CyFlex® Continuous Fuel Measurement User Guide

Version 6

February 21, 2024

Developed by Transportation Laboratories

Version History

Version	Date	Revision Description
1	1/25/2016	Initial publication
2	8/23/2016	Format with SGS brand
3	4/7/2020	Retrofit to new template
4	12/9/2021	Revised *Section 2 Using Continuous Flow Measurement* on page 3 to remove inline `cf_scales` usage content and add hypertext linked cross-reference to its usage help on cyflex.com.
5	6/14/2022	Updated hypertext linked cross-reference to cyflex.com usage help description for `cf_scales` in *Section 2 Using Continuous Flow Measurement* on page 3
6	2/21/2024	Rebrand to TRP Laboratories

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

List of Figures

1 Overview

The CyFlex ® `cf_scales` command accurately measures fuel consumption for diesel engines that use only a portion of the fuel that is delivered to them. The command addresses the problem for when the return fuel has been aerated by engine. When this happens, the return fuel must be captured in a vented vessel and given sufficient time in the vessel for the small air bubbles to escape. If any air remains in the fuel when it is sent back to the engine, there is the possibility that it will come out of suspension and accumulate in the fuel supply lines and/or cause cavitation at the pump inlet. This accumulation of air will also distort the apparent volume flow rates over time.

In addition to aeration, the return fuel has normally been heated to a varying degree and must be cooled before it is sent back to the engine.

The `cf_scales` command accounts for the amount of fuel returned to the vessel along with the fresh fuel that is provided as makeup for the fuel consumed by the engine by addressing:

- Accurate measurement of mass flow
- Measurement over fixed and arbitrary time intervals
- Temperature control

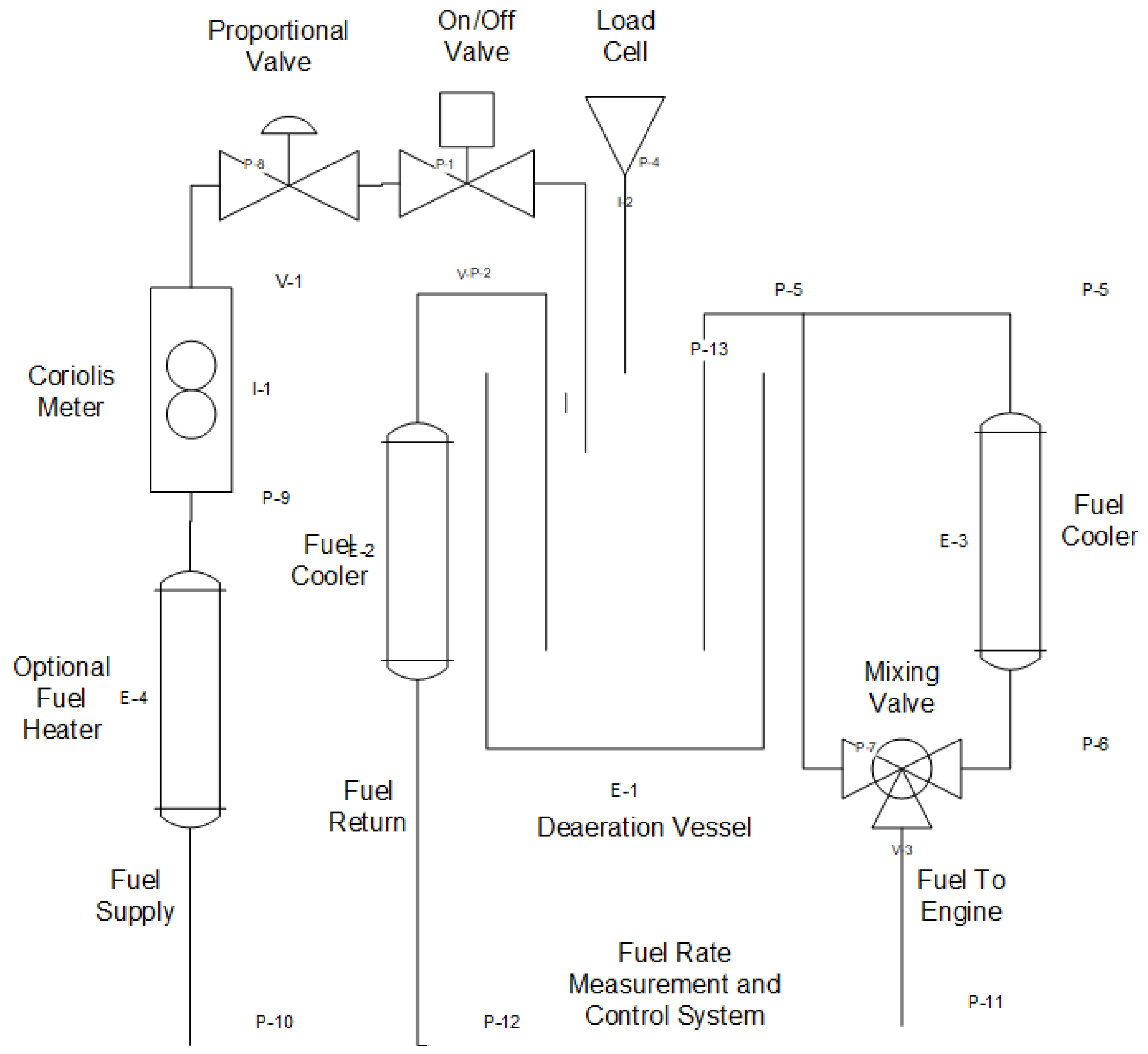
Refer to *Section 2 Using Continuous Flow Measurement* on page 3 for additional information.

