



# **CyFlex® PAM Data Collection and Transfer**

**Version 8**

February 9, 2024

**Developed by Transportation Laboratories**



### Version History

Version	Date	Revision Description
1	1/25/2016	Initial publication
2	8/23/2018	Format to SGS brand
3	4/16/2020	Retrofit to new template
4	5/18/2020	Added Section 6 <i>Datapoint Calculations for Engine Performance</i> on page 10
5	10/5/2021	Revised <i>Section 5 PAM Programs</i> on page 7 to remove command usage content and replace with hypertext linked cross-references to cyflex.com usage help as applicable.
6	5/26/2022	Updated all hypertext linked cross-references to cyflex.com usage help descriptions
7	2/7/2024	Rebrand to TRP Laboratories
8	2/9/2024	Removed section describing <code>get_pam_data</code> as it is no longer included in CyFlex 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.

### 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

<b>1</b>	<b>OVERVIEW</b> .....	<b>1</b>
<b>2</b>	<b>DATA FILES</b> .....	<b>2</b>
2.1	PAM FILE TYPES.....	2
2.2	PAM DATA FILE NAMES.....	2
2.3	PAM DATA FILE TRANSFER.....	2
2.4	PAM DATA FILE ACKNOWLEDGEMENT.....	3
<b>3</b>	<b>SPECIFICATION AND KEYWORD FILES</b> .....	<b>4</b>
3.1	PAM HEADER BLOCK.....	4
3.2	PAM PARAMETER BLOCK.....	4
3.3	PAM DATAPOINT BLOCK.....	5
3.3.1	PAM Keyword Files.....	5
3.4	PAM DATAPOINT LOG FILE.....	5
<b>4</b>	<b>OPERATING TIPS</b> .....	<b>6</b>
4.1	CHANGING A CHANNEL FOR A PAM DATAPOINT.....	6
4.2	CHANGING A PARAMETER FOR A PAM HEADER.....	6
4.3	CHANGING A PAM TEST ID.....	6
<b>5</b>	<b>PAM PROGRAMS</b> .....	<b>7</b>
5.1	SETPAMPPOINT.....	7
5.2	SETPAMTESTID.....	7
5.3	DATAPOINT.....	7
5.4	PAM_SPECS.....	8
5.5	TRANMOVE.....	8
5.6	EXTDATMAN.....	9
5.7	PAMRESEND.....	9
5.8	PAMRESENDSITE.....	9
<b>6</b>	<b>DATAPOINT CALCULATIONS FOR ENGINE PERFORMANCE</b> .....	<b>10</b>
6.1	FLOW RATE CALCULATIONS.....	10
6.2	FUEL AIR RATIO CALCULATIONS.....	10
6.3	VOLUMETRIC EFFICIENCY.....	11
6.4	PERFORMANCE CALCULATIONS.....	11
	<b>APPENDIX A. SAMPLE PAM SPECIFICATION FILE</b> .....	<b>12</b>

**LIST OF TABLES**

TABLE 1: PAM KEYWORDS AND FILES ..... 5

### 1 Overview

CyFlex test systems support the Performance Analysis Module (PAM) system, an Oracle database running on a network server. PAM provides comprehensive database and analysis facilities for engine test data. Its primary data format is "Fuel Readings", time-averaged sets of engine performance measurements which are the most widely used Engineering test data set.

This document covers the specification of PAM measurement channels and test parameters, plus the CyFlex system of PAM data files and their transfer to PAM.

## 2 Data Files

### 2.1 PAM File Types

The PAM database needs two distinct types of files from each test cell: Headers and Datapoints. Both file types are written by CyFlex programs at each participating test cell and are transferred automatically to the PAM database.

1. *Headers* are files sent to PAM in advance of acquiring Datapoints to document the test environment and configuration. The particular attributes are selected from a large set of standardized PAM keywords that identify test parameters and units to the PAM system.
2. *Datapoints* are sets of engineering values, one for each channel specified in a PAM Specs file described below. These values are time averages of channel measurements, accumulated over a *Fuel Reading*. Fuel Readings are typically specified by a certain weight of fuel, and in turn the time required to consume it at a steady-state operating point.

