FMC **Smith Meter Inc.**
An FMC Corporation subsidiary

Electronic Crude Oil Gathering System

TCP-CO

Specifications

Bulletin SS0625P

The *Smith Electronic Crude Oil Gathering System* may consist of three components: a TCP cab-mounted flow computer, an optional transceiver for RF communications, and the routing and dispatch data processing software. The heart of the system is the TCP-CO microprocessor-based flow computer, specifically designed for the custody transfer of crude oil at production lease sites. The optional transceiver transmits data to and from the TCP-RAD and the TCP-CO.

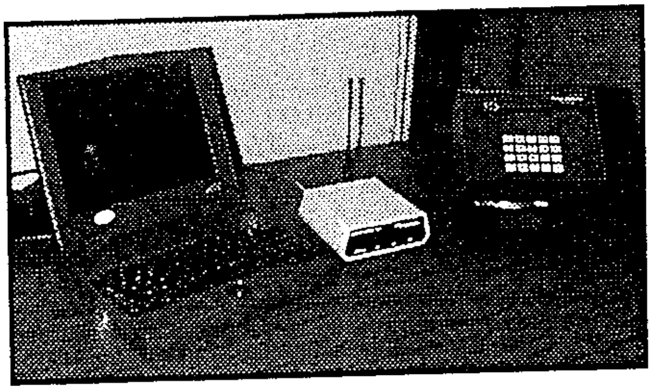
Features

- Improved accuracy by on-board metering
- Automatic temperature compensation
- Automatic sampling
- Automatic BS&W
- Automatic Netting of Load Ticket
- Computerized routing and dispatch
- Instantaneous data recording
- Reduced manpower requirements
- Improved safety

Applications

Typical applications include electronic measurement, data recording, and data transfer of crude oil gathering by truck. State-of-the-art meters combined with these flow computing devices eliminate the errors arising from current manual gaging, reduce loading time, and drastically improve safety. The package will include on-board cab-mounted metering electronics, providing online measurements currently obtained by hand gaging. The measurements include volumes (net and gross), oil, online density, automatic sampling.and BS&W monitoring. The data can be transmitted to an onboard printer or to a host computer. This system eliminates errors from current manual gauging methods, reduces loading time, improves safety and automates data transfer.

The flexible system design may also be applied to fixed truck unloading stations. The load lease information is tracked automatically through the RAD system, providing the possibility of unmanned unloading stations with instaneous transfer of data to an off-site central computer system.



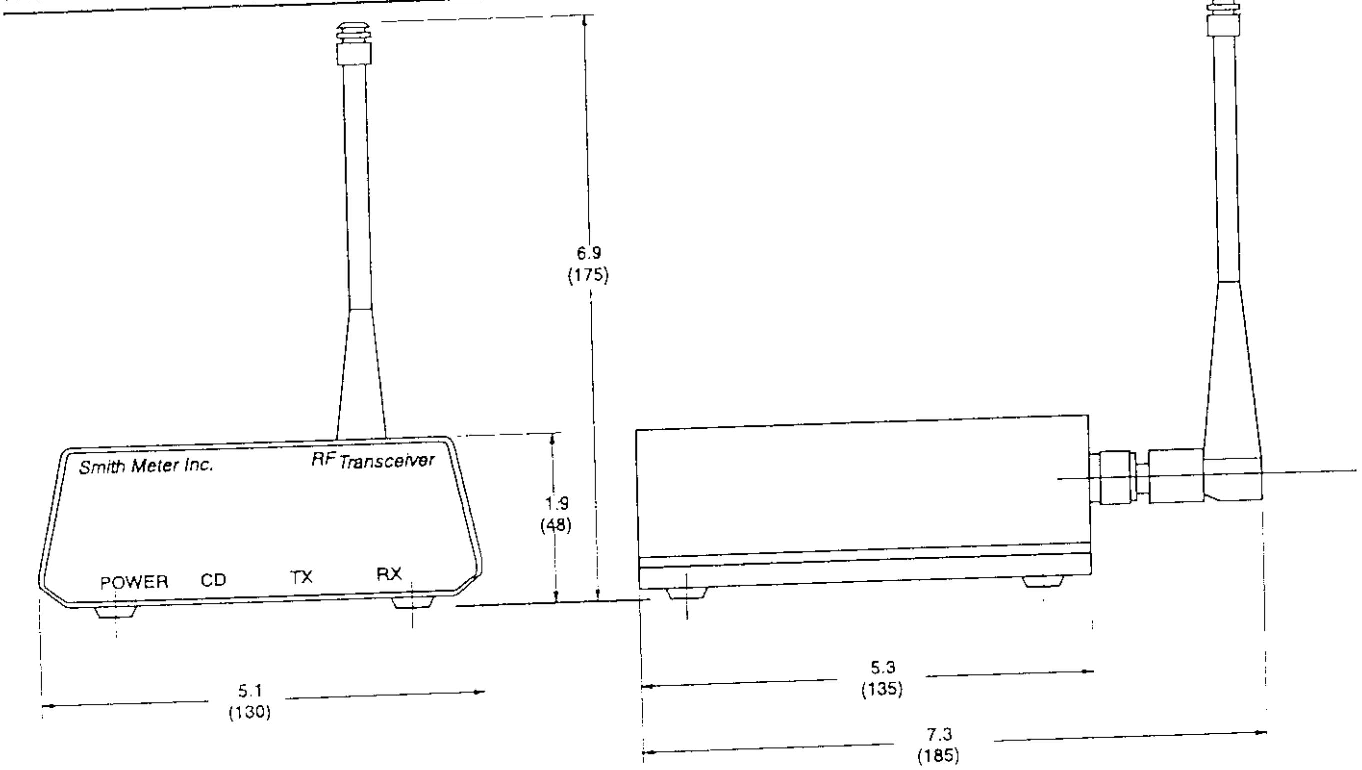
System Operation

The TCP-CO electronic crude oil gathering system is designed to automate the routing, dispatching, and data recording, and improve the accuracy of crude oil gathering. The system includes the electronic equipment, TCP-CO electronic totalizer, TCP-RAD computer program, an optional radio frequency transmitter and receiver, and a meter.

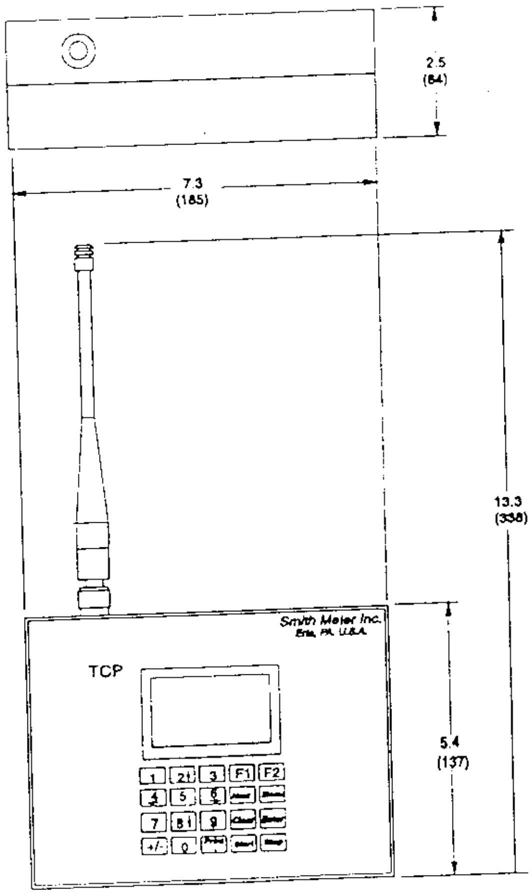
The TCP-CO Electronic Totalizer is designed to be mounted in the cab of a truck. It is wired to the meter, which is mounted on the back of the truck, and either receives pulses from the meter or, in the case of a Smith ST Mass Meter, communicates directly with the meter. The TCP-RAD computer program is installed on a computer in a dispatch office or loaded onto a portable laptop computer. The TCP-RAD and the TCP-CO totalizer will communicate data either via a communications line or through radio frequency communications. The TCP-RAD can be used to program the parameters in the TCP-CO, set up routes and dispatch schedules that are downloaded to the TCP-CO, and gather data from the TCP-CO at the end of a shift or receipt. Once the data has been received from the TCP-CO, the TCP-RAD will store the data in a text file that can be easily transferred to a host system for the purposes of record-keeping, reporting, auditing, etc.

The TCP-RAD is a Windows application designed to streamline and organize the routing and dispatch of tank trucks for crude oil gathering tank collection operations. It is designed to be in operation at the truck terminal, and allow the dispatcher to (1) configure his TCP-CO units, (2) define the scheduled stops for each truck, and (3) retrieve the collected product volume data upon the return of the truck. It will handle the communications interface with the TCP-CO truck-mounted electronics and

Dimensional Diagram



Transceiver



TCP - CO

ON - TRUCK METERING

Application

ude Oil is metered from production storage sites onto trucks to be taken to larger storage sites. The crude oil contains water which can not be metered as oil. The metered volume must be corrected for temperature, density and water content. The oil/water is metered into the truck with a volumetric device, the fluid's temperature is measured at periodic intervals, the water content is determined by centrifuge on periodic samples, and the density is determined by a hydrometer on periodic samples.

At the end of the batch, the average measured temperature, density, and % water content are calculated. The net oil is then determined.

The gross volume, the net volume of oil, the average water content, the average temperature, and the average density are mechanically recorded.

In order to gather samples and to measure the fluid temperature, the driver must get on top of the lease tank. This creates a risk of the driver falling. When the oil is thought to contain H_2S, two people must be present to perform this task in case there is a release and help is needed. The safety risk to personnel and overhead costs have prompted transportation managers to look for alternatives to their current truck loading practices.

he S-Mass Meter (MAY NOW BE KROHNE)

A single coriolis meter can provide accurate mass flow measurement, fluid density and fluid temperature.

Using the meter's mass measurement and its density, **gross volume** can be calculated by the following formula:

$$Q = \text{Mass Rate} \div \text{Density}.$$

Net volume is determined by subtracting the water content and then correcting for temperature based on oil density.

The following benefits should be derived from using a coriolis meter:

(1) A coriolis meter can eliminate the need for the driver getting on the lease tank to take temperature measurements. Thereby, reducing the safety risk and eliminating the overhead associated with the second person in the case of oil containing H_2S.

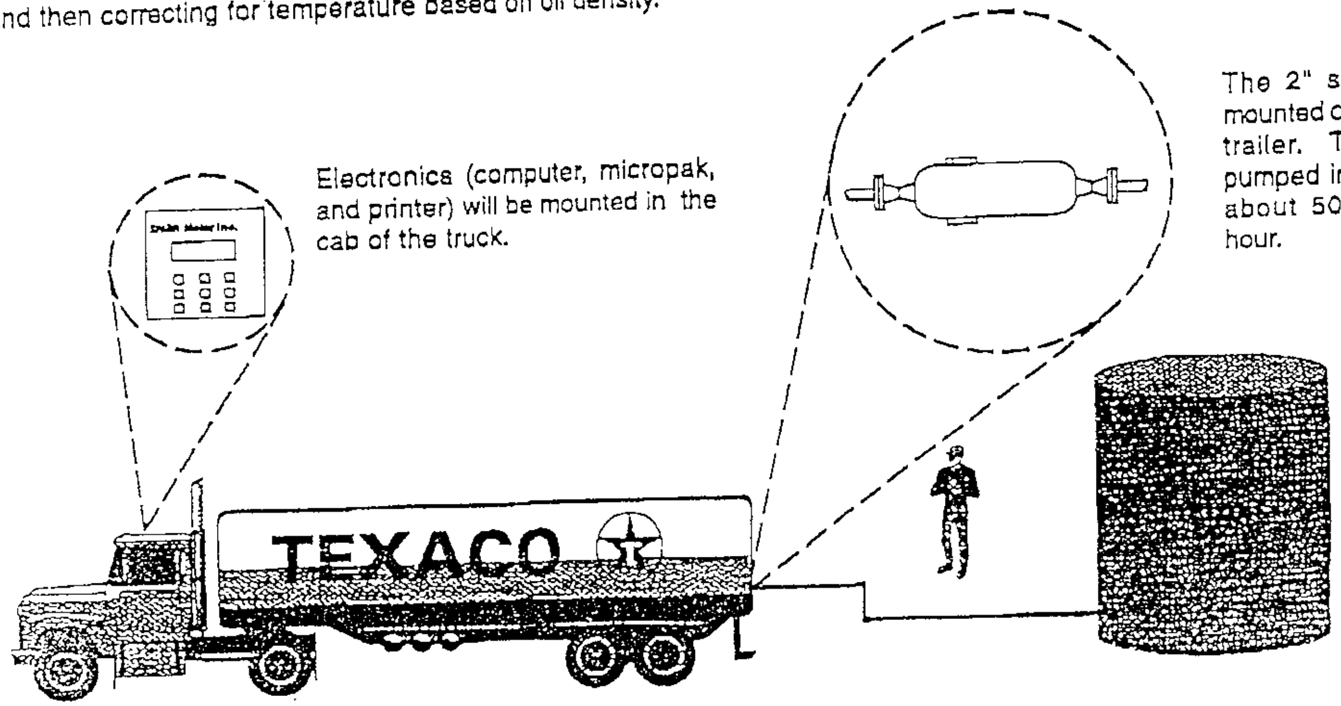
(2) A coriolis meter monitors instantaneous density. This measurement can be used by the meter to detect out of range conditions such as slugs of water or slugs of air. The meter can be configured to ignore its flow measurement under certain conditions.

