

WHEN YOU NEED TO BE SURE



CyFlex® AVL Link Application User Guide

Version 6

May 24, 2022

Developed by SGS North America, Inc.

Version History

Version	Date	Revision Description
1	1/25/2016	Initial publication
2	8/23/2018	Format to SGS brand
3	3/31/2020	Retrofit to new template
4	8/24/2020	<p>Added <i>Section 2.1 Installing the Software on a Windows 10 Computer</i> on page 2</p> <p>Refreshed <i>Section 4 Specification Files</i> on page 4 with latest format to support the Avl XION system:</p> <ul style="list-style-type: none"> The file format now includes @ Keywords to separate information types. Keywords now may be used to adjust timeout waiting on AVL device response instead of hardcoded values. Register Name Keywords may now be used as a system can have two concurrently running copies of <code>avl_link</code> running Added <code>AVL_INSTRUMENT_NAME</code> so each instance of <code>AVL_Link</code> can be Identified by that name. Currently our software default name is <code>INDICOM</code> however these are user-configurable on Windows software. <p>Example: Indicom system name: <code>INDICOM</code> and the Avl X-ION system could be identified as <code>AVL-XION</code>.</p>
5	8/30/2021	Added hypertext linked cross-references to <code>cyflex.com</code> usage help for <code>csdump</code> and <code>avl_link</code> .
6	5/24/2022	Updated hypertext linked cross-references to <code>cyflex.com</code> usage help for <code>csdump</code> and <code>avl_link</code>

Document Conventions

This document uses the following typographic and syntax conventions.

- Commands, command options, file names or any user-entered input appear in Courier type. Variables appear in Courier italic type.
Example: Select the `cmdapp-relVersion-buildVersion.zip` file...
- User interface elements, such as field names, button names, menus, menu commands, and items in clickable dropdown lists, appear in Arial bold type.
Example: **Type**: Click **Select Type** to display drop-down menu options.
- Cross-references are designated in Arial italics.
Example: Refer to *Figure 1...*

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

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

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

Related Documents

[CyFlex Connection Server Setup](#)

[Human Service Data API Suite \(HSDA\)](#)

AVL Indicom Documentation; see <https://www.avl.com/-/indicom-indicating-software>

CyFlex Documentation

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

Table of Contents

1	OVERVIEW	1
1.1	CYFLEX VARIABLES	1
2	PREREQUISITE SYSTEM TASKS	2
2.1	INSTALLING THE SOFTWARE ON A WINDOWS 10 COMPUTER	2
2.2	SPECIFIC SYSTEM CONSIDERATIONS	2
3	RUNNING AND USING AVL LINK	3
4	SPECIFICATION FILES	4
4.1	SPECIFICATION FOR AVL LINK	4
4.1.1	Example for AVL XION	7
4.2	SPECIFICATION FOR AVL LINK GEN_LABELS	8
4.3	SAMPLE SPECIFICATION FOR GP TEST SCRIPT	10

LIST OF TABLES

TABLE 1: CYFLEX VARIABLES WITH AVL LINK	1
TABLE 2: EVENT NAMING FOR INDEPENDENT INSTRUMENT CONTROL	4

1 Overview

The AVL Link application links to the Indicom combustion analysis system to provide similar services as HSDA link does with the HSDA system. To accomplish this, the AVL link application does the following:

1. Communicates with the AVL Indicom system via the connection server
2. Monitors and sets the same events as HSDA link
3. Sets and gets variables associated with each event

The names of the events and variables are controlled by the spec file, usually `/specs/avl_link.nnn`. Refer to *Section 4 Specification File* on page 4.

`avl_link` should work with most `gp` test scripts that worked with HSDA. Refer to *Section 4.3 Sample Specification for GP Test Script* on page 10.

1.1 CyFlex Variables

Table 1 lists the CyFlex Variables used with the AVL Link application. Default values are used if a variable is undefined.

Table 1: CyFlex Variables with AVL Link

CyFlex Variable	Type	Default	Description
AVL_TESTPATH	String	Puma.xpa	Name of test to run on the Indicom system
AVL_DATAPATH	String	C:\IndiCom1.60\MyData\IFiles\	Save file location on the Indicom system
Test_ID	String	testid	Current Asset/ CyFlex Test ID
PAMdatapoint	Integer	ffff	PAM data point number
cell_number	Integer	xxx	Test cell number

The current date, `Test_ID`, `PAMdatapoint`, and `cell_number` are used to create the data file names on the Indicom system (i.e.