Instantaneous values may be included in Datapoints and are also specified in PAM Specs. These *Snapshot* data can complement the Fuel Reading data to characterize operating conditions at the end of a Fuel Reading or represent computed values desired by some other process.

A typical test is represented in PAM as an initial Header, followed by a series of Datapoints collected over a series of steady-state operating points. Multiple Datapoints may be taken at each operating point. If a change in the test environment occurs, e.g., the value of an engine parameter changes, then a new Header is sent to document subsequent Datapoints.

### 2.2 PAM Data File Names

For a given test, Datapoint filenames are designated by a test ID and a sequential datapoint number. The test ID is a unique string of 1 - 25 characters. A typical Datapoint filename might be `ASSET2000.100`, the 100th datapoint for the test ID `ASSET2000`.

Headers use the same filename convention, except the datapoint number is the "effective datapoint number", the next datapoint collected under that Header. Assuming the current PAM Test ID is `ASSET2005` and the current Datapoint number is 324, then running the `pam_specs` command would generate a header file with a filename of `ASSET2005.325`.

### 2.3 PAM Data File Transfer

PAM data file transfers from CyFlex test cells are performed automatically. To the user, test cell data simply appears in the PAM system shortly after it is acquired.

In fact, each active CyFlex test cell polls continuously for data files to transfer to the database server. When a PAM Header or Datapoint file is created, the `tranMove` program at the cell copies it to a dedicated CyFlex "communications node", refer to *Section 5.5 tranMove* on page 8. This communication node conducts file transfers to non-CyFlex network servers under the TCP/IP File Transfer Protocol (FTP).

PAM Header files are written by the `pam_specs` program as the user requests and at startup. PAM Datapoint files are written automatically by the `datapoint` program at the end of a Fuel Reading. The two file types are kept in different sub-directories of `/data/` on the test cell node, depending on their type and status: Refer to *Section 5.3 datapoint* on page 7 and *Section 5.4 pam\_specs* on page 8.

Status	PAM Headers	PAM Datapoints
Transfer	/data/transfer/ PAM_header/	/data/transfer/ PAM_datapoint/
Archive	/data/PAM_header/	/data/PAM_datapoint/

Initially, both of these files are created as "Transfer" files, ready to transfer to PAM. As files are copied to the communication node for transfer, they are moved to a different directory. When the copy to the communication node is complete, they are "Archive" files and moved to the final directory. Archive files are automatically deleted from the test cell node after 30 days by the `cron` utility.

### 2.4 PAM Data File Acknowledgement

The process of sending a Datapoint to the PAM database includes an acknowledgement that indicates the success/failure of the transfer process. The success/failure of the transfer process is indicated by setting the CyFlex variable `NOTIFY` on the originating test cell. Specifying this CyFlex variable in a display group will provide visual feedback of the Datapoint transfer status.



### 3 Specification and Keyword Files

Specifications for PAM data are kept on the test cell node in the PAM Specs file `/specs/pam_specs.n`, where  $n$  = test cell #. Like other CyFlex specs files, it is an editable text file, divided into logical blocks by lines of the form `$KEYWORD`. The various PAM Specs translators read one or more of these blocks. Refer to *Appendix A. Sample PAM Specification File* on page 12.

Each block has a specific format which must be maintained. However, comment lines may be added for clarity or documentation with an initial `#`-character. Unused fields are set to `"-"` to hold place.

The following terms are used in these notes:

label	label string for a CyFlex Variable
keyword	PAM Keyword string
units	PAM Units Keyword string
string	User-specified character string
value	User-specified numeric value

The following sub-sections describe the entries in each of the blocks of the PAM Specs file.

#### 3.1 PAM Header Block

The PAM Header block specifies the test ID and the test ID extension:

```
$PAMHeader

# Test ID      test ID extension
  ASSET2000          ctc
```

#### 3.2 PAM Parameter Block

The PAM Parameter block specifies test parameters used in PAM data analyses.