(3) The meter's ability to measure mass flow allows for the truck to be loaded by weight assuring that the truck is carrying the max allowed load and at the same time the customer can be billed by volume.

(4) A single coriolis meter reduces the instrumentation needed to detect the presence of flow (a flow switch), to provide the measurement of flow, density, and temperature. A coriolis meter reduces initial instrumentation costs, installation costs, and maintenance costs.

System Requirements

- 2" 150# RF flanged Sensor with Micro-Pak
- Flow Computer capable of taking frequency inputs from Micro-Pak and calculating gross volume, net volume of oil, average density, average temperature. The computer should be able to take a live or manual entry for water content.
  The computer should be capable of sending this data to a printer.
- Thermal printer capable of printing a ticket showing the above calculated infomation.



Electronics (computer, micropak, and printer) will be mounted in the cab of the truck.

The 2" sensor will be mounted on its side on the trailer. The oil will be pumped into the truck at about 500 barrels per hour.

TCP-CO

Hardware Specification

(Engineering Specification -- Hardware Specific Portion)

Project #RS970501

Smith Meter, Inc.
Erie, PA

Prepared by: ______________________________ 5/16/97

Senior Electronic Engineer & Group Leader (R. Smith) Date

5/16/97

Approved by: ______________________________ 5/16/97

Manager Electronic Engineering & Automation (D. Resch) Date

1.) DC Instrumentaion Power: 11-30 Vdc, x.x Watts maximum

2.) Pulse Input (Qty-2):
(Note: TPU ports)

a.) Sensitivity -
$V_{Threshold}$ - 1.32 Vdc +0.72/-0.25
b.) Minimum Pulse Width - 50uS
c.) Maximum Frequency - 10kHz
d.) (Tenative) Resistive Pull-Up to 5Vdc
or Resistive Pull Down

3.) Bi-State Outputs (Qty-3):
(Note: TPU ports)

a.) Optically Isolated Solid State Relay
b.) Maximum load current - 200 mA
c.) Maximum Load Voltage - 30 Vdc
d.) Turn-On - 3mS max @ 50mA
e.) Turn-Off - 3mS max @ 50mA
f.) Capacitance - 200pf Typ.

4.) Bi-State Inputs (Qty-3):
(Note: TPU ports)

a.) Optically Isolated Solid State Voltage Sensor
b.) Input Voltage Range - 11-30 Vdc
c.) Current at maximum voltage -
10 mA maximum

5.) Analog Input (Temperature):

a.) 4-wire 100 ohm platinum resistance
temperature device (PRTD)
b.) Temperature coefficient at 0°C -
0.00385 ohms/ohms/°C
c.) Temperature Range - -100°C to 300°C
d.) Accuracy - 0.1°C maximum error

6.) Analog Input (BS&W, or other):

a.) 4-20 mA input
b.) Negative terminal galvanically connected to
negative power input (truck ground-frame)
c.) 100 ohm input burden

7.) Async. Communications (Qty-3):

a.) (1)Hardwire:
i.) EIA232 Compatible
ii.) 300 to 19,200 bits per second
b.) (1)Hardwire:
i.) EIA485 Compatible
ii.) 300 to 19,200 bits per second
c.) (1)Radio Frequency
i.) ~~900~~Mhz ISM band
ii.) No user license required
iii.) 50' minimum distance
iv.) 300 to 4800 bps

8.)	Display:	a.) 128 x 64 pixel Liquid Crystal Display (LCD) b.) LED Backlighting
9.)	Key Board:	a.) Sealed pillow-embossed membrane keypad b.) 20 position
10.)	Enviromental:	a.) General Purpose Locations (Non-Hazardous) *(Note: All circuits eminating to/from unit are NOT intrinsically safe)* b.) IP32 Enclosure c.) -25°C to 60°C ambient operating
11.)	Enclosure:	a.) Plastic Bopala enclosure b.) 7.25" W x 5.5"H x 2.5"T
12.)	Mounting:	a.) Panavise 3-bolt-circle compatible
13.)	(*Tenative*) Optional Power Supply for Transmitter and/or Printer	a.) 24Vdc +/- 5% @ 1.2 Amps max (non isolated) (not available in extruded enclosure)
14.)	Weights and Measures Approvals:	a.) None
15.)	Safety Approvals:	a.) UL General Purpose Locations (UL508)

FMC **Smith Meter Inc.**
An FMC Corporation subsidiary

Truck Computer

TCP-CO™

TCP-CO

SOFTWARE SPECIFICATION

PROJECT # RS970501

SMITH METER, INC.

A FMC COMPANY

Prepared by: ______________________
Richard G. Braun, Senior Software Engineer

Approved by: ______________________
Constance L. Gibbons, Senior Software Engineer

Approved by: ______________________
Robert M. Smith, Senior Electronic Engineer

Approved by: ______________________
Dale A. Bohman, Electronic Products Marketing Manager

Approved by: ______________________
David P. Resch, Manager of Electronic Engineering

Rev. 0 6/20/97

Product Description

Scope

This document is the preliminary software function specification for the development of TCP-CO software. It provides the detailed descriptions of the software related functions required of TCP-CO, display/report formats, input/output requirements, parameter definitions, and calculations necessary to achieve that functionality.

Overview of General Functions

The TCP-CO will be a truck mounted microprocessor based electronic instrument designed for the recording of crude oil transported between an oil tank (producer) and a pipeline which is considered a custody transfer. The TCP-CO will be capable of running with unlimited loading with a manual or automatic truck full (end load) shut off.

There will be three digital inputs and outputs which will be available for use as (inputs) permissive, truck full and (outputs) such as sampler, pulse output, alarm relay, and block valve or pump. A single pulse input will be available and may be in mass or volume units. Two analog inputs will be available, one RTD for temperature and a 4-20 milliamp input for density, temperature, or sediment & water. Three serial communications ports will be available for a printer, host communications, radio modem, or mass meter communications.

The TCP will require that an API table be programmed for the product to be loaded. Prior to starting a load, the driver would select a stop record containing the lease ID and Tank ID. This information could be downloaded as part of the route information via communications by a host computer via radio modem at the start of a shift A local method of entering this information will also be available. All critical shift information will be stored in non-volatile memory.

At the start of the selected load, an optional permissive would be checked. Once met, a block valve /pump may be turned on and the load would begin. During the load loss of permissive could result in an alarm, and (optionally) closing of block valve. A run display will be available displaying in the programmed units such as indicated volume (IV), gross (GV), gross at standard operating conditions (GSV), net standard volume (NSV) and mass. The load may be terminated depending on programming by truck full or end load permissive, no flow or volume time out, or manually. All volumes accumulated and compensation will be done in accordance with API chapters 11 & 12. Rounding (versus full resolution) to these standards will be a programmable option. A corresponding set of throughput volume & mass totals will be kept and can only be reset via diagnostic or full system initialization with rollover at 10 whole digits.

The TCP will be able to store stop data in non-volatile (local storage) memory. At the end of a shift all stop information retained in non-volatile memory since the last shift could be uploaded to the host computer. Note that under ordinary circumstances the current and one previous shift will be retained in non-volatile memory. Once this information has been retrieved by the host and the next shift data is downloaded the current shift becomes the previous shift and the current shift has begun. However, if for some reason the shift information is not downloaded to the host (e.g., host computer is down) the unit would enter the local mode of operation (i.e., when no host communications is configured) and continue storing stops until the maximum amount of stops that can be stored has been reached at which time a local storage full alarm will be generated, if enabled. Optionally, the operator could continue running thereby losing the previous shift information or could not run until the information was cleared by the host (or diagnostic). A maximum number of 16 stops per shift is possible. A protocol for communications to/from the host computer has yet to be determined and will be specified in a separate document.

A communications port will be available on the TCP for printing of receipts after each stop. Prior to the printing of the receipt(s) up to 10 optional series of prompts request the operator to enter any observed information desired which may then be printed on the receipt. Each prompt would consist of a text message part and a optional numeric or alpha-numeric part. If a mistake is detected on the receipt of the load just completed, it may be re-printed at which time the operator would be re-prompted. It should be noted that any volume detected after the printing of the receipt would be ignored. A receipt can also be reprinted (no re-prompting would occur) for any stop that is stored in local storage at any time as long as the receipt for the last stop has been printed. The receipt format is configurable in that the items that can be printed and their line and column locations are programmable. This will be two configurable reports available that can be changed through PC support software-TCPmate and summary reports would be generated by the host computer.

The TCP may also be required to interface with a Walker unit through communications from which it would obtain information such as the start and ending tank height, temperature, estimated load, etc.

An automatic sampler output (digital) is available which may be manually enabled/disabled during the load and the process monitored. Sediment and water percent may be entered manually through prompting at the end of a load. The NSV load total would then be recalculated based on the entered data. Recalculation of a particular load would be permitted until another load has begun. As noted above, an analog input would be available for automatic input of the BS & W percentage as well as an alarm that could signal a percentage out of range. The automatic BS & W may be overridden when the load is done by manual entry.

Certain alarms non-diagnostic such as temp probe alarm can be configured to only inform the operator, prevent/stop a load, turn on the alarm relay.

A logging feature will record local old and new parameter changes and time and date of change occurrence. This information can then be obtained through the TCPmate PC support software for audit trail report generation.

The TCP will be designed with the future capability of language translations by the TCPmate such that the TCPmate would be capable of presenting the literals in a menu by menu fashion thereby simplifying the translation process.

Up to two security levels are possible, and each is configurable. The security levels may be created in the security directory. Each security level may have an access code. A digital input may be assigned to the highest level used. Each parameter in the database may then be assigned a security level. Changing of that parameter would then require that the security restrictions have been met. This will allow control of access to sensitive parameters, such as those which affect volume accuracy, to be restricted to the degree of security desired.

In the event of a power failure, the TCP-CO will store all of the variables and parameters needed to ensure proper operation when power is restored. The program data will contain a checksum used to detect corruption. In the event that data corruption has been detected, a diagnostic alarm will occur on power-up.

The ROM's will include a signature. This will be used to generate a message when the system is first powered up after a new ROM revision is installed. This prompt will identify to the user that a new revision of firmware has been installed. This will allow them the opportunity to upload any current configuration data to a host system before proceeding. The data can then be downloaded after initializing the TCP-CO. Then only new parameters associated with the new release will need to be entered.

Terminology

"Reports" describes the reports that are available for use through the TCP.