`C:\IndiCom1.60\MyData\IFiles\20070731EVEREST_ISL21486.305`).

2 Prerequisite System Tasks

2.1 Installing the Software on a Windows 10 Computer

Refer to Knowledge Article *Installing the AVL Link Application on Windows 10*, available on the Cummins Wiki.

2.2 Specific System Considerations

It is possible to run both the AVL Indicom system and HSDA at the same time, but significant modifications to the event names must be done. The default mode is to run `avl_link` as a substitute for `hsda_link`. Unless your system has been explicitly setup to work with both `avl_link` and `hsda_link` at the same time, ensure `hsda_link` is not running before starting `avl_link`.

`avl_link` also requires that a specific test setup is running on the AVL Indicom system. Enter the correct test setup name into the `AVL_TESTPATH` asset string variable before starting `avl_link`.

During setup it is advisable to use `csdump` to test that the connection server is communicating with the AVL Indicom system. Refer to [cyflex.com](#) usage help for [csdump](#).

3 Running and Using AVL Link

Run `avl_link` to link to the Indicom combustion analysis system. Specify a specification file. Refer to cyflex.com usage help for [avl_link](#).

Once the connection server and `avl_link` have been started, usually the first test script to try is `gp_check_hstda`. If this test is successful, check a test fuel reading (`fr_hstda`) for correct operation.

Assuming the connection server is operating correctly, monitor the events related to `avl_link` with the `events` utility. Correlate the events with the state of the `gp` test script. For example:

```
$> events
HS_Reset
TS_StrtRdy
TS_StrtAcq
TS_OpCondCmp
HS_Initd
HS_AcqInPrg
HS_AcqCmp
HS_AnlsCmp
```

Contact SGS CyberMetrix if experiencing program anomalies with `avl_link`.

4 Specification Files

If running multiple copies of AVL Link for Indicom or XION, the event keywords must be unique between the two, other than those for *TIMEOUT* keywords.

Change the following events to ensure the `avl_link_xion.def` and `avl_link_indicom.def` independently control the instruments:

Table 2: Event Naming for Independent Instrument Control

Event Name	Example Indicom Name	Example XION Name
@RESET_READY	HS_Reset	HS_Reset_X
@START_READY	TS_StrtRdy	TS_StryRdy_X
@START_ACQ	TS_StrtRdy	TS_StrtAcq_X
@TS_OP_COND_COMPLETE	TS_OpCondCmp	TS_OpCondCmp_X
@HS_QUIT	HS_QUIT	HS_QUIT_X

4.1 Specification for AVL Link

The events in the specification file must remain the same as with AVL Link.

Note:

Event names must be unique. Distinguish events for AVL-XION by appending X to an event name. For example, an event for Indicom may be `HS_Reset` and for AVL-XION the event may be `HS_ResetX`.

The following is an example of the AVL Link specification file:

```
@AVL_REGISTERED_NAME
AVL_LINK

# This is then the name of Indicom Server on the Windows System
@AVL_INSTRUMENT_NAME
indicom

$
# Maximum time in msec to wait for results.
@AVL_RESULT_TIMEOUT
10000

# Maximum time in msec to wait for response after START_ACQ.
@AVL_START_ACQ_TIMEOUT
20000

# Maximum time in msec to wait for response after START_ANA.
@AVL_START_ANA_TIMEOUT
120000
```

```

# Maximum time in msec to wait for response after issuing EndSession
@AVL_QUIT_TIMEOUT
5000

# Maximum time in msec to wait response
@AVL_DEFAULT_TIMEOUT
5000

# Reset Ready Event - input event to avl_link
@RESET_READY
HS_Reset

#usually there are no variables set to gotten from AVL system
#upon this event
$
# Asset start ready event - input event to avl_link
@START_READY
TS_StrtRdy

#usually there are no variables set to gotten from AVL system
#upon this event
$
# Asset start acquisition event - input event to avl_link
@START_ACQ
TS_StrtAcq
# ASSET_Variable or value ("")      AVL variable      AVL units

# change the number of engine cycles for test
o      Indicom_Cycles                NACY                none

# The following section is an example of writing and reading string
#variables to/from Indicom.
# NOTE:  Operating variables can NOT be string variables (OP_P0n),
#Indicom requires that these be numbers.

o  Darts_Program          %Darts_Program      str
o  Darts_TestID          %Darts_TestID       str
o  Darts_Test            %Darts_Test          str
o  Darts_Mode            %Darts_Mode         str
o  Darts_Measurement     %Darts_Measurement str
#
#  read back %Darts_Program to verify it was set

i  Darts_Program_verify  %Darts_Program      str
$

# Event associated with reading variables FROM AVL system.
@TS_OP_COND_COMPLETE
TS_OpCondCmp - input event to avl_link