**Note:**

Parameters which are already specified in the Engine Specs file `/specs/engine_specs.n` are automatically included in the PAM Header. They should NOT be specified in this block.

Parameter entries are identified by a PAM Parameter Keyword, such as `AIR_CONF`. They are uppercase and selected from the PAM Parameter Keywords file. Units strings are selected from the PAM Units Keywords file. To obtain a list of valid units keywords enter the command `units`.

```
$PAMParameters

# keyword      value      units
AIR_CONF      3          none
CMP_IN_DIA    5.5        in
...
```

### 3.3 PAM Datapoint Block

The PAM Datapoint block specifies the CyFlex Variables by label that is included in the PAM Datapoint output file. Each entry is a set of space-delimited fields for a CyFlex Variable label and its keywords. Refer to *Appendix A. Sample PAM Specification File* on page 12.

Labels are assigned to one or more keywords from the PAM Channel Keywords file according to the following rules:

1. A label may be assigned to 1 - 8 PAM Keywords
2. If a label is assigned to one PAM Keyword that is a case-insensitive copy, then no keyword field is needed
3. If a label is assigned to more than one PAM Keyword, then all of its keyword fields must be included
4. If a label is for a Snapshot channel, then the first keyword field must be SNAP (required)
5. If a label is for a Fuel Reading channel, then the first keyword field may be AVER (optional and redundant)

#### 3.3.1 PAM Keyword Files

PAM identifies its various channels, parameters and units through standardized keywords maintained in files on each test cell node:

*Table 1: PAM Keywords and Files*

PAM Keywords	Keywords File
Channel	/cyflex/channel.dat
Parameter	/cyflex/parameter.dat
Units	/cyflex/units.dat

PAM Keywords used in the PAM Specs file must be selected from these files. For more information on PAM keywords go to the PAM home page on the Cummins intranet. Both `pam_specs` and `datapoint` will check keywords as they process `/specs/pam_specs.n`, skip invalid ones and log the errors. The reported errors will also be emailed to addresses contained in the CyFlex variable `user_email`. Refer to *Section 5.3 datapoint* on page 7 and *Section 5.4 pam\_specs* on page 8.

### 3.4 PAM Datapoint Log File

The current PAM Datapoint, i.e., the number for the most recent PAM Datapoint completed at the test cell, is logged in the PAM Datapoint Log file `/specs/PAM_datapoint`. This file is used only to initialize the current PAM Datapoint at system startup.

**Note:**

Changing this file alone does NOT change the current PAM Datapoint, either on the display or for further PAM Datapoint files.

## 4 Operating Tips

In general, making changes to the current PAM Datapoint specifications involves changing the PAM Specs file `/specs/pam_specs.n` and then notifying the CyFlex system that the PAM Specs have changed. Refer to *Appendix A. Sample PAM Specification File* on page 12.

### 4.1 Changing a Channel for a PAM Datapoint

1. Edit the Channel entry in the `$PAMDatapoint` block of the PAM Specs file

```
/specs/pam_specs.n.
```

```
$PAMDatapoint
```

```
label      keyword(s)
```

```
( find the desired label in this section and make the  
  modifications. When complete, save the file)
```

2. Translate the PAM Specs file by entering the following command on the command line:

```
pam_specs
```

A Channel entry may also be added or deleted in these blocks.

### 4.2 Changing a Parameter for a PAM Header

When the test environment or engine parameters change, often when a new engine goes into a test cell, change PAM Parameters accordingly:

- Change any parameters in Engine Specs file `/specs/engine_specs.n`. All of the Engine Specs parameters are automatically included in the PAM Header file
  - a. Edit the Parameter entry in the `$PAMParameters` block of the PAM Specs file `/specs/pam_specs.n`. When complete, save the file.
  - b. Translate the PAM Specs file by entering the following command on the command line:

```
pam_specs
```

A Parameter entry may also be added or deleted in any of these blocks.