"Operations" describes the sequence required to start and end a load as well as any related processes such as sampling. The process of manually adding a customer to the route is also described. The occurrence and operation of any exception conditions such as local storage full is described where appropriate.

"Sampler" describes the process of enabling, starting and stopping a sample cycle and entering new values for the percentage of BS&W.

"Alarms" describes the active and logged alarms and provides a description of each alarm.

"Diagnostic Menu" describes the information and functions of items listed on the Diagnostic Menu.

"The Index" is provided to enable the user to easily locate the topic that he is looking for.

The examples presented in this specification are for clarity and convenience. The values will vary for each particular installation and/or operation.

Description of the Modes of Operation

The Program Mode section discusses Program Mode entry, (see program mode section) operation, parameter setting and description. Where there is a conflict between programming of parameters for the desired operation, appropriate error messages will be displayed.

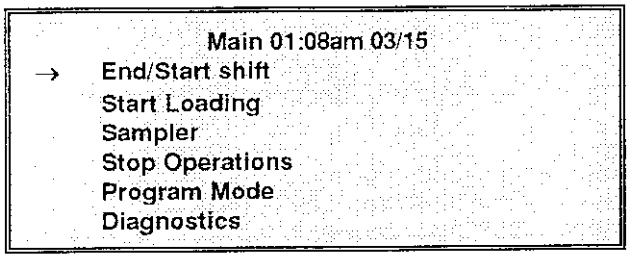
A load may be run by selecting a stop record which contains the lease and tank ID. A stop record may be programmed locally through use of the Add Stop Menu that is accessed through the Stop Operations Menu. This menu can also be accessed through communications.

Any volume that flows through the meter between the time one load is ended and the next is started is considered unauthorized and is ignored.

A meter can have several I/O points set up as permissive Start, Truck full, Block Valve Control, and a Pump Control. These I/O points can be set up in the I/O menus.

Start Up

When the unit is powered up, a logo screen will briefly appear (5 seconds, or press any key to clear) followed by the main menu if no stop records have been downloaded:



```
          Main 01:08am 03/15
 →  End/Start shift
    Start Loading
    Sampler
    Stop Operations
    Program Mode
    Diagnostics
```

If the unit powers up with stop records, then the next stop record would appear in the same manner as if start loading was selected (see start loading section below).

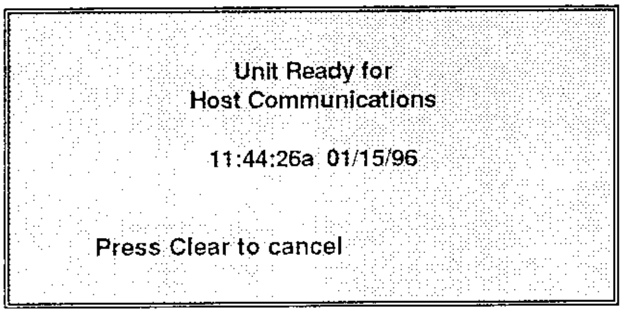
To select a main menu item, use the up/down arrow key to select an item then press enter.

Note: Where operations permit, the main menu may be accessed by pressing the menu key. The diagnostics alternative menu may also be accessed from other screens by pressing the F1 key.

End/Start Shift Menu

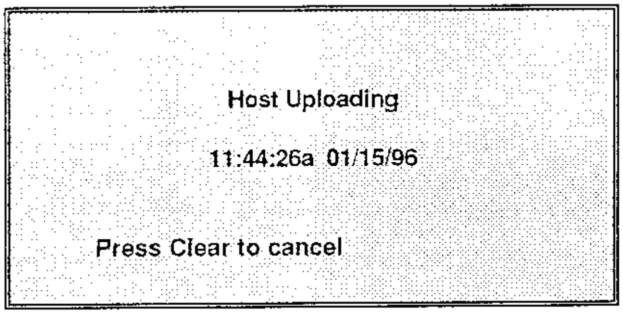
When it is desired to upload shift information to the host computer and subsequently download route information to the TCP, this entry must be selected in order to perform any necessary housekeeping, place the TCP in a known state, and allow the operations to be monitored for correctness.

When the unit is ready the following will be displayed:



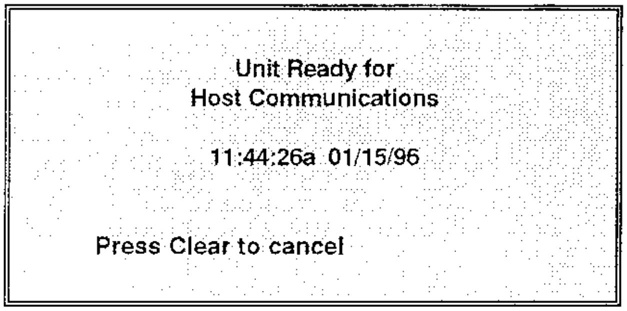
```
        Unit Ready for
     Host Communications

      11:44:26a 01/15/96

  Press Clear to cancel
```

Once the host starts uploading shift information the following appears:



```
        Host Uploading

      11:44:26a 01/15/96

  Press Clear to cancel
```

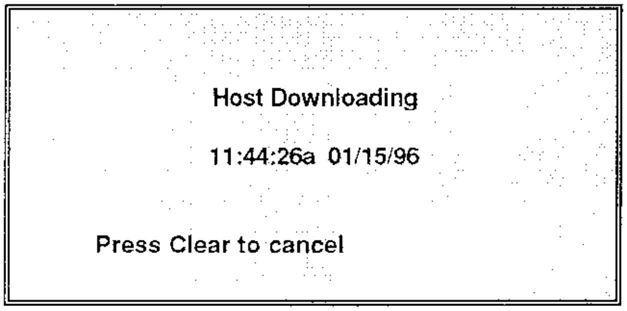
Once complete, the following appears:


Unit Ready for
Host Communications

11:44:26a 01/15/96

Press Clear to cancel

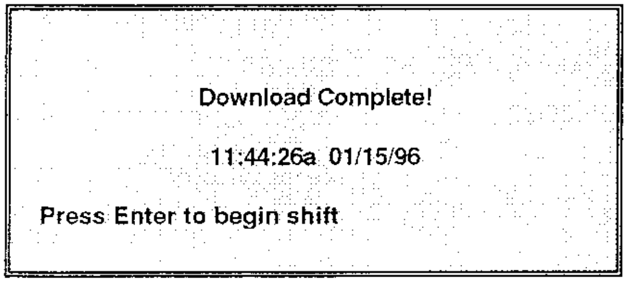
Once the host starts downloading route information for the next shift the following appears:


Host Downloading

11:44:26a 01/15/96

Press Clear to cancel

Note: If the host initiates the download/upload process, then this screen will automatically appear. However, this will not be allowed if the TCP-CO is in the program mode or currently in the loading sequence. Once all the new shift information has been received, the following will be displayed:

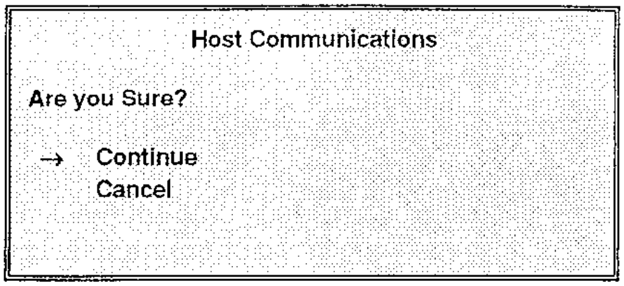

Download Complete!

11:44:26a 01/15/96

Press Enter to begin shift

Press enter to begin the new shift and return to the main menu.

If clear is pressed prior to shift completion, the following will appear:


Host Communications

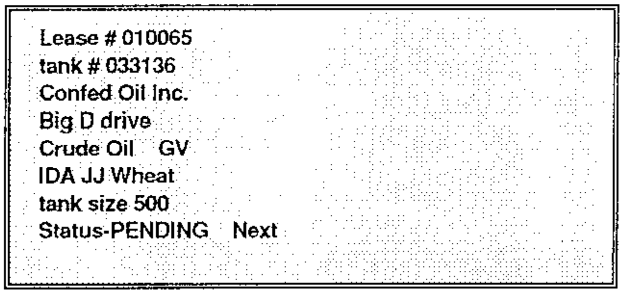
Are you Sure?

→ Continue
Cancel

Selecting cancel will result in the loss of any new shift information uploaded and the unit will return to the main menu.

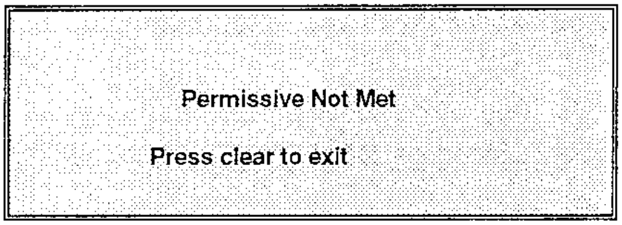
Start Loading

To start a load, select "Start Loading" on the Main Menu after the shift had been downloaded. The view all stops display will appear and the operator would use the next key to view the stops that are scheduled.


Lease # 010065
tank # 033136
Confed Oil Inc.
Big D drive
Crude Oil GV
IDA JJ Wheat
tank size 500
Status-PENDING Next

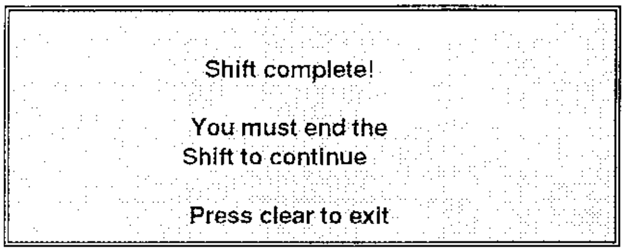
You cannot select a previous loaded stop. However, you may obtain a re-print of a stop that was loaded by selecting that stop and pressing the print key.

Press the start key to select the stop to be loaded At this time, the large run display will appear, any block valve or pump will be turned on. If a permissive is programmed, a display will appear informing the operator that a permissive is to be met is needed before the load can continue.


Permissive Not Met

Press clear to exit

Note that if all stops have been loaded, and the unit is configured to run with host control, then a shift complete message will issued :



Shift complete!

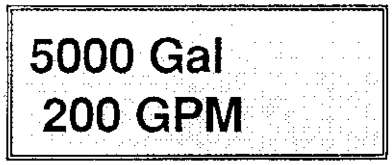
You must end the
Shift to continue

Press clear to exit

You must end the shift either through host communications or manually in the program mode. At that time all stop status' will be reset and all stops will again be available for loading and editing. If host communications is lost, then this process will continue until the storage capacity (non-volatile) has been reached. at which time a storage full alarm will be issued. Depending on programming, operation will be permitted but the oldest shift information would be lost. Press "clear" to exit to the previos menu or press the "menu" key to go to the main menu.

It is possible to manually end the shift before the completion of all stops by entering the program mode, diagnostics menu and selecting the end shift parameter.

Large Run Screen



5000 Gal
200 GPM

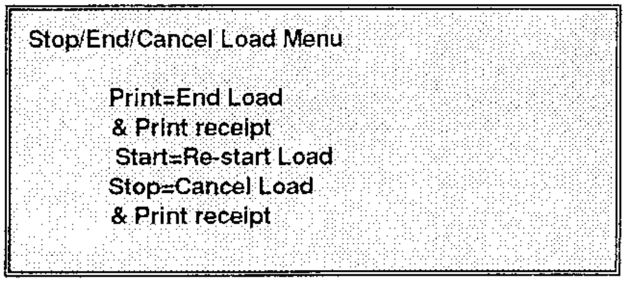
F1=Diagnostics, Next=Run Screens, Print=enter prompts, end load.

This screen displays the delivery volume and flow rate in a large font for ease of reading for the current load. The volume type units (e.g., GSV) and the flow units are those programmed in the program mode. You may view the other run time data screens by pressing the next key (goes to the next menu in sequence) or pressing the number key that corresponds to the screen number. If it is desired to begin the prompt entry sequence while loading, press the Print key to begin the prompt sequence with the first prompt in order. See the prompt sequence description below under end load.