1.1 Continuous Fuel Measurement System Overview

As in *Figure 1* on page 2, the system consists of the following components:

- A vented deaeration vessel (bucket) to capture the return flow
- A load cell to measure the bucket weight
- A single Coriolis meter to measure make up flow into the bucket
- A proportional control valve to maintain a relatively constant weight of fuel in the bucket under certain modes of operation
- An on/off valve that is used to fill the bucket under other modes of operation
- A heat exchanger to cool the return flow
- A float switch to shut off the fuel supply in case of failure of the fill or control valves
- A temperature control system that will vary with the type of engine under test

Figure 1: Fuel Measurement System Elements and Flow



2 Using Continuous Flow Measurement

Enter `cf_scales` to invoke the command.

Refer to `cf_scales` usage help on cyflex.com for command option details.

2.1 Modes of Operation

Set the value of a CyFlex integer variable to select among five different modes of operation This can be done manually, in a command shell script, or during a test procedure. The mode can be changed at any time.

A standard CyFlex control loop is commanded in various ways by the program and the user is responsible for setting up this loop. The fuel measurement system will put the control loop into the proper mode depending on the selections.

2.1.1 Mode One: Continuous Estimate of Fuel Mass Flow Rate

Mode one is where a continuous estimate of fuel mass flow rate is desired. In this mode, the controller is put into closed loop operation with a weight set point that has been configured by the user. The on/off (fill) valve is held open and fuel rate is calculated by changes in output of the Coriolis meter and adjusted by changes in weight of the bucket. No filtering is added by the `cf_scales` program.

$$\text{FuelRate} = \frac{(\text{CoriolisWeight}_i - \text{CoriolisWeight}_{i-1} + \text{BucketWeight}_{i-1} - \text{BucketWeight}_I)}{(\text{Time}_I - \text{Time}_{i-1})}$$

All weight readings of the bucket are adjusted by the buoyancy correction factor which accounts for any tubes, pipes, of thermocouples that are inserted into the fuel. Refer to *Section 3 Buoyancy Correction* on page 5.

Filter the signal from the load cell measuring the bucket weight with an appropriate hardware filter to minimize noise. Because there can be residual noise on the load cell signal, the instantaneous calculation can also be noisy so filter the fuel rate value with a computed user-created expression of the or pipe it into a CyFlex running average variable.

The amount of noise can be reduced by increasing the execution interval, thus increasing the change in value of the Coriolis meter, increasing the denominator of the above equation, and reducing the importance of changes in the bucket weight. Also, modest filtering should be applied to the weight value that is read from the Coriolis meter because, without filtering, the value will change in discrete steps according to the setup of the meter.

2.1.2 Mode Two: Conventional Load Cell Scales Behavior

In mode two, the system behaves much like the conventional load cell scales program. The control valve is set to open loop at 100% command and the bucket automatically refills whenever a fuel reading is not in progress. Start fuel readings with standard CyFlex commands or test procedures. End fuel readings when a specified time interval has expired. The fuel rate is calculated as the RMS value of the first derivative of the bucket weight, which is corrected for buoyancy.

$$\text{Fuel Rate} = (\text{BucketWeight}_{i-1} - \text{BucketWeight}_I) / (\text{Time}_I - \text{Time}_{i-1})$$

2.1.3 Mode Three: Arbitrary Length Fuel Readings

Use mode three when arbitrary length fuel readings are to be taken. A reading is started via an event, as in mode two, but the reading does not end after a prescribed time interval. Instead, another specified event must be set by a test procedure to end the reading.