```

```

# ASSET_Variable or value ("")      AVL variable  AVL units
o int_mnf_p                          OP_P01      kpa
o int_mnf_t  OP_P02      deg_C
o FR_TORQUE                          OP_P03      n-m
o FR_RPM                              OP_P04      rpm
o FR_Fuel_rate  OP_P05      kg/hr
o PAMdatapoint                          OP_P06      none
$
#HSDA_initialized - output event from avl_link
@HS_INITIALIZED
HS_Initd
$
# HSDA_acq_started - output event from avl_link
@HS_ACQ_IN_PROGRESS
HS_AcqInPrg
$
# HSDA_acq_complete - output event from avl_link
@HSDA_ACQ_COMPLETE
HS_AcqCmp
$
# HSDA_analysis_complete - output event from avl_link
@HSDA_ANLS_COMPLETE
HS_AnlsCmp

# ASSET_Variable or value ("")      AVL variable  AVL units
i  h_pk_cyl_p1                        PMAX1      psi
i  h_pkcyl_1_ca                       APMAX1     deg
i  h_sda_soi_1i                       SOI1       deg
i  h_eer_1                             EEEXP1     none
i  h_ctrd95_1                          CTR1       none
i  h_strt_comb1                        HR_Start1  deg
i  h_end_comb1                         HR_End1    deg
i  h_nimep1                            IMEP1     psi
i  h_gimep1                            GIMEP1    psi
i  h_pmepl                             PMEP1     psi
$
#HSDAQuit drop connection to AVL system - input event to avl_link
@HS_QUIT
HS_Quit
$

# Name of Event use send status
@AVL_LINK_STATUS_EVENT
AVLinkStat
$

# String Variable to store the name of the data file
@DATA_FILENAME_STRING_VARNAME
AvLinkDataFile

```

```
$
```

```
# Logical variable used to indicate when to save the data file
@SAVE_DATAFILE_LOGICAL_VARNAME
AvlLinkSaveDataFile
```

```
$
```

```
# Name of String Variable used to store name of test "test.xpa"
@TESTPATH_STRING_VARNAME
AVL_TESTPATH
```

```
$
```

```
# Name of String Variable to use to directory containing data
@DATAPATH_STRING_VARNAME
AVL_DATAPATH
```

```
$
```

4.1.1 Example for AVL XION

The following is an example of changes to the specification file for AVL XION. This file is located in `/specs.def/avl_link_xion.def`.

```
@AVL_REGISTERED_NAME
AVL_LINK_XION
```

```
# This is then the name of Indicom Server on the Windows System
```

```
@AVL_INSTRUMENT_NAME
avl-xion
```

```
.
.
```

4.2 Specification for AVL Link gen_labels

VERSION_2 (do not remove this line)

@REG_NAME

GL_hsdA

#label	units	format	initial_value	interval/event	hst_flag	tolerance
std_dev1 -999[none]	none	1	-999	FR_IP	OFF	1.00
std_dev2 -999[none]	none	1	-999	FR_IP	OFF	1.00
std_dev1CA -999[none]	none	1	-999	FR_IP	OFF	1.00
std_dev2CA -999[none]	none	1	-999	FR_IP	OFF	1.00

.
.
.

```
# *****
# Integer variable sent to the hsdA to prevent offset. (PAM_point updates
# at the end of the fuel reading. This info sent to hsdA prior to that)
#
# dco 7/12/96
# *****
PAMdatapoint none - FR_RQST OFF 1
PAM_point + 1[none]
```