You may press print from this screen to end the load and start the prompt sequence at any time.

Stop Load/End Load/Cancel Load

Press the stop key to stop a load in progress. This will shut any block valve and turn off any pump configured. Once the flow is stopped the operator has the choice of restarting the load,canceling the load or ending the load using the following menu:



Stop/End/Cancel Load Menu

Print=End Load
& Print receipt
Start=Re-start Load
Stop=Cancel Load
& Print receipt

This menu will also be displayed for permissive, zero flow or volume cutoff options. If cancel load is selected, the stop record will be flagged as aborted and cannot be re-selected for another load. If it is desired to load this stop again, use the add load menu to copy this stop and create a new stop. Then proceed as before.

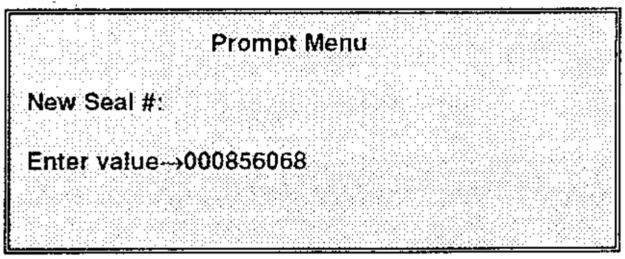
Note: This screen will only be displayed if a load is in progress .

Re-start Load

To start a load that was "stopped" press the start key . This will open a block value and turn on a pump if so configured. Any permissive configured would be required as per start of a load.

End Load

Pressing the Print key will start the prompt sequnce and end the load if desired and print a report if so programmed.



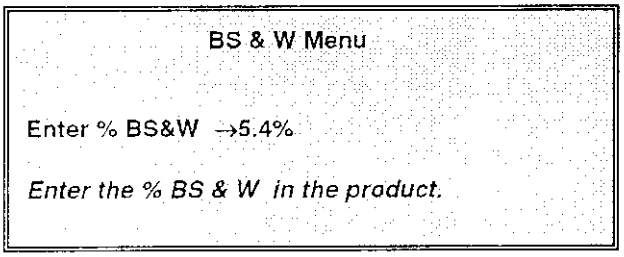
Prompt Menu

New Seal #:

Enter value→000856068

This sample prompt menu shows the prompt message, "New Seal #:" followed by the prompt for its value. This information could then be printed on the receipt. Up to 20 prompts are possible the number being determined by the host download at start of shift. After the prompts, the BS & W menu will appear that will allow the operator to Change the BS&W and the sampler menu described below will appear.

Note: During this prompting process, the operator may cancel this process by pressing the stop key. The stop/end load menu will appear and he may then have the option to re-start the load or the prompting process over again.

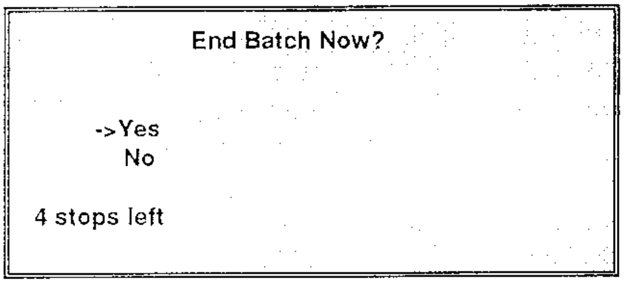


BS & W Menu

Enter % BS&W →5.4%

Enter the % BS & W in the product.

Once the BS & W entry is complete, the prompt confirmation menu will appear:



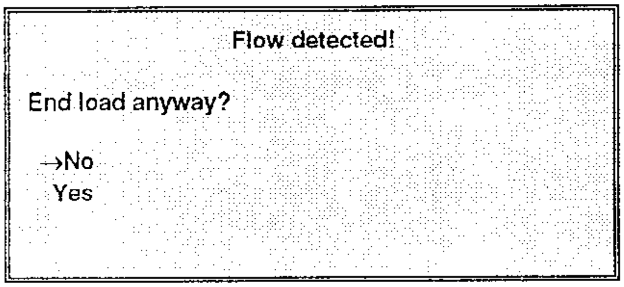
End Batch Now?

->Yes
No

4 stops left

If "No" is selected, the large run screen appears and the operator may press print again to modify/re-enter only those in error. If "Yes" is selected, the stop record will be updated, the receipt will be printed and the customer select menu will appear again. It will begin with the next customer on the route. If no stops are left, "No stops left-end shift" will appear in place of the number of stops left. Selecting "save prompts" in this case brings up the main menu allowing the operator to select ending the shift for host communications or enter the program mode to manually end the shift.

If end the load is selected but flow is present then the following appears:



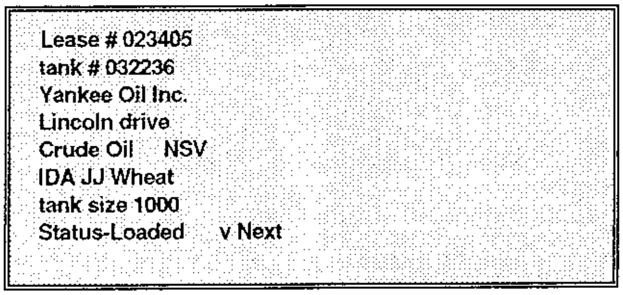
Flow detected!

End load anyway?

→No
Yes

Selecting "No" returns the operator to the large run sceen while "Yes" continues to the next stop as with no flow present.

Note that once the receipt is printed, all subsequent volume will be ignored until the next load is started.

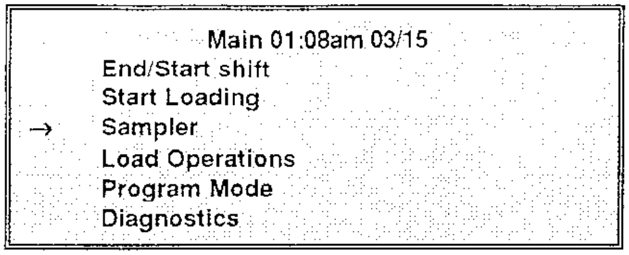


Lease # 023405
tank # 032236
Yankee Oil Inc.
Lincoln drive
Crude Oil NSV
IDA JJ Wheat
tank size 1000
Status-Loaded v Next

Just press the start key to start the next load.

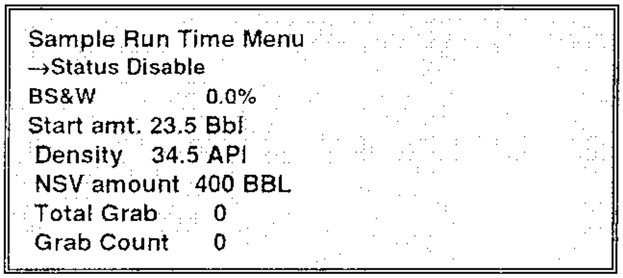
Sampler Menu

Select the Sampler Menu from the Main Menu to enable or disable the sampler, set the start amount, and the percentage of sediment and water allowed in the product.



```
        Main 01:08am 03/15
    End/Start shift
    Start Loading
→   Sampler
    Load Operations
    Program Mode
    Diagnostics
```

Pressing 'ENTER' with the cursor in front of "6 Sampler" displays the Sample Run Time Menu.

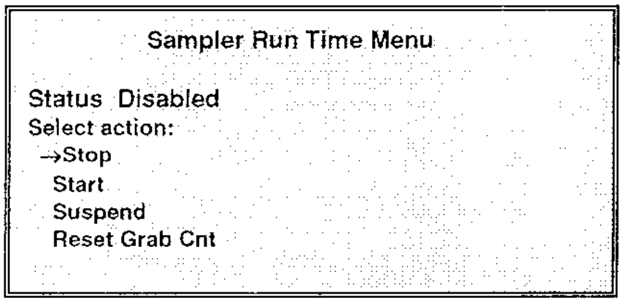


```
Sample Run Time Menu
→Status Disable
BS&W          0.0%
Start amt. 23.5 Bbl
 Density   34.5 API
 NSV amount  400 BBL
 Total Grab      0
 Grab Count      0
```

From the screen the sampler status and % BS&W can be changed. Information that is available for viewing only is the start amount, density, I/O point assigned to the sampler, the NSV product amount, total amount of product that has been taken (Total Grab) and the number of samples taken (Grab Count). The sampler start amount can be changed in the program mode.

Sampler Status

Sampler Status is selected by placing the cursor in front of "1. Sampler Status" and pressing 'ENTER'. The screen for changing the status will appear.



```
        Sampler Run Time Menu

Status  Disabled
Select action:
 →Stop
  Start
  Suspend
  Reset Grab Cnt
```

From the screen the operator can start, stop or suspend or reset the grab count of the sampler.

Stop - If the sampler has been started or is suspended selecting 'STOP' will disable the sampler and reset the totals.

Start - If the sampler has been stopped or suspended selecting 'START' will either start a new sampling load or restart a suspended sampling load.

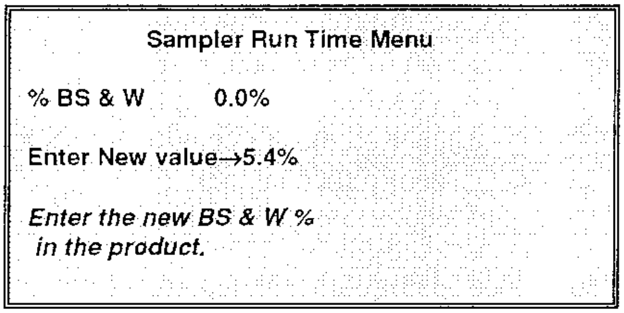
Suspend - Selecting 'SUSPEND' from the menu will stop the sampling process but will not reset the totalizers. When suspended the operator will have the choice of restarting the sampler activity or stopping it.

Reset Grab Count - Selecting "Reset Grab Count" from the menu will reset the TCP grab count counter to zero.

When the selection has been made and 'ENTER' pressed the unit will return to the Sampler Run Time Menu.

Percentage Sediment & Water

The operator can enter the percentage of sediment and water by selecting "% BS & W" and pressing 'ENTER'. The percentage of sediment and water entry screen will be displayed.



```
        Sampler Run Time Menu

% BS & W      0.0%

Enter New value→5.4%

Enter the new BS & W %
 in the product.
```

Enter the new percentage of sediment and water. Pressing 'ENTER' will store the new percentage and the TCP will recalculate the Dry Product Volume using the new entries and return the unit to the Sampler Run Time Menu.