This mode is applicable for transient tests where the engine power varies and the average fuel rate is not constant. In this case, the specified fuel weight variable gives the amount of fuel consumed since the start of the reading.

$$\text{Weight}=\text{CoriolisWeight i} - \text{CoriolisWeight 0} - \text{BucketWeight i} + \text{BucketWeight}_0$$

Bucket weights are buoyancy corrected.

2.1.4 Mode Four: Alternate between Mode One and Mode Two

In mode four the system alternates between modes one and two. When a fuel reading is not in progress, the bucket weight is maintained constant with the proportional valve. When a reading is requested, the system behaves as a load cell scales.

2.1.5 Mode Five: Constant Bucket Weight

In mode five, the bucket weight is always held constant. When a reading is requested, a timer is started and the fuel reading is ended after a user specified time interval. The fuel rate is the slope of the RMS value of the following:

$$\text{Weight}=\text{CoriolisWeight}_i - \text{CoriolisWeight}_0 - \text{BucketWeight}_i + \text{BucketWeight}_0$$

Bucket weights are buoyancy corrected.

This mode provides the best fuel temperature control at a given speed and load because all flows into and out of the system remain constant rather than turning on and off.

2.2 Additional Information

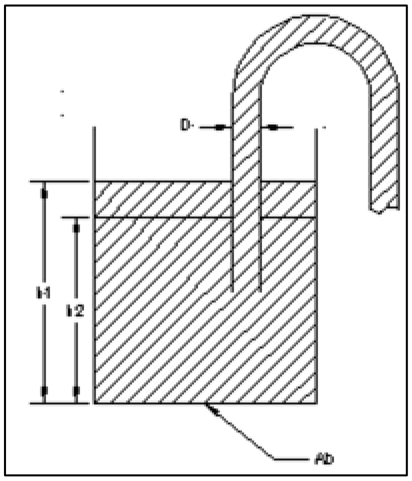
Whenever the engine is not running, the system commands the fill valve to close and the controller is put into open loop with zero commanded output.

If the system is in mode three, the engine can stop and restart during a fuel reading or test cycle without losing track of the fuel that is being consumed. This is especially useful for testing on CyberApps or other transient cycles where the test procedure calls for the engine to stop in the middle of the cycle.

3 Buoyancy Correction

Consider the bucket of fuel shown in *Figure 2*.

Figure 2: Bucket of Fuel



To perform a static analysis, assume a suction tube of diameter, D, is immersed in the fuel to a depth well below the minimum fuel height in the bucket and that the flow through the tube is negligible. Prior to a fuel reading, the fuel height above the bottom of the bucket is h_1. After the fuel reading, the fuel height is h_2. Assume that the bucket has vertical sides and the bucket cross-sectional area is A_b. The fuel pressure, P, at a depth, *h*, is given by

$$P = \rho g h \qquad \text{fb.1}$$

where

ρ is the fuel density and *g* is the local gravitational acceleration.

The vertical force exerted by the fuel on the bucket, *F*, is the fuel pressure at the bottom of the bucket times the bucket area.

$$F = PA_b = \rho g h A_b \qquad \text{fb.2}$$

The observed change in force, ΔF_{obs}, between the beginning and end of the fuel reading would be:

$$\Delta F_{obs} = F_1 - F_2 = \rho g (h_1 - h_2) A_b \qquad \text{fb.3}$$

To convert this observed change in force to change in mass, Δm_{obs}, divide by the gravitational acceleration:

$$\Delta m_{obs} = \frac{\Delta F_{obs}}{g} = \rho (h_1 - h_2) A_b \qquad \text{fb.4}$$

If one assumes that the suction tube starts and ends the fuel reading completely full of fuel, then the actual volume of fuel consumed during the fuel reading is the doughnut shaped volume, ΔV, given by

$$\Delta V = \left(A_b - \frac{\pi}{4} D^2 \right)(h_1 - h_2) = (A_b - A_t)(h_1 - h_2) \qquad \texttt{fb.5}$$

The actual change in mass over the fuel reading, Δm_{act}, is then

$$\Delta m_{act} = \rho \Delta V = \rho (A_b - A_t)(h_1 - h_2) \qquad \texttt{fb.6}$$

Therefore, the ratio of the actual amount of fuel consumed to the observed fuel consumption is

$$\frac{\Delta m_{act}}{\Delta m_{obs}} = \frac{\rho (A_b - A_t)(h_1 - h_2)}{\rho A_b (h_1 - h_2)} = \frac{A_b - A_t}{A_b} \qquad \texttt{fb.7}$$

This is the buoyancy correction factor.