### 4.3 Changing a PAM Test ID

A new test should be the occasion for a new PAM Test ID to group similar PAM Datapoints for future reference. To change the Test ID or to re-activate an existing Test ID, use the command `setPAMtestid`. Refer to *Section 5.2 setPAMtestid* on page 7.

## 5 PAM Programs

CyFlex maintains several programs to conduct PAM data collection and transfer. They run essentially without user intervention once the CyFlex system has been started. The following lists several utilities that are associated with the PAM database.

### 5.1 setPAMpoint

This command sets the effective PAM Datapoint to a specified datapoint, the integer number for the next datapoint under the current PAM TestID. The datapoint number is determined when the command `setPAMtestid` is used. Refer to [cyflex.com](http://cyflex.com) usage help for [setPAMpoint](#) for supplemental information.

Example:

```
setPAMpoint 1997
```

sets the next PAM Datapoint to 1997 for the current PAM TestID.

### 5.2 setPAMtestid

Syntax:

```
setPAMtestid {testid} [{datapoint}] [{ext}]
```

This command changes the PAM TestID string to a specified `{testid}` and by default resets the effective PAM Datapoint to 1 for a new TestID.

- If an integer `{datapoint}` argument is included, then it is the NEXT PAM Datapoint number that will be assigned to `{testid}`.
- If `{testid}` is an existing PAM TestID, the next datapoint number is automatically determined from existing information.

The argument `{id_extension}` is optional but if present it is the extension that is added to the TestID to make up the complete PAM TestID. The default extension is specified in the `$PAMheader` section of the `pam_specs.n` file.

Refer to [cyflex.com](http://cyflex.com) usage help for [setPAMtestid](#) for details on this command.

### 5.3 datapoint

The Fuel Reading program `datapoint` runs continuously at the CyFlex test cell to acquire PAM and statistical data on Fuel Readings. It waits for Fuel Reading start and stop events sent from any one of a variety of "collector" tasks which monitor fuel measurement hardware. When a Fuel Reading is finished, `datapoint` writes a PAM Datapoint file in `/data/transfer/PAM_datapoint/`.

`datapoint` is started in the background in the `/cell/go.scp` startup script:

```
datapoint &
```

When changes are made to the PAM Specs file `/specs/pam_specs.n`, it is necessary to notify `datapoint` from the command line to reconfigure. Accomplish this by entering the `pam_specs` command: Refer to *Section 5.4 pam\_specs* on page 8.

When `datapoint` starts or reconfigures, it reads the `$PAMDatapoint` block of PAM Specs. Each entry is an assignment of a CyFlex Variable to 1-8 PAM Keywords to be included in subsequent PAM Datapoint files. Refer to [cyflex.com](http://cyflex.com) usage help for [datapoint](#) for supplemental information.

### 5.4 pam\_specs

The `pam_specs` program is the translator for the PAM Specs file `/specs/pam_specs.n`. It can be run from the command line after any change to PAM Specs.

`pam_specs` causes a PAM Header file to be written and it notifies `datapoint` to read the `$PAMDatapoint` assignments in PAM Specs for later PAM Datapoint files.

### 5.5 tranMove

The `tranMove` program runs continuously at the CyFlex test cell. It copies new PAM Header and PAM Datapoint files to a CyFlex communications node for transfer to the PAM/Oracle network server. It also migrates new PAM files from the Transfer directory to the Archive directory, as outlined in *Section 2.3 PAM Data File Transfer* on page 2.

The program follows instructions for the copy and move specified in the Transfer Specs file `/specs/transfer_specs`:

```
# transfer node
@DESTINATION
# IP or Alias      account    pwd
ctclfnode9       tcp       xnq

# Transfer Type
PAM Datapoint

# Source: Ready                                Complete
/data/transfer/PAM_datapoint/                 /data/PAM_datapoint/

# Destination:
# Node          Path
$NODE          /data/PAMdatapoint/ready/$TC/
$
```

This sample entry for PAM Datapoints specifies the local move from `/data/transfer/PAM_datapoint/` to `/data/PAM_datapoint/` on the test cell node.