Indicom_Cycles none 100 - OFF 1

-

\$

```

#label      true_event      false_event  true_desc   false_desc  int/event  hst_flag
hsda999     set_hsda           trigger_hsda  ON          OFF         -         OFF
-

#label      true_event      false_event  true_desc   false_desc  int/event  hst_flag
hsda_complete  hsda_complete_ev  -           Complete   InProgress  -         OFF
-
$
HSDA_file  HS_AnlsCmp
AvlLinkDataFile

$
fr_HSDA   " "           fr_done           OFF
AvlLinkDataFile

#AVL_TESTPATH "TC421_new_cec.xpa" - OFF
AVL_TESTPATH "IndicomSVB.xpa" - OFF
-

AVL_DATAPATH "M:\\" - OFF
-

fr_XION   " "           fr_done           OFF
XionLinkDataFile

XION_TESTPATH "XionSVB.xpa" - OFF
-

XION_DATAPATH "M:\\" - OFF
-
$end of strings
$end of computed status
XionLinkStat
TS_AcqInPrg
$ end of section for creating zero-length "signal" events

```


4.3 Sample Specification for GP Test Script

The following is an example of a gp_test script that communicates with avl_link.

```
#####

#start_mode ( mode where the test begins )
1

@GLOBAL_EVENTS
#event_name          next_mode          test_procedure
abort_limit          -                /specs/gp/gp_shutdown
emergency            -                /specs/gp/gp_emergency

@REGISTERED_EVENTS
#event_name          next_mode          test_procedure
idle_mode            -                /specs/gp/gp_idle
stop_test            -                /specs/gp/gp_shutdown

#*****
# Call gp_reset_hsda only if gp_flex_fr is the root_procedure
#*****

@MODE
#mode_number      max_timeout      default_next_mode
1                 0.0[sec]        100
# description
Reset HSDA

@IF_TRUE
"@strcmp_lbl_lit(root_procedure, '/specs/gp/gp_flex_fr')"
hsda_req

@SET_EVENTS
AT_START      set_hsda

@ELSE_MODE
100

@PROCEDURE
/specs/gp/gp_reset_hsda

#*****
# Execute mode 100 when Smoke and Hsda Required else check 200
#*****

@MODE
#mode_number      max_timeout      default_next_mode
100               "target_fr_tm + 3[min]"        190
```

```

# description
fuel + hsda + smoke

@IF_TRUE
hsda_req
smoke_req

@ELSE_MODE
200

@FUEL_READING
#start_type          stop_path
AT_START             NONE
#number_of_readings interval  extern_sync_event  desired_time
1                   0.00[sec]      -                  target_fr_tm

@FUEL_READING_SYNC
#timeout      output_event  <list of up to 4 input events>
0[s]         TS_StrtAcq    fr_ave_strt FR_RQST
0[s]         TS_AckInPrg  fr_ready HS_AckInPrg HS_AckInPrg_X
0[s]         TS_OpCondCmp TS_AckInPrg HS_AcqCmp HS_AckCmp_X
0[s]         fr_write_ok  HS_AnlsCmp HS_AnlsCmp SM_coll_done
# 0[s]       fr_write_ok  HS_AnlsCmp SM_coll_done FM_coll_done

@SET_EVENTS
AT_START      set_hsda

@PARAMETERS
#start_type label      value
AT_START     fr_wrt_delay 10[min]

@TERMINATION_EVENTS
#event_name      termination_path
fr_done          105

#*****

@MODE
#mode_number      max_timeout      default_next_mode
105               10.0[sec]       900
# description
Loop Control

@LOOP_CONTROL
# num_repeats      next_loop_mode      loop_counter
num_of_fr          100                  -

```

```

*****
@MODE
#mode_number      max_timeout      default_next_mode
190                10.0[sec]        /specs/gp/gp_shutdown
# description
Failure

@PARAMETERS
#start_type label      value
AT_START    NOTIFY 'Failed to Complete Fuel_Smoke_HSDA Reading' NO

*****
# Execute mode 200 when Smoke Required and Not HSDA else check 300
*****

@MODE
#mode_number      max_timeout      default_next_mode
200                "target_fr_tm + 3[min]"      290
# description
fuel + smoke

@IF_TRUE
!hsda_req
smoke_req

@ELSE_MODE
300

@FUEL_READING
#start_type      stop_path
AT_START         NONE
#number_of_readings interval      extern_sync_event      desired_time
1                0.00[sec]      -                        target_fr_tm

@FUEL_READING_SYNC
#timeout      output_event      <list of up to 4 input events>
0[s]          fr_write_ok      fr_ready      SM_coll_done
# 0[s]        fr_write_ok      fr_ready      SM_coll_done
FM_coll_done

@PARAMETERS
#start_type label      value
AT_START    fr_wrt_delay 10[min]
    
```

```

@TERMINATION_EVENTS
#event_name          termination_path
fr_done              205

#*****

@MODE
#mode_number          max_timeout          default_next_mode
205                  10.0[sec]            900
# description
Loop Control

@LOOP_CONTROL
# num_repeats         next_loop_mode        loop_counter
num_of_frs           200                   loop1

#*****

@MODE
#mode_number          max_timeout          default_next_mode
290                  10.0[sec]            /specs/gp/gp_shutdown
# description
Failure

@PARAMETERS
#start_type label          value
AT_START    NOTIFY 'Failed to Complete Fuel_Smoke Reading'    NO

#*****
# Execute mode 300 Hsda Required and Smoke is Not required else check
#400
#*****

@MODE
#mode_number          max_timeout          default_next_mode
300                  "target_fr_tm + 3[min]"    390
# description
fuel + hsda

@IF_TRUE
hsda_req
!smoke_req

@ELSE_MODE
400

@FUEL_READING
#start_type          stop_path
AT_START             NONE