The fact that the buoyancy correction factor is always less than one is an indication that the tube exerts a downward force on the bucket that lessens as fuel is consumed.

One can come up with the same answer using Archimedes Principle which says that the buoyancy force is equal to the weight of the displaced fluid. Referring to *Figure 2* on page 5, the force on the fuel is equal and opposite to the buoyancy force on the tube and the change from the beginning to the end of the fuel reading would be given by:

$$\Delta F_{buoy} = \rho g A_t (d_1 - d_2) = \rho g A_t (h_1 - h_2) \qquad \texttt{fb.8}$$

The change in force exerted by the fuel would be:

$$\Delta F_{fuel} = \rho g (V_1 - V_2) = \rho g (A_b - A_t)(h_1 - h_2) \qquad \texttt{fb.9}$$

Combining these two forces and dividing by the gravitational constant to get the observed change in mass gives:

$$\Delta m_{obs} = \frac{\Delta F_{fuel} + \Delta F_{buoy}}{g} = \rho (A_b - A_t)(h_1 - h_2) + \rho A_t (h_1 - h_2) = \rho A_b (h_1 - h_2) \qquad \texttt{fb.10}$$

The ratio of actual change in fuel mass to observed change is again:

$$\frac{\Delta m_{act}}{\Delta m_{obs}} = \frac{\rho (A_b - A_t)(h_1 - h_2)}{\rho A_b (h_1 - h_2)} = \frac{A_b - A_t}{A_b} \qquad \texttt{fb.11}$$

which is the same buoyancy correction factor that was calculated using fuel pressure

The amount of fuel being held up by the sub-atmospheric pressure in the tube above the fuel surface increases as the fuel level falls. This fuel is being held up by the tube and not the bucket. If the tube is removed, the fuel would go with it as though the fuel and tube were a solid object.

The above analysis considers only the immersed suction tube. The cross-sectional area of other objects such as thermocouples that also penetrate the fuel surface must also be added to the suction tube area. These objects should be of constant cross-sectional area over their entire submerged length and should remain submerged at the lowest fuel level expected during a fuel reading.