It also specifies the copy to the CyFlex communication node. When `$NODE` is specified as the first entry of the destination node, the IP that is specified under the `@DESTINATION` keyword is used as the destination node. In addition, the `account/pwd` that is specified on the same line is the used to perform the copy to the `Ready` directory reserved for this test cell. The directory, under the `Ready` directory, that will receive the file is specified via `$TC`, which is replaced with the test cell number of the local node.

Refer to [cyflex.com](http://cyflex.com) usage help for [tranMove](#) for supplemental information.

## 5.6 extdatman

The program `extdatman` runs on the communication node and is responsible for transferring the data files to the PAM/Oracle database. It is always running on the communication node and periodically checks the various `Ready` directories to determine if `tranMove` has placed any files there that need to be transferred.

## 5.7 pamResend

The `pamResend` program causes data files to be resent to the PAM/Oracle database. It accomplishes this by moving any files in the `Hold` directory to `Ready` on the communication node. Once the files are in the `Ready` directory, the `extdatman` program will resend them to the PAM/Oracle database. This program is normally run on the communication node by the site administrator.

Refer to [cyflex.com](http://cyflex.com) usage help for [pamResend](#) for details on this command.

## 5.8 pamResendSite

It is possible that the communication node for a given `site` resides on a different `site`. For example, the communication node for `site BAK` could be located at `site CTC`. There are two requirements for this to be successful:

- The reserved directory for the testcell under the `Ready`, `Hold`, `Complete` directories must not conflict with the existing directories. This means that the testcell number at the source `site` (BAK) must not be the same as any testcell at the destination 'site' (CTC).
- A `site-specific cell_map` file, `cell_map.site` must exist on the destination node.

This program is also normally run on the communication node by the site administrator.

Refer to [cyflex.com](http://cyflex.com) usage help for [pamResendSite](#) for details on this command.

## 6 Datapoint Calculations for Engine Performance

### 6.1 Flow Rate Calculations

The average values of air and fuel flow for a datapoint value are obtained from the `stat_child` module launched by [datapoint](#) at the end of a fuel reading interval. If the averaging process is not successful, the instantaneous values at the end of a reading are used.

Appropriate units conversion is performed and the changes of values for the flows are checked against the specified tolerances for data compression. A "delta" message is sent to the data compression manager if required.

The computations for fuel rate depend on the type of device that is used to control it. For devices that give a direct reading of fuel rate (flowtron for example), the instantaneous fuel rate value is averaged over a predetermined interval by the [fa\\_total](#) program and the weight consumed over the interval is computed at the end of the reading. For fuel weighing systems like [lc\\_scales](#) and [cf\\_scales](#), the computation is done in device-specific modules.

All of the `FR_*` fuel reading variables are created by the `perf_labels` program. It processes the system file `/cyflex/perf_labels` which contains information required to create frequently used performance parameters.

Both the `lc_scales` and `cf_scales` programs which are most commonly used to control fuel flow devices, set the values of `FW_wt` - the fuel weight consumed, `FR_time` - the length of a fuel reading, `FR_Fuel_rate` - calculated as `FR_wt/FR_time`, and `FR_BSFC` at the end of a fuel reading.

Continuous calculation of air mass and volume flows is done with one or more of several different programs, each working with a different type of device. They include:

- LFE and LFE\_UCC
- Venturi
- Critical flow venturi and `cfv_1065`
- Subsonic venturi
- Vcone

Other devices are added as they are purchased by TRP Laboratories or customers.

Multiple instantaneous air and fuel flow readings are summed by `fa_total`.

### 6.2 Fuel Air Ratio Calculations

The computations for fuel/air ratio are performed in a units-independent manner so that the results do not depend on the particular units of measurement of either fuel rate or air flow rate.

The equivalence ratio is calculated as fuel/air ratio divided by the stoichiometric fuel/air ratio, that is, the fuel/air ratio where all available oxygen is used.

Storage labels for this information are specified in `/cell/perf_labels` and instantaneous values for both are normally calculated by the [fa\\_total](#) program which will sum the flow from one or more air flow meters. The properties of the fuel are stipulated by [init\\_properties](#) and [init\\_composition](#) programs, each having their own specification file.