GV x (1 - bs&w/100) x ctl(dry density) = NSV

Print Reports

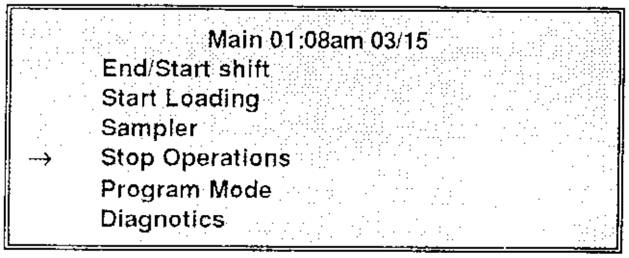
As noted, two default configurable reports will be available for use and modification by the TCPmate. An example of these reports, one for land management use and the other for non-land management use, respectively, are shown on the following pages.

```
Run Ticket             No. 134023

04-06-97 9:11 AM

Load Date/Time:
Start: 04-06-97 9:02 AM
End:   04-06-97 9:07 AM

Operator: Quay Valley

Shipper: Pride Pipeline Co.

Lease: Russell Fed #17
Destination:

Lease No. 403747

Tank No. 032801
Truck/trailer  No. 854 / 711

Product Record           1
  Product Name:          Crude
  API Table:             5A
  Reference Density:     34.5API

Tank Gauge Start:07'05.00" End:01'10.75"
Open Temp:065F   Close Temp: 067F
Tank Bottom 008
Estimated Bbl            160.24
Trip Tick # 421505 Meter Tick # 240725
      Meter Totals
GV                       160.89 Bbl
GSV                      159.86 Bbl
NSV                      159.54 Bbl
Mass Delivered           50688.32

CTL                      0.9936
CSW                      0.9980
CCF                      0.9916

Load Avg Temperature     66.7   F
Load Avg Density         32     API
Load Avg BS&W            0.2    %

Alarms:

Old Seal # _ _ _ _ _ _ _ _ _ _

New Seal # _ _ _ _ _ _ _ _ _ _

Station # _ _ _ _ _

Seller ________________________________

Federal I.D.#__________________________

Legal Desc. 1/4 sec./1/4 Sec. _________
            Sec.______________________

            T.__________R.____________
Remarks:


------------------------------------
Driver Gauger

------------------------------------
Operator Representative
```

```
Run Ticket              No. 134023

04-06-97 9:11 AM

Load Date/Time:
Start: 04-06-97 9:02 AM
End:   04-06-97 9:07 AM

Operator: Confed Oil Inc

Shipper: Pride Pipeline Co.

Lease: IDA JJ Wheat
Destination:

Lease No. 10065

Tank No. 033136
Truck/trailer  No. 894 / 713

Product Record          1
   Product Name:        Crude
   API Table:           5A
   Reference Density:   36.2API

Tank        Gauge       Start:07'05.00"
End:01'10.75"
Open Temp:065F   Close Temp: 067F
Tank Bottom 008
Estimated Bbl           160.24
Trip Tick # 435317


      Meter Totals
GV                 160.89 Bbl
GSV                159.86 Bbl
NSV                159.54 Bbl
Mass               50688.32

CTL                     0.9936
CSW                     0.9980
CCF                     0.9916

Load Avg Temperature    66.7    F
Load Avg Density        32      API
Load Avg BS&W           0.2     %

Alarms:

Old Seal # _ _ _ _ _ _ _ _ _ _

New Seal # _ _ _ _ _ _ _ _ _ _

Station # _ _ _ _ _

Remarks:



-----------------------------------
Driver Gauger



-----------------------------------
Operator Representative
```

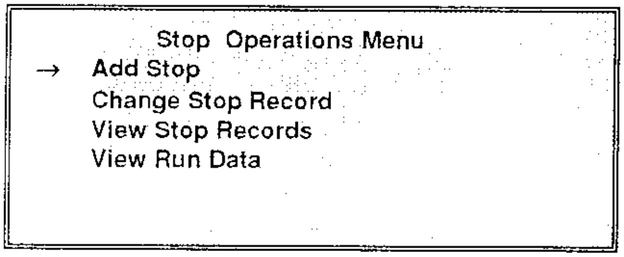
Stop Operations

Stop records may created by entering the Stop Operations Menu.



```
          Main 01:08am 03/15
     End/Start shift
     Start Loading
     Sampler
 →   Stop Operations
     Program Mode
     Diagnotics
```

Selecting the Stop Operations entry by using the arrow keys to point to the desired entry and pressing 'ENTER' will display the Stop Operations Menu.



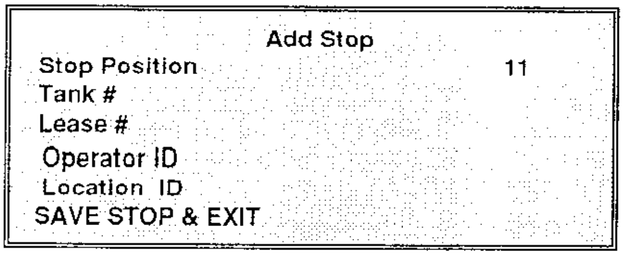
```
        Stop  Operations Menu
 →   Add Stop
     Change Stop Record
     View Stop Records
     View Run Data
```

If there are no stop records set up for this truck. the only options that will be accessible on the menu will be "Add a Stop" or viewing of data.

Once stops have been added to the setup, the other options will become accessible to the operator. Note that if under host control, only stops not yet loaded may be edited within the stop limit set. Once all stops are loaded, the shift is considered complete and the shift information must be downloaded to the host in order to continue operation and obtain new route (stop) information.

Add Stop

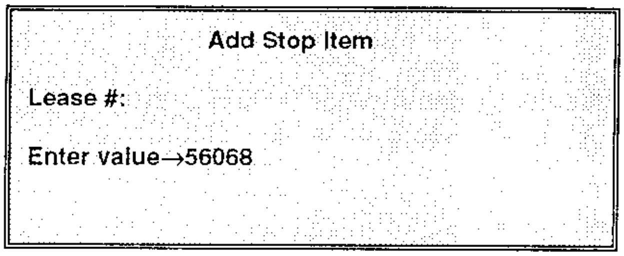
When "Add Stop" is selected from the menu the Add Stop screen will appear:



```
                Add Stop
 Stop Position                       11
 Tank #
 Lease #
 Operator ID
 Location  ID
 SAVE STOP & EXIT
```

The Stop Position indicates the selection position of the stop that is being set up. This will be a number from 1 to 16; 16 is the maximum number of stops that can be set up.

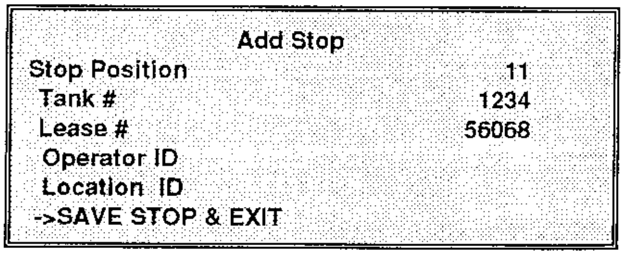
The operator would select the Tank, Lease number, and up to five additional items designated by the host such as the customer name, location, lease name, etc. from the route information downloaded from the host at the start of the shift. (This may include a generic customer entry which could allow print out titles such as "Customer:" followed by a blank area allowing for a handwritten entry). For example, use the arrow key to move the arrow to Lease # and press enter the following would appear:



```
              Add Stop Item

 Lease #:

 Enter value→56068
```

For numeric items numbers would appear as typed. For apha-numeric entries, a 4 line box would appear indicating the character being selected using the arrow keys. Consult the program mode for a description of how to make alpha-numeric entries.

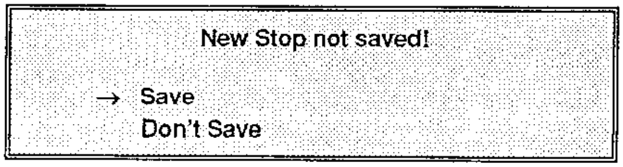
When the process of adding stop itemse is completed, move the arrow to "SAVE NEW STOP & EXIT" as shown in the following display:



```
                Add Stop
 Stop Position                       11
  Tank #                           1234
  Lease #                         56068
  Operator ID
  Location  ID
 ->SAVE STOP & EXIT
```

Press 'ENTER' to add the new stop to the memory of the TCP.

If "Save New Stop & Exit" is not selected and the 'CLEAR' key is pressed, a pop up screen will appear asking if the stop should or should not be saved.

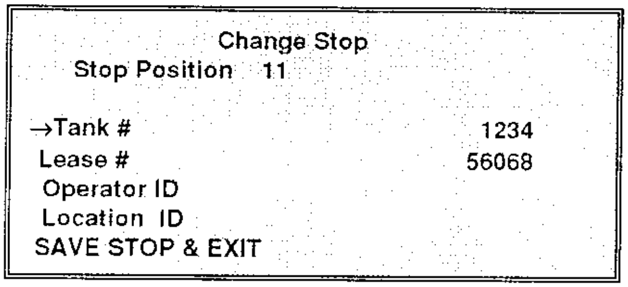


```
           New Stop not saved!

     →  Save
        Don't Save
```

When either is selected and the 'ENTER' key is pressed the TCP will revert back to the "Stop Operations Menu". If 'CLEAR' is pressed the unit will revert back to the stop definition screen.

Change A Stop

When Change Stop Record is selected the display for changing a stop will be displayed. The stop record that can be changed is the current stop position shown on the View Stop Screen.



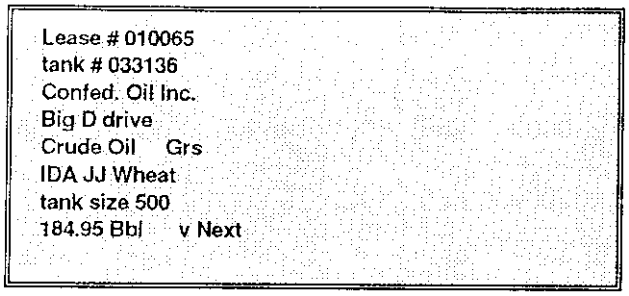
```
            Change Stop
   Stop Position  11

→Tank #                         1234
 Lease #                       56068
 Operator ID
 Location  ID
SAVE STOP & EXIT
```

This screen shows the current stop information that is programmed. It indicates the stop position, the tank and lease number, and up to five additional items designated by the host. This information can be changed by going through the same steps that were described for adding a stop and when the information has been changed pressing "**SAVE STOP & EXIT**" and 'ENTER' to store the information in memory. If the information is not stored in memory the information will be lost.

Note: Only stops added locally can be changed. Stops downloaded from the host cannot.

View All Stops Defined

Selecting "View All Stops Defined" from the Operations Menu will display all the stops that are currently defined. This display is an informational display only. No changes to the stops can be made through this display. However, you may obtain a re-print of a stop that was loaded by selecting that stop and pressing the print key.



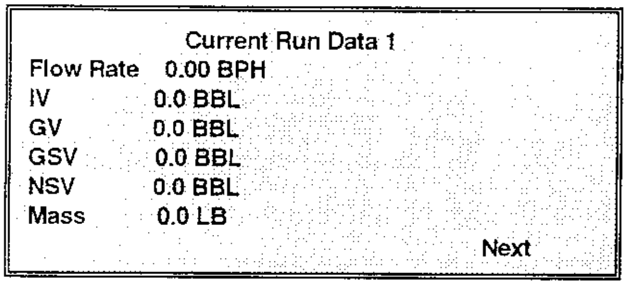
```
Lease # 010065
tank # 033136
Confed. Oil Inc.
Big D drive
Crude Oil    Grs
IDA JJ Wheat
tank size 500
184.95 Bbl    v Next
```

Using the next key to view each load defined on the route .Pressing 'CLEAR' will return the unit to the Operations Menu. The stop status symbol will also be shown where:

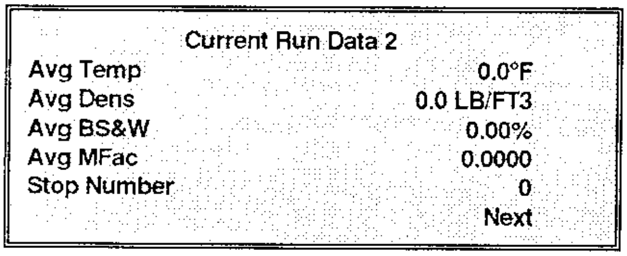
The load amount will be shown if the customer had been loaded since the last shift. A "v Next" in the lower right hand corner means there are more stops to view. Press the Next key to do so.