Appendix A. Example Specification File

```
#################################################################
# This is a specification file for a flow rate measurement      #
# system which is managed by the task "cf_scales"               #
#                                                               #
# The "cf_scales" task can be used to measure mass flow that    #
# is a combination of a continuous mass output device and       #
# a load cell scales.  A proportional control valve is used to  #
# maintain a constant level in the fuel bucket.  Any deviation  #
# from a constant bucket weight is accounted for by using the   #
# load cell mass as a correction.  The fuel rate can be updated #
# continuously, providing input for a running average variable  #
#                                                               #
# The fuel weight control is accomplished with the standard     #
# CyFlex control task.                                          #
#                                                               #
# Five distinct modes are possible with this task               #
#                                                               #
#  1. Mode 1 is the default mode.  It has a continuous output   #
#     and behaves as described above.  The fill valve is left   #
#     open and the bucket weight control loop is in closed loop.#
#     This mode is not useful for taking fuel readings.         #
#                                                               #
#  2. In Mode 2, the system behaves much like a standard load   #
#     cell scales system.  The bucket weight control loop is    #
#     put into open-loop and the fill valve is closed during    #
#     the fuel reading.  The fill valve will open and close     #
#     to keep the bucket near full until a fuel reading is      #
#     requested via an event specified in this file             #
#                                                               #
#  3. In Mode 3, the system behaves much like mode 1 with       #
#     the addition that external events can be used to start    #
#     and stop a fuel reading over an arbitrary length of time. #
#     This mode of operation should be used when running        #
#     any type of transient test cycle.                         #
#                                                               #
#  4. In mode four, a control loop is used to maintain a        #
#     "constant" weight in the bucket.  Signal events are used  #
#     to trigger the starting and stopping of a fuel reading.   #
#     When a fuel reading is started, the controller will       #
#     be placed in open loop, the fill valve will close, and a  #
#     fuel reading will begin after the stability requirements  #
#     are met.  The reading will be processed as if the system  #
#     were an LC_scales.  However, at the end of the fuel       #
#     reading, the system will be placed back into level        #
#     control. The results will be displayed as they are for    #
#     standard fuel readings.  This mode is most useful when    #
#     the return fuel flowrate is minimal and/or near the       #
#     supply temperature so that temperature disturbances       #
#     within the bucket are minimized.                          #
#                                                               #
#  5. In mode five, the system attempts to maintain a constant  #
#     weight in the fuel bucket and the fill valve remains      #
#     open.  Fuel readings may be taken for steady state        #
#     conditions over a timed interval.  This is the best mode  #
#     to use for engines that have a significant volume of      #
#     return flow from the engine because there is a constant   #
#     flow of both return and makeup flow, allowing the fuel    #
#     temperature control loop to reach equilibrium.            #
#                                                               #
#################################################################


# Only "EMPTYING" devices are supported at this time
#
#------------- data loss event ------------------------------------------
#
#  The name of an event to set for the case where a problem is detected.
#  This problem might be the variable age of the bucket weight, flow
#  meter mass, or other problems with calculations.  The event will be
#  created if it doesn't already exist.  Use a dash for the event name
#  to disable this feature.
#
#  The time out value indicates how long to wait for things to correct
#  themselves before setting the abort event.  If an abort event is
#  used, a timeout value must also be specified.  The value can be any
#  valid CyFlex expression.
#
#  event_name     timeout
   abort_limit    30[sec]

#------------- fill valve label -----------------------------------------
#
# The label of the LOGICAL_VARIABLE used to operate the fuel fill valve
   FMS_FSO_Valve

#------------- bucket weight --------------------------------------------
#
# The label of the REAL_VARIABLE that contains the bucket weight
#
   FMSBktWt

#------------- "end_value" ----------------------------------------------
#
# The bucket weight which will always terminate a sample and will cause
# the fill valve to be turned on.
#
# This field may be a constant, variable, or expression
#
# This is the minimum bucket weight value
   7.5[lb]

#------------- "start_value" --------------------------------------------
#
# The bucket value which will cause the fill valve to be turned off
# and a stability delay to be started
#
# This field may be a constant, variable, or expression
#
# This is the maximum bucket value
```

```
   15[lb]

#------------- "stabilized weight" ---------------------------------------
#
# The stabilization weight - a reading can't start until the bucket value
# has reached this value
#
# This field may be a constant, variable, or expression
#
# The reading will start when the bucket value drops below this value
   14.5[lb]

#------------- stabilization time ----------------------------------------
#
# The minimum time delay for stabilization after the filling process
#
# This field may be a constant, variable, or expression
   5[sec]

#------------- buoyancy correction ---------------------------------------
#
# The buoyancy correction factor for the scales bucket system
#
# This field may only be a constant without units
   0.987783

#------------- power value -----------------------------------------------
#
# The label of a real variable that represents the power value that should
# be used to find specific consumption
#
# power label
   Dyno_power

#------------- mode label ------------------------------------------------