```

```

#number_of_readings  interval      extern_sync_event      desired_time
1                    0.00[sec]              -                       target_fr_tm

@FUEL_READING_SYNC
#timeout            output_event            <list of up to 4 input events>
0[s]                TS_StrtAcq              fr_ave_strt
0[s]                TS_AckInPrg             HS_AckInPrg HS_AckInPrg_X
0[s]                TS_OpCondCmp            TS_AckInPrg HS_AcqCmp HS_AcqCmp_X
0[s]                fr_write_ok             HS_AnlsCmp HS_AnlsCmp_X
# 0[s]              fr_write_ok             HS_AnlsCmp FM_coll_done

@PARAMETERS
#start_type label          value
AT_START    fr_wrt_delay  10[min]

@TERMINATION_EVENTS
#event_name          termination_path
fr_done              305

@SET_EVENTS
AT_START    set_hsda

#*****

@MODE
#mode_number          max_timeout           default_next_mode
305                  10.0[sec]             900
# description
Loop Control

@LOOP_CONTROL
# num_repeats         next_loop_mode         loop_counter
num_of_frs           300                    -

#*****

@MODE
#mode_number          max_timeout           default_next_mode
390                  10.0[sec]             /specs/gp/gp_shutdown
# description
Failure

@PARAMETERS
#start_type label          value
AT_START    NOTIFY    'Failed to Complete Fuel_HSDA Reading'    NO

```

```

*****
# Execute mode 400 No Hsda and No Smoke is Required else check 500
*****

@MODE
#mode_number          max_timeout          default_next_mode
400                   "target_fr_tm + 3[min]"          490
# description
fuel reading

@IF_TRUE
!hsda_req
!smoke_req

@ELSE_MODE
500

@FUEL_READING
#start_type          stop_path
AT_START             NONE
#number_of_readings interval      extern_sync_event      desired_time
1                    0.00[sec]              -                       target_fr_tm

@FUEL_READING_SYNC
#timeout          output_event      <list of up to 4 input events>
# 0[s]           fr_write_ok      fr_ready      FM_coll_done
0[s]            fr_write_ok      fr_ready

@PARAMETERS
#start_type label          value
AT_START   fr_wrt_delay  10[min]

@TERMINATION_EVENTS
#event_name          termination_path
fr_done              405

*****

@MODE
#mode_number          max_timeout          default_next_mode
405                   10.0[sec]           900
# description
Loop Control

@LOOP_CONTROL
# num_repeats          next_loop_mode      loop_counter
num_of_frs            400                 -

```



```

*****
@MODE
#mode_number      max_timeout      default_next_mode
490                10.0[sec]        /specs/gp/gp_shutdown
# description
Failure

@PARAMETERS
#start_type label      value
AT_START  NOTIFY  'Failed to Complete Fuel Reading'      NO

*****

@MODE
#mode_number      max_timeout      default_next_mode
500                10.0[sec]        /specs/gp/gp_shutdown
# description
Error The process should never find this mode

# Should only be possible to find this mode if the *_req values
# are changing during the test.

*****

@MODE
#mode_number      max_timeout      default_next_mode
900                0.0[sec]        100
# description
adv to take another reading

@IF_TRUE
"@strcmp_lbl_lit(root_procedure, '/specs/gp/gp_flex_fr')"

@ELSE_MODE
905

*****

@MODE
#mode_number      max_timeout      default_next_mode
905                1.0[sec]        RETURN
# description
Return Mode
#####

```