View Current Run/Dynamic Data

Selecting "View Current Run Data" from the Stop Operations Menu will display the screen data for the current stop being run or the previous stop that was run if there is not a stop currently being run



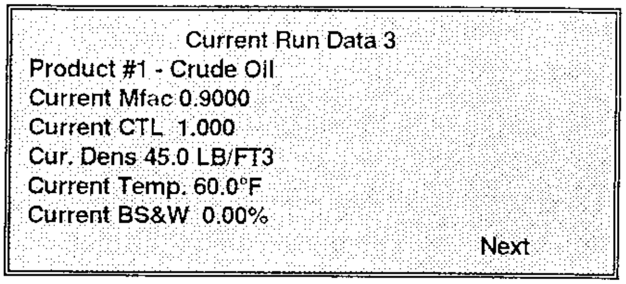
```
            Current Run Data 1
Flow Rate   0.00 BPH
IV          0.0 BBL
GV          0.0 BBL
GSV         0.0 BBL
NSV         0.0 BBL
Mass        0.0 LB
                                    Next
```



```
            Current Run Data 2
Avg Temp                            0.0°F
Avg Dens                      0.0 LB/FT3
Avg BS&W                           0.00%
Avg MFac                          0.0000
Stop Number                            0
                                    Next
```

These screens provide information on the product being delivered,the load totals (IV, GV, GSV, NSV, Mass) the current flow rate, load average temperature, load average density of the product ,the load average meter factor, the volume type being used for the load, and the load number . If a load is not currently being delivered, the last completed load will be displayed.

Pressing next will display the next page or enter the screen number to directly go to the desired screen.

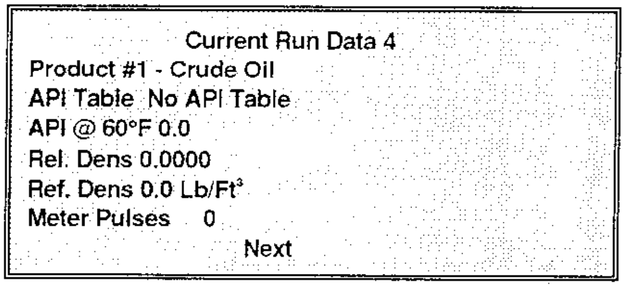


```
            Current Run Data 3
Product #1 - Crude Oil
Current Mfac 0.9000
Current CTL  1.000
Cur. Dens 45.0 LB/FT3
Current Temp. 60.0°F
Current BS&W  0.00%
                                    Next
```

This screen provides information of the the current meter factor, temperature, CTL, density, and BS&W percent.

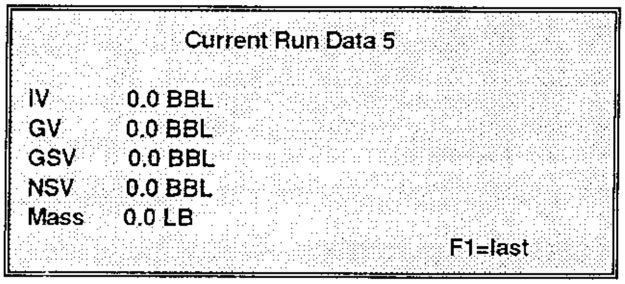
If no load is active at the time the data is being requested, the information displayed will be from the previous load.

Pressing the next key will display the fourth page of information.



```
            Current Run Data 4
Product #1 - Crude Oil
API Table  No API Table
API @ 60°F 0.0
Rel. Dens 0.0000
Ref. Dens 0.0 Lb/Ft³
Meter Pulses    0
                        Next
```

This screen provides information of the API Table being used, the corrected API, the corrected relative density and the corrected reference density.

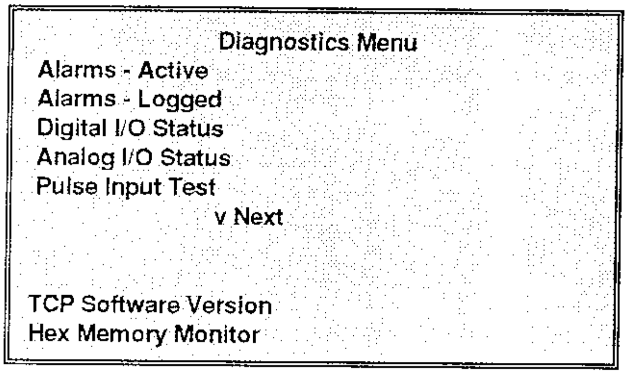


```
            Current Run Data 5

IV      0.0 BBL
GV      0.0 BBL
GSV     0.0 BBL
NSV     0.0 BBL
Mass    0.0 LB
                                F1=last
```

This screen provides throughput totals of all volume types.

Diagnostics Menu

The TCP has been set up to allow the operator to have access to certain displays by selectiing the diagnostic menu in the main menu. The displays that are on the "Diagnotics Menu" are all fully described in the manuals listed below.

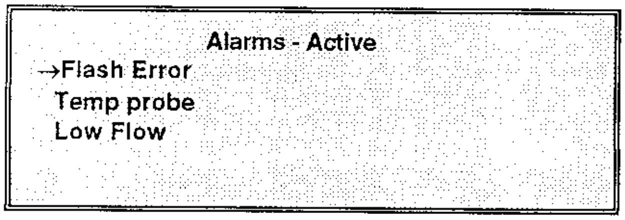


```
          Diagnostics Menu
 Alarms - Active
 Alarms - Logged
 Digital I/O Status
 Analog I/O Status
 Pulse Input Test
                v Next


TCP Software Version
Hex Memory Monitor
```

Active Alarms

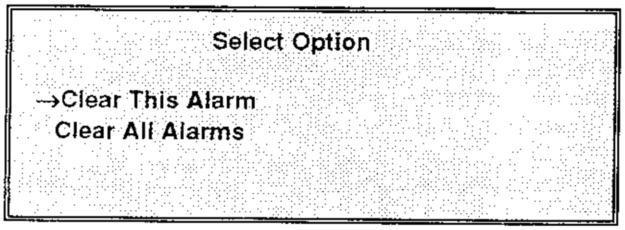
The Diagnostics menu provides a selection "Alarms - Active" where all active alarms can be viewed and cleared.

When 'ENTER' is pressed the Active Alarms Menu will be displayed with the cursor next to the first alarm in the menu.



```
            Alarms - Active
→Flash Error
  Temp probe
  Low Flow
```

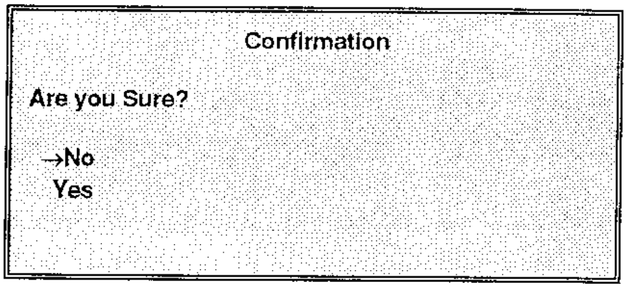
Up to six active alarms will be displayed on the Active Alarm screen. With the cursor in front of the alarm that is to be cleared pressing 'ENTER' will bring up a pop up screen.



```
            Select Option

→Clear This Alarm
  Clear All Alarms
```

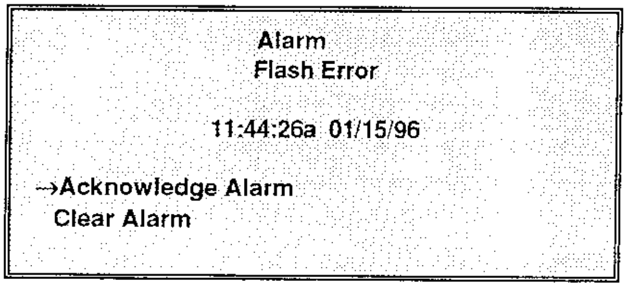
Selecting option "Clear This Alarm" will clear the alarm that the cursor was pointing to.

If all the alarms are to be cleared selecting option "Clear All Alarms" will clear the alarms that are listed on the display after a confirmation is made on the following pop-up screen.



```
              Confirmation

Are you Sure?

 →No
  Yes
```

If No is selected the screen will revert to the Select Option display. If Yes is selected and 'ENTER' is pressed all active alarms will be cleared.

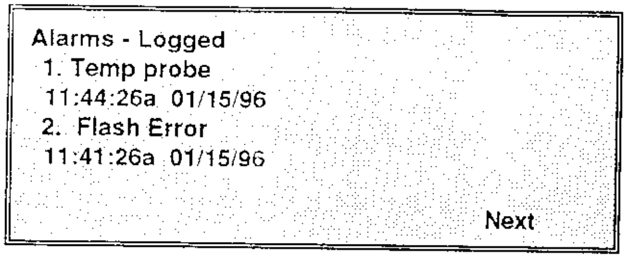


```
                Alarm
               Flash Error

          11:44:26a 01/15/96

→Acknowledge Alarm
  Clear Alarm
```

NOTE: Clearing the alarm without correcting the problem that caused the alarm will cause the TCP to go into alarm condition displaying the pop up Alarm Screen.

Alarms Logged

The main menu provides a selection "Alarms Logged" where the last 18 alarms that have occurred will be displayed. The screen provides the alarm, the time and the date that it occurred.



```
Alarms - Logged
 1. Temp probe
 11:44:26a 01/15/96
 2. Flash Error
 11:41:26a 01/15/96

                          Next
```

The TCP will log six pages of alarms with three alarms on a page. When all six pages are full the next alarm that occurs will be displayed and the first one that had occurred will be replaced. Pressing the up and down arrow keys will step the operator through the pages of alarms. When the page is full the message "Next" will appear.

TCP Alarms:

System Related Alarms

Flash Error Indicates that there was a failure in the flash memory when data was being stored.

Watchdog Reset Indicates that an internal check feature has detected a possible operational problem in the microprocessor that may have affected information stored in memory.

RAM Test Failed Indicates that there was a RAM failure.

ROM Test Failed Indicates that there was a ROM failure.

Meter Related Alarms

No Flow Indicates that the TCP has signaled the block valve to open to start flow but has not received any pulses from the transmitter within time out period.

Temperature Indicates that the temperature probe or transducer has failed

Density Indicates that the density transducer has failed.

Sampler Full Indicates that the sampler can is full. This alarm is generated when the programmed percent full has been exceeded.

Sampler Full Count Alarm Indicates that the sampler programmed full count has been exceeded.

BS&W Limit Exceeded Indicates the sediment and water limit set in the sediment and water menu has been exceeded.

Permissive Removed Indicates that the programmed permissive was removed during the course of a load.

Storage Full Indicates that non-volatile memory is full and that the host needs to retrieve shift data.

Digital I/O Status

See Diagnostics Section of the Programming Mode.

Analog I/O Status

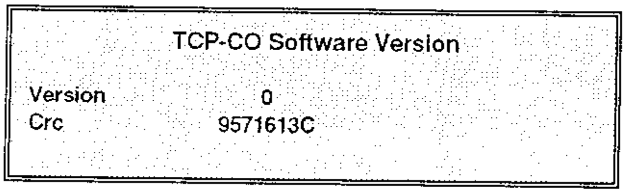
See Diagnostics Section of the Programming Mode.

Pulse Input Test

See Diagnostics Section of the Programming Mode.

TCP-CO Software Version

The TCP-CO Software Version screen displays the current software version and CRC .



TCP-CO Software Version

```
Version         0
Crc       9571613C
```

The other displays that are accessed through the Diagnostic Menu are the Hex Memory Monitory which are all used for factory internal testing.

FMC **Smith Meter Inc.**
An FMC Corporation subsidiary

TCP-RAD

PC-Based Routing and Dispatch Application for the TCP-CO

Functional Specification

June 18, 1997
Revision 0
Project # RS050197

Prepared by: ____________________
James M. Pettinato - Associate Software Engineer

Approved by: ____________________
Robert M. Smith - Project Engineer

Approved by: Constance L Gibbons
Constance L. Gibbons - Sr. Software Engineer

Approved by: Dale A. B
Dale A. Bohman - Product Marketing Manager

Approved by: ____________________
David P. Resch - Manager
Electronic Products and Automation

TCP-RAD Functional Specification

Contents

I. Overview

The TCP-RAD will be a Windows application designed to streamline and organize the routing and dispatch of tank trucks for crude oil gathering tank collection operations. It is designed to be in operation at the truck terminal, and allow the dispatcher to (1) configure his TCP-CO units, (2) define the scheduled stops for each truck, and (3) retrieve the collected product volume data upon the return of the truck. It will handle the communications interface with the TCP-CO truck-mounted electronics and control the upload and download of required data between the trucks and the main terminal computer on which it is running.