#
# The label of the INTEGER_VARIABLE where the requested operating mode
# will be placed by the user.
#
#  Acceptable values are described above
#
#  label
   FMS_mode

#------------- instantaneous fuel rate label ------------------------------
#
# The label where this program will put the instantaneous fuel rate
# The variable should be the input to a running average variable which
# is specified by the use in the RA specification file
#
#  label
   I_FuelRate

#------------- target time -----------------------------------------------
#
# The label of the REAL_VARIABLE where the requested sample time will be
# found when operating in Mode 2 (lc_scales)
#
# ATTENTION: This label will almost ALWAYS be taken from /cell/cell_special
#            index 20...  probably specified as target_fr_tm
#
#  label
   target_fr_tm

#------------- fuel supply mass ------------------------------------------
#
# The label of the REAL_VARIABLE that contains the total mass that
# has been supplied to the fuel measurement system.  The variable
# must already exist when the cf_scales task is started
#
#  label
   FMSFuelSup

#------------- bucket controller name ------------------------------------
#
# The name of the control loop that is maintaining the fuel bucket weight
#
#  label
   FMSBktWt


###########################################################################
#
#  The following variables are outputs of this program and the user has
#  the option of specifying the variable label or having the program create
#  one.  If a <-> is entered for the variable, it will be created and the
#  registered name that is entered as a spawning argument is prepended to
#  the name listed below.
#
#  InProg     LOGICAL_VARIABLE that is set TRUE when a reading is in progress
#
#  FuelRate   REAL_VARIABLE with units of mass flow (lb/hr)
#
#  SampleSize REAL_VARIABLE with units of mass (lb)
#
#  RunTime    REAL_VARIABLE with units of time (min)
#
#  StdErr     REAL_VARIABLE with units of mass (lb)
#
#  Rsqrd      REAL_VARIABLE with units of none
#
#  SlopeErr   REAL_VARIABLE with units of mass flow (lb/hr)
#
#  _BSFC      REAL_VARIABLE with units of mass flow / time (lb/hp-hr)
#
###########################################################################

#------------- fuel reading in progress logical variable label ------------
#
#  The name of the logical variable where the fuel reading in progress flag
#  will be written - the default label is <reg_name>InProg
#
```

```
# ATTENTION: This label will almost ALWAYS be taken from /cell/logi_specs
#            index 10...  usually specified as FR_in_prg
#  label
     FR_in_prg

#------------- calculated fuel rate  ------------------------------------
#
#  The name of the real variable where the fuel rate will be stored - the
#  default label is <reg_name>FuelRate
#
# ATTENTION: This label will almost ALWAYS be taken from /cell/perf_labels
#            FUEL_READING_DATA index 3...  usually specified as FR_Fuel_rate
#  label
     FR_Fuel_rate

#------------- sample size output -----------------------------------------
#
#  The name of the real variable where the weight of fuel consumed will be
#  stored - the default label is <reg_name>SampleSize
#
# ATTENTION: This label will almost ALWAYS be taken from /cell/perf_labels
#            FUEL_READING_DATA index 15...  usually specified as FR_wt
#  label
     FR_wt

#------------- run time output --------------------------------------------
#
#  The name of the real variable where the length of the fuel reading will
#  be stored - the default label is <reg_name>RunTime
#
# ATTENTION: This label will almost ALWAYS be taken from /cell/perf_labels
#            FUEL_READING_DATA index 16...  usually specified as FR_time
#  label
     FR_time

#------------- standard error output -------------------------------------
#
#  The name of the real variable where the standard error in the calculated
#  fuel rate will be stored - the default label is <reg_name>StdErr
#
#  label
     -

#------------- R squared statistic output --------------------------------
#
#  The name of the real variable where the R squared statistic will be stored
#  - the default label is <reg_name>Rsqrd
#
#  label
     -

#------------- slope error statistic output ------------------------------
#
#  The name of the real variable where the slope error statistic will be
#  stored - the default label is <reg_name>SlopeErr
#
```

```
#  label
     -

#------------- specific consumption output---------------------------------
#
#  The name of the variable where the specific consumption will be stored
#  - the default label is <reg_name>_BSFC
#
#  label
     -

###########################################################################
#
#  The following events are inputs to this program and the user has
#  the option of specifying the event or having the program create
#  one.  If a <-> is entered for the event, it will be created and the
#  registered name that is entered as a spawing argument is appended to
#  the name listed below.
#
#     start_   name of the event that is set by the user to cause
#              a fuel reading to be started
#
#     stop_    name of the event that is set by the user to cause
#              a fuel reading to be ended for modes three and four
#
# ATTENTION: This event will almost ALWAYS be taken from /cell/logi_specs
#            index 11 true event...  usually specified as FR_RQST
#  start_event
      FR_RQST

#  stop_event
      -

#
#  The following label is set true by various fuel reading tasks
#
# ATTENTION:  This label will almost ALWAYS be taken from /cell/logi_specs
#             index 11... usually specified as FR_request
#
#  fuel reading request label
   FR_request


###########################################################################
#
#  The following events are inputs to this program and the user must
#  provide a name.  If the event does not exist when the program is
#  started, the event will be created and a warning will be issued.
#
#     engine_running - name of an event that will be set when
#                      fuel is being consumed
#
#     engine_stopped - name of an event that will be set when
#                      fuel is no longer being consumed
#
#     NOTE - when started, the program will set the bucket controller
#            in open loop and the fill valve closed until the
#            engine_running event is set
#
#
#  ATTENTION: These events are typically found in /cell/logi_specs true and
#              false events for index 8...  usually spd_gt_400 and spd_lt_400
#  engine_running_event
   spd_gt_400

#  engine_stopped_event
   spd_lt_400
```