### 6.3 Volumetric Efficiency

Volumetric efficiency is calculated as the intake volume flow rate, based on manifold pressure and temperature, divided by the volume displacement rate of the engine. It is continuously calculated by the [volef](#) program and sampled by the [stat\\_child](#) process. If statistical sampling is successful, the average value is used. Otherwise the value at the end of the fuel reading is used.

The [volef](#) program is configured with a specification file to enable direct of the program to use the right intake conditions and number of cylinders. It gets engine speed from the variable for performance calculations configured with the `/specs/spcl_chans` file. The total engine displacement rate is configured with values from `/specs/engine_specs` file. Refer to [Computing Volumetric Efficiency](#) for related information.

### 6.4 Performance Calculations

If the [stat\\_child](#) program is successful, the mean values of engine speed and dyno speed will be extracted from the average results. Otherwise, the instantaneous values at the end of the fuel reading will be used in computations.

The following values are computed:

- Average speed  
Using values calculated by [comp\\_perf](#) with labels specified in `/specs/spcl_chans`
- Average usable torque  
Using values calculated by [comp\\_ctrl](#) if the old control program is running and [eng\\_ctrl\\_task](#) if the new one is used using labels specified in `/specs/spcl_chans`
- Average in-cylinder pressure that results in usable work (BMEP)  
Using values calculated by [comp\\_ctrl](#) if the old control program is running and [eng\\_ctrl\\_task](#) if the new one is used using labels specified in `/specs/spcl_chans`
- Brake (usable) horse power (BHP)  
Using values calculated by [comp\\_perf](#)
- BSFC (fuel rate divided by BHP)  
Calculated in datapoint using values supplied by [fa\\_total](#)
- Fuel mass/cylinder stroke  
Calculated in [datapoint](#) from fuel mass (above) and engine speed (above)
- Fuel volume/cylinder stroke  
Calculated in datapoint from fuel mass (above) and fuel density which is specified in `/cell/perf_labels`



## Appendix A. Sample PAM Specification File

```

#*****
#
#       P A M   s p e c s
#
#*****
#
#   PAM Specs assign Cyflex Variables and parameter values to keywords
#   reserved for the Performance Analysis Module (PAM) Database.
#
#   PAM is maintained on the PAM/Oracle network server.
#
#   The Pam Specs file must be /specs/pam_specs.N, where N is the integer
#   test cell number maintained in /cell/test_cell_number.
#
#   PAM Specs replaces the /specs/data_specs.NNN specs file.
#
#*****

```

Version 1.0

```

#*****
#
#   PAM Header
#
#*****
#
#   This section specifies the PAM TestID and the parameters that
#   identify the test cell location to the PAM Database:
#
#       TestID:  unique 1-24 character Test Identification string
#                 allowed: (letters) (numbers) - _
#                 disallowed: ! (space)
#
#       id extension - the string that is concatenated to the TestID to
#       make up the complete Test ID. The prefix `!' is
#       added to this string.
#
#*****

```

\$PAMHeader

```

# TestID          testid extension
ASSET2000        ctc

```

```

#*****
#
#   PAM Parameters
#
#*****
#
#   This section reports the current values for PAM Parameters:
#
#       Keyword      PAM Parameter Keyword: 1-15 character string

```

```

#           (select from /cyflex/parameter.dat)
#
#   Value   Real/Integer parameter value
#
#   Units   PAM Units Keyword: 1-15 character string
#           (select from /cyflex/units.dat)
#
#   They are sent to the PAM Database only when the "PAMheader" program
#   runs.
#
#   All entries in Cyflex Engine Constants are included automatically.
#   They are maintained in /specs/engine_specs.N.
#
#*****