Output will be in simple delimited text files to make it trivial to access the data via the Windows application of choice for purposes of data manipulation, record-keeping and reporting. TCP-RAD will not include those features; instead the design will draw upon the functionality of existing Windows applications to allow the user to work with the data in the applications with which they are comfortable.

II. Terminology

Tank: A crude oil storage tank fed from field well(s) and not part of a pipeline system, that requires periodic offloading to a tank truck.

Truck: A tractor or tanker containing a TCP unit with an address unique to this system.

Trailer: A tank trailer containing meter and other I/O that will interface to the TCP unit. In the field, a 'trailer' may be permanently matched with a truck (i.e. a tank truck that is not segmented) or be freely exchanged with other trucks (a true semi-trailer). For the TCP-RAD, a Trailer is the collection of data that is associated with the meter and I/O attached to a particular 'trailer'.

Stop: A pickup of product from a client tank, whether pre-defined or not, and all of the information associated with the dispatching to the tank site and pickup of the product in that tank.

Accounting Period: That amount of time used to base scheduling operations for each truck. It may be a day, a week, an 8-hour shift, etc. programmable by the user.

Dispatcher: The individual(s) located at the truck terminal responsible for scheduling the stops for each truck during any particular accounting period. In this document, the user performing the daily operations with the TCP-RAD.

III. TCP-RAD Features

A. Parameter Programming Functions

One major function of the TCP-RAD will be to allow the programming of TCP-CO parameters to be modified via an easy-to-use interface. The directory structure of the TCP-CO will be mimicked by the interface at the TCP-RAD so that parameters will be presented to the user in a logically organized manner. Notebook-page type interfaces will be used to present the parameters to the user. Pick lists with valid choices will be used whenever feasible to make it easy for the user to avoid making errors. A program mode data synchronization feature operating at the TCP-RAD will inform the dispatcher of any changes made (at either end) and require the dispatcher to resolve conflicts so the programming matches at the TCP-CO and the TCP-RAD. This will occur each time communications is established between the TCP-RAD and a TCP-CO.

B. Operations Functions

The other major function of the TCP-RAD will be to streamline the daily operations of a crude oil gathering tanker fleet by computerizing to some extent the driver task assignments and custody transfer data gathering. This function will consist of the ability to provide the dispatcher straightforward methods to assign customer tanks requiring offloading to a particular truck in the fleet. Each truck may be assigned one or more tank stops to gather in each activity period. Upon return to the terminal, the data will be offloaded from the truck and stored in a flat text file with the rest of the data for that activity period.

C. TCP-CO Interface

TCP-RAD will support both the configuration of the TCP-CO and the downloading/uploading of daily activity in the form of stop data records. Configuration support will include all program mode data, with the TCP-RAD maintaining a local disk file copy of the current programming for all units. There will be method of synchronizing data between truck and TCP-RAD computer, so that the information is up-to-date and identical.

The communications interface will consist of an RS-232 interface via the PC serial port to a radio transceiver using a secure packet protocol, to be defined in a separate specification.

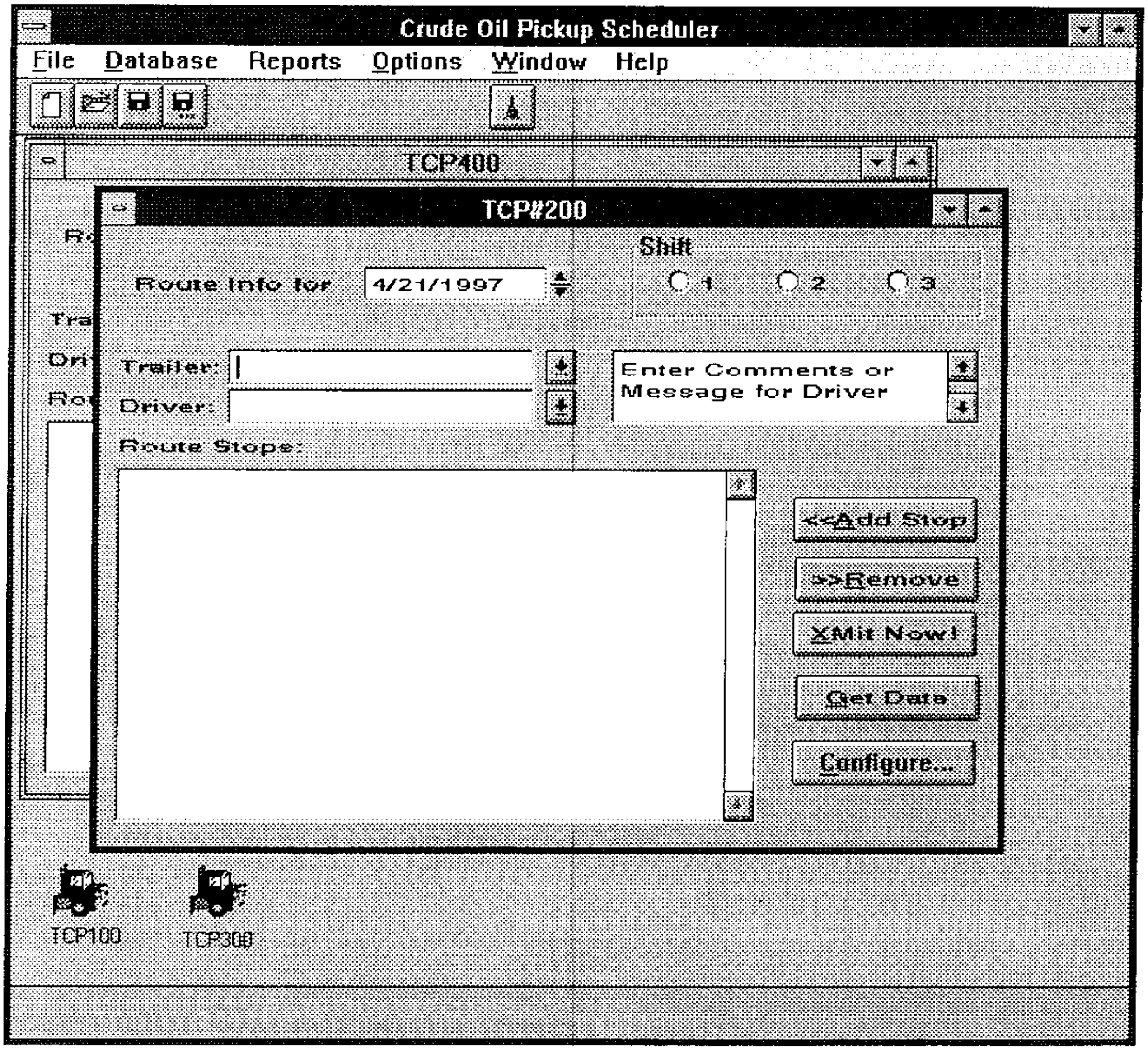
D. External Application Interface

TCP-RAD will support ASCII text file record formats to allow for easy portability of data to other applications for purposes such as record-keeping, reporting, auditing, etc. Consideration will be given to the potential for future revisions to be enhanced to support DDE/OLE, so as to allow semi-automated dispatching, data retrieval and data manipulation via a macro language resident in applications such as Excel, or for custom applications written by a 3rd party to interact with TCP-RAD data.

IV. TCP-RAD Operation

A. Program Start Up

Upon launch, the Main Window appears (without the child windows shown here):



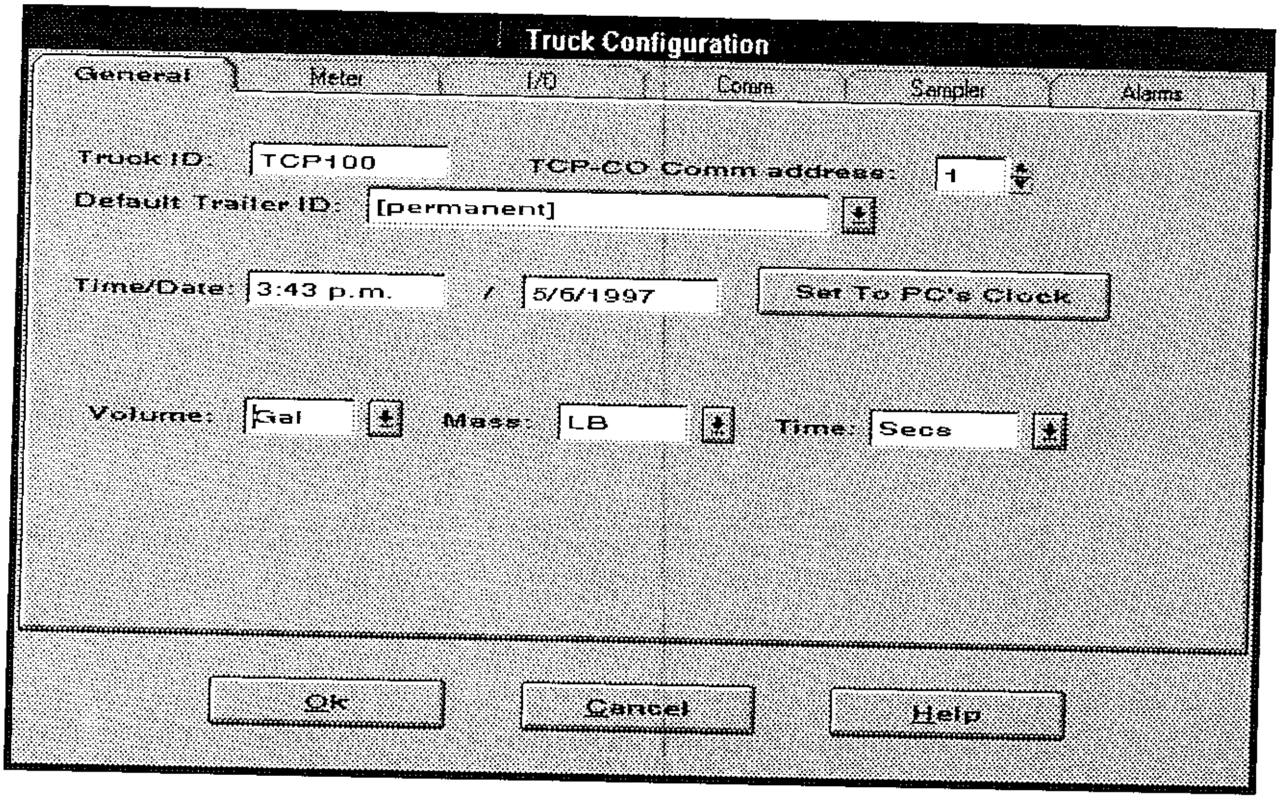
Route Info for 4/21/1997

By selecting a specific truck via a File/Open operation, defined stops during a predetermined period for that truck can be viewed and edited. The period could be a day, or a shift, etc. at the discretion of the user configuring the TCP-RAD. The stops for that period will be listed along with the status for that stop: Defined, Defined and Dispatched, or Completed. Defined stops can be issued (or *dispatched*) to the TCP-CO one at a time throughout the period, or a complete list of stops can be transmitted at the beginning of the period; the result will be identical. Any stop that is merely Defined (entered but not yet downloaded) may be removed or edited. A stop that has been Defined and Dispatched may be removed or edited only if the truck is in communications with the TCP-RAD. A stop that is Completed (delivery data has been read back from the TCP-CO by the TCP-RAD) cannot be changed by the TCP-RAD.

B. Programming Mode

User Interface

The program data interface for each TCP-CO unit will be entered via the Configure... button on the Truck window. The data fields that are programmable will each be represented by an object in a notebook window within an organization of dialog boxes. The top-level dialog box will appear as follows:



Truck Configuration

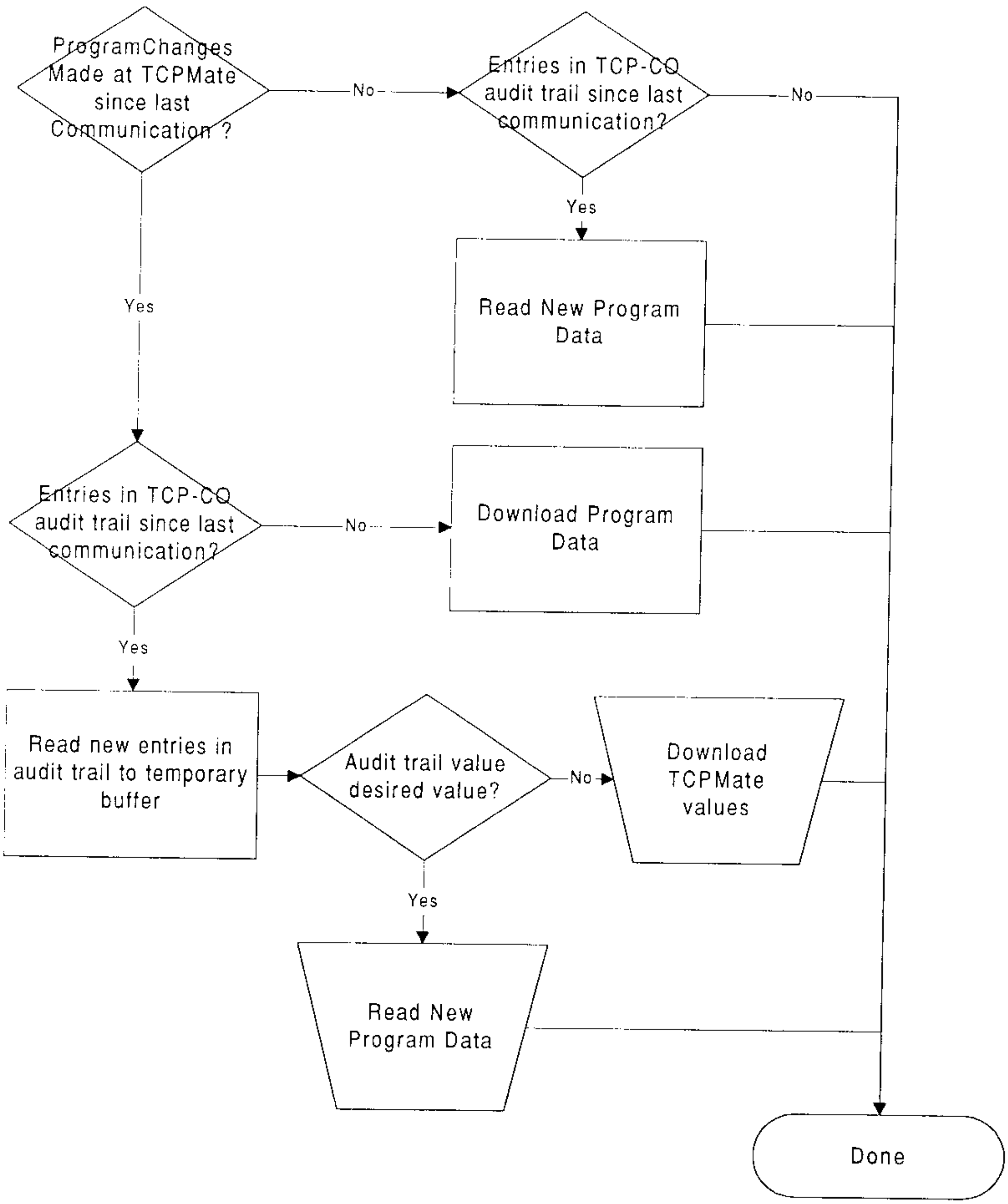
Truck ID: TCP100
TCP-CO Comm address: 1
Default Trailer ID: [permanent]
Time/Date: 3:43 p.m. / 5/6/1997
Volume: Gal
Mass: LB
Time: Secs

Program Data Dialog Box Organization

Each 'directory' or main section in the Program Mode of the TCP-CO will have a corresponding notebook page. Some variation may occur due to asthetics, i.e., some directories may be divided into multiple pages. Truck configuration dialog box notebook pages will be labeled as in the figure above. Each notebook page will be organized in a logical manner so the data is grouped as to be most intuitive for the user.

Program Mode Data Synchronization between TCP-RAD and TCP-CO

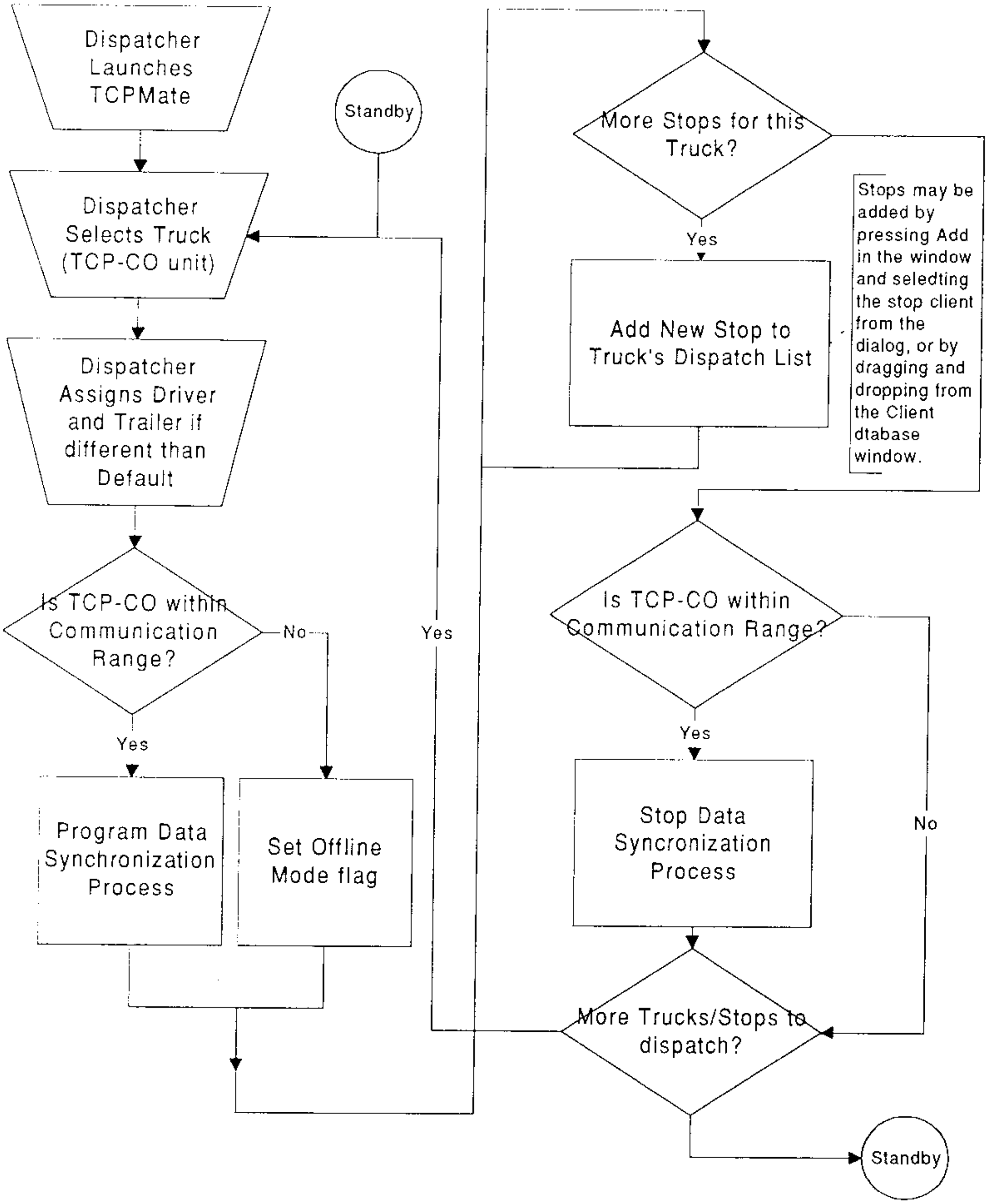
The TCP-RAD will automatically determine when communications is established whether changes have been made to the program information (at either side) and prompt the dispatcher to update the out-of-date information. Synchronization will be automatic if the dispatcher responds positively to the prompt. Synchronization will follow the following steps:


ProgramChanges Made at TCPMate since last Communication ?
No
Entries in TCP-CO audit trail since last communication?
No
Yes
Read New Program Data
Yes
Entries in TCP-CO audit trail since last communication?
No
Download Program Data
Yes
Read new entries in audit trail to temporary buffer
Audit trail value desired value?
No
Download TCPMate values
Yes
Read New Program Data
Done

C. Operations Mode

Normal Operations

The dispatcher will use the TCP-RAD to define the stops for each truck for the activity period. The operation of the TCP-RAD/TCP-CO system is outlined in the following flow diagram:


Dispatcher Launches TCPMate
Standby
Dispatcher Selects Truck (TCP-CO unit)
Dispatcher Assigns Driver and Trailer if different than Default
Is TCP-CO within Communication Range?
No
Yes
Yes
Program Data Synchronization Process
Set Offline Mode flag
More Stops for this Truck?
Yes
Add New Stop to Truck's Dispatch List
Stops may be added by pressing Add in the window and seledting the stop client from the dialog, or by dragging and dropping from the Client dtabase window.
Is TCP-CO within Communication Range?
Yes
No
Stop Data Syncronization Process
More Trucks/Stops to dispatch?
Standby

Default Assignment Definition

A default trailer and list of drivers for each TCP-CO will be available from user specified entries in the Truck definition screen. This data will be stored in the header of the Truck configuration file. When a specific unit is selected by the dispatcher via the Truck combo box initially in a new time period, the Driver and Trailer fields will be filled with the default values assigned in the Truck's file. This value may be overriden by the user of the TCP-RAD at any time.

Offline Definition of Stops

The dispatcher will be able to add new stops to a truck's list without being currently in range of the truck. When the truck comes into range and communications is initiated, the truck's current data will be read, any stops not yet complete will be added to the current list, and the dispatcher will be notified of the incomplete stops. The dispatcher will be given the opportunity to edit the stop list before downloading the new data. If the truck does not have any incomplete stops, all stop information will be downloaded as defined offline. All completed stop information will be written to the daily activity file in either case. In addition, the ability to retrieve a list of stops from any previous dispatched period for any truck, and insert them into the currently active unit's list will be considered for future expansion. This feature would allow cyclic routes to be handled easily. The source of the list of stops for the truck would be the activity log for the period desired, and hence would be limited in time to the oldest activity log not purged.

D. Data Retrieval

The stop data returned by the truck after completion of an activity period will be stored verbatim in a flat ASCII text file on disk. It will be named according to the following standard:

MMDDYYXX.LOG

where MM is the numeric month, DD is the day of the month, YY is the last two digits of the year, and XX is the shift indicator for the current activity period. For the first shift, XX will be absent from the filename, hence for single-shift per day operations, or multiple-day operations, filenames will not contain any shift indicator data. A new file will be started for each new activity period, and all data collected during that period will be appended to the same file. All stop data returned will be stored in the .LOG file associated with the activity period in which the data was RECEIVED BY THE DISPATCH COMPUTER. Flexibility will be given in the defining of the activity period to allow for situations such as trucks waiting until the next morning to submit the data to the host, even though there is no significance in the synchronization of .LOG file name and the actual date the stop was made; the stop date/time will be a part of the record structure within each line of the .LOG file. The truck ID will be a part of the stop data, as well, and all information from all trucks will be appended to the same .LOG file for that activity period. No sorting will be performed; that will be left to whatever method the dispatcher chooses for post-processing of the data.

E. Error Handling

In the event that communications is not available with a TCP-CO unit but the stop data is required, or stop data becomes lost by a hardware failure, incomplete stop data can be forced into the activity file, with zero totals and live values, to be modified at a later time by the user from the application of choice. NO MODIFICATION of stop data, other than deleting defined stops before downloading (removing from the list for a truck), will be handled by the TCP-RAD.

