINATION_4_SOCKET=55501

## GRAVIMETER
[INPUT_11]
DATA_TYPE="BGM"
DATA_SOURCE="BGM3"
SOURCE_TYPE=2
SERIAL_PORT_NAME="/dev/ttyUSB10"
BAUD=9600
TERM_CHAR="\n"
FILENAME_PREFIX="ar"

DESTINATION_1=1
DESTINATION_2=0
DESTINATION_2_IPADDRESS="192.147.41.60"
DESTINATION_2_SOCKET=55701

USE_DESTINATION_SPECIFIC_FILE_EXTENSION=1
DESTINATION_1_EXTENSION="BGM"

## Winch Data
[INPUT_12]
DATA_TYPE="WINCH"
DATA_SOURCE="MTN"
SOURCE_TYPE=2
SERIAL_PORT_NAME="/dev/ttyUSB11"
BAUD=19200
TERM_CHAR="\n"
FILENAME_PREFIX="ar"

DESTINATION_1=1
DESTINATION_2=0
DESTINATION_2_IPADDRESS=“192.147.41.60"
```

```
DESTINATION_2_SOCKET=55801
DESTINATION_3=0
DESTINATION_3_IPADDRESS=“192.147.41.61"
DESTINATION_3_SOCKET=55801

USE_DESTINATION_SPECIFIC_FILE_EXTENSION=1
DESTINATION_1_EXTENSION="MTN_WINCH"

## Sonardyne USBL ATS ASCII string
[INPUT_13]
DATA_TYPE="NAV"
DATA_SOURCE="USBL"
SOURCE_TYPE=2
SERIAL_PORT_NAME="/dev/ttyUSB12"
BAUD=9600
TERM_CHAR="\n"
FILENAME_PREFIX="ar"

DESTINATION_1=1
## This needs to be changed
#DESTINATION_2=0
#DESTINATION_2_IPADDRESS="10.10.10.1"
#DESTINATION_2_SOCKET=55803

USE_DESTINATION_SPECIFIC_FILE_EXTENSION=1
DESTINATION_1_EXTENSION="ATS_ASCII"

#RAD
[INPUT_14]
DATA_TYPE="MET"
DATA_SOURCE="RAD"
SOURCE_TYPE=2
SERIAL_PORT_NAME="/dev/ttyUSB13"
BAUD=19200
TERM_CHAR="\n"
FILENAME_PREFIX="ar"

DESTINATION_1=1
DESTINATION_2=0
DESTINATION_2_IPADDRESS="127.0.0.1"
DESTINATION_2_SOCKET=55406
DESTINATION_3=0
DESTINATION_3_IPADDRESS="192.147.41.60"
DESTINATION_3_SOCKET=55406

USE_DESTINATION_SPECIFIC_FILE_EXTENSION=1
DESTINATION_1_EXTENSION="RAD"

##PAR
[INPUT_15]
DATA_TYPE="MET"
DATA_SOURCE="PAR"
SOURCE_TYPE=2
SERIAL_PORT_NAME="/dev/ttyUSB14"
BAUD=9600
TERM_CHAR="\n"
FILENAME_PREFIX="ar"

DESTINATION_1=1

DESTINATION_2=0
DESTINATION_2_IPADDRESS="127.0.0.1"
```

```
DESTINATION_2_SOCKET=55405
DESTINATION_3=0
DESTINATION_3_IPADDRESS="192.147.41.60"
DESTINATION_3_SOCKET=55405

USE_DESTINATION_SPECIFIC_FILE_EXTENSION=1
DESTINATION_1_EXTENSION="PAR"

##Flowmeter
[INPUT_16]
DATA_TYPE="SSW"
DATA_SOURCE="FLOW"
SOURCE_TYPE=2
SERIAL_PORT_NAME="/dev/ttyUSB15"
BAUD=9600
TERM_CHAR="\n"
FILENAME_PREFIX="ar"

DESTINATION_1=1

DESTINATION_2=0
DESTINATION_2_IPADDRESS="127.0.0.1"
DESTINATION_2_SOCKET=55506
DESTINATION_3=0
DESTINATION_3_IPADDRESS="192.147.41.60"
DESTINATION_3_SOCKET=55506

USE_DESTINATION_SPECIFIC_FILE_EXTENSION=1
DESTINATION_1_EXTENSION="FLOW"

### Network Threads

## Knudsen 3260
[INPUT_17]
DATA_TYPE="DEP"
DATA_SOURCE="KN3260"

FILENAME_PREFIX="ar"

SOURCE_TYPE=0
IN_SOCKET=55604
### Define each of the different destinations

USE_DESTINATION_SPECIFIC_FILE_EXTENSION=1
DESTINATION_1=1
DESTINATION_1_EXTENSION="KN3260"
DESTINATION_2=0
DESTINATION_2_IPADDRESS="192.147.41.60"
DESTINATION_2_SOCKET=55603
DESTINATION_3=0
DESTINATION_3_IPADDRESS="127.0.0.1"
DESTINATION_3_SOCKET=55603

#em122
[INPUT_18]
DATA_TYPE="DEP"
DATA_SOURCE="EM122"

FILENAME_PREFIX="ar"

SOURCE_TYPE=0
IN_SOCKET=55601
```

```
### Define each of the different destinations

USE_DESTINATION_SPECIFIC_FILE_EXTENSION=1
DESTINATION_1=1
DESTINATION_1_EXTENSION="EM122"
DESTINATION_2=0
DESTINATION_2_IPADDRESS="192.147.41.60"
DESTINATION_2_SOCKET=55602
DESTINATION_3=0
DESTINATION_3_IPADDRESS="127.0.0.1"
DESTINATION_3_SOCKET=55602

#em710
[INPUT_19]
DATA_TYPE="DEP"
DATA_SOURCE="EM710"

FILENAME_PREFIX="ar"

SOURCE_TYPE=0
IN_SOCKET=55605
### Define each of the different destinations

USE_DESTINATION_SPECIFIC_FILE_EXTENSION=1
DESTINATION_1=1
DESTINATION_1_EXTENSION="EM710"
DESTINATION_2=0
DESTINATION_2_IPADDRESS="192.147.41.60"
DESTINATION_2_SOCKET=55606
DESTINATION_3=0
DESTINATION_3_IPADDRESS="127.0.0.1"
DESTINATION_3_SOCKET=55606

[INPUT_20]
DATA_TYPE="NAV"
#DATA_SOURCE="GYRO1_11424"
DATA_SOURCE="GYRO"

FILENAME_PREFIX="ar"

SOURCE_TYPE=0
IN_SOCKET=55110
### Define each of the different destinations

USE_DESTINATION_SPECIFIC_FILE_EXTENSION=1
DESTINATION_1=1
DESTINATION_1_EXTENSION="GYRO1_11424"
DESTINATION_2=0
DESTINATION_2_IPADDRESS="127.0.0.1"
DESTINATION_2_SOCKET=55105

[INPUT_21]
DATA_TYPE="NAV"
#DATA_SOURCE="GYRO2_11431"
DATA_SOURCE="GYRO"

FILENAME_PREFIX="ar"

SOURCE_TYPE=0
IN_SOCKET=55111
### Define each of the different destinations
```

```
USE_DESTINATION_SPECIFIC_FILE_EXTENSION=1
DESTINATION_1=1
DESTINATION_1_EXTENSION="GYRO2_11431"
DESTINATION_2=0
DESTINATION_2_IPADDRESS="127.0.0.1"
DESTINATION_2_SOCKET=55106

[GUI_THREAD]
PORT=60001
#IPADDRESS=255.255.255.255
TO_IP_ADDRESS=127.0.0.1
TO_PORT=10500
destination_1=CMD_THREAD_1

# network threads
[TCP_SERVER_THREAD]
TCP_SERVER_THREAD_SOCKET=40000
```

Document History

- R001 – 2012-12-15
  - This version documented the code as it was delivered by Jon Howland
- R002 – 2012-12-26
  - Added this Document History section.
  - Modified to reflect changes in code made by T. Thiel.
- R003 – 2013-01-11
  - Added section describing dsLog status output messages.
- R006 – 2017-08-14
  - Extensive extension of documentation.
  - Use Armstrong configuration file to provide tutorial style walk-through of configuration file syntax.