```

\$PAMParameters

Keyword	Value	Units
AIR_CONF	3	none
CMP_IN_DIA	5.5	in
CMP_OT_DIA	3.0	in
TUR_IN_DIA	3.0	in
TUR_OT_DIA	5.5	in
CLNT_MTR0_SN	86963	integer
TRB_LTYP	1	integer
TRB_LFAC	0	integer
CLNT_CONF	1	none
AEG	0.5	none
CL_TYP	1	integer
CL_TYP_MTR1	0	integer

```

#*****
#
#   PAM Datapoint
#
#*****#
#   This section identifies the Cyflex Variables by label that are sent
#   to the PAM Database when each Fuel Reading finishes.  It also
#   assigns PAM Keywords to the label, to identify its function to PAM.
#
#   Label       Cyflex Variable label: 1-15 character string
#               (specified in other Cyflex specs files)
#
#   Averaging   = (blank) value averaged over Fuel Reading [default]
#               = AVER    averaged, same as blank
#               = SNAP    "snapshot" value at end of Fuel Reading
#
#   Keywords    1-8 PAM Keywords: 1-15 character strings, blank
#               delimited
#               (select from /cyflex/channel.dat)
#
#   If Averaging field is blank, then averaging (AVER) is assumed.
#
#   If Keywords fields are blank, then the uppercase version of the
#   Label field is substituted automatically as the only PAM Keyword.

```

```

#       If any Keywords fields are used, then they are the only PAM Keywords
#       for that label.
#
#       If invalid PAM Keywords are used, an entry in the Error Log is made
#       and the value is excluded from the PAM Datapoint.
#
#       One of the following extensions may be appended to PAM Keywords, to
#       indicate a measurement location:
#
#       @1 - @99      user
#
#       @L      left           @R      right
#       @LF     left front     @RF     right front
#       @LR     left rear      @RR     right read
#
#       Run the "startup" command after any changes to this section.  It
#       notifies the datapoint task that a change has occurred.
#
#*****

```

\$PAMDatapoint

```

# Label      Averaging      Keywords (1-4)
#              (AVER/SNAP)

```

# Analog Input Channels

# Pressures

```

air_mtro_p      AVER
cmp_in_p        AVER
cmp_ot_p        AVER
int_mnf_p       AVER
tur_fi_p        AVER
blk_in_p        AVER
tur_ri_p        AVER      TUR_IN_P
exh_stk_p       AVER
blow_by         AVER
fpmp_in_p       AVER
rail_p          AVER
oilrfl_p        AVER
clnt_ot_p       AVER

```

# Temperatures

```

air_mtr0_t      AVER
cmp_in_t        AVER      CMP_IN_T      SAE_REF_T
cmp_ot_t        AVER
int_mnf_t       AVER
exh_prt1_t      AVER
exh_prt2_t      AVER
exh_prt3_t      AVER
exh_prt4_t      AVER
exh_prt5_t      AVER
exh_prt6_t      AVER
tur_fi_t        AVER

```

```

tur_ri_t      AVER
fpmp_in_t     AVER
oilrfl_t      AVER
oilpan_t      AVER
blk_in_t      AVER
clnt_ot_t     AVER      CLNT_OT_T      CLNT_MTR0_T
clnt_in_t     AVER
ac_w_in_t     AVER
ac_w_ot_t     AVER      AC_W_OT_T      CLNT_MTR1_T
test_cell_t   SNAP

```

# Other

```

RPM           AVER      ANA_SPD
TORQUE        AVER      ANA_TORQ
Throttle_CM   AVER      MAN_THT_CTL
pier_mean     SNAP      PIER_SMK

```

# Fuel Reading  
 # (use SNAP for labels with "FR\_" prefix)

```

FR_Fuel_rate  SNAP      FUEL_RATE
FR_time       SNAP      ET
FR_wt         SNAP      FUEL_WT

```

# Frequency Input Channels

```

SPEED         AVER      FREQ_SPD
air_mtr0_vf   AVER
trb_spd       AVER
clnt_mtr0_op  AVER

```

# Engine Performance Channels

```

engine_hrs    AVER      ENG_ET
test_hrs      AVER      SNAP      TEST_TM
vap_pa        AVER
cel_opac      AVER

```

# General Use Variables

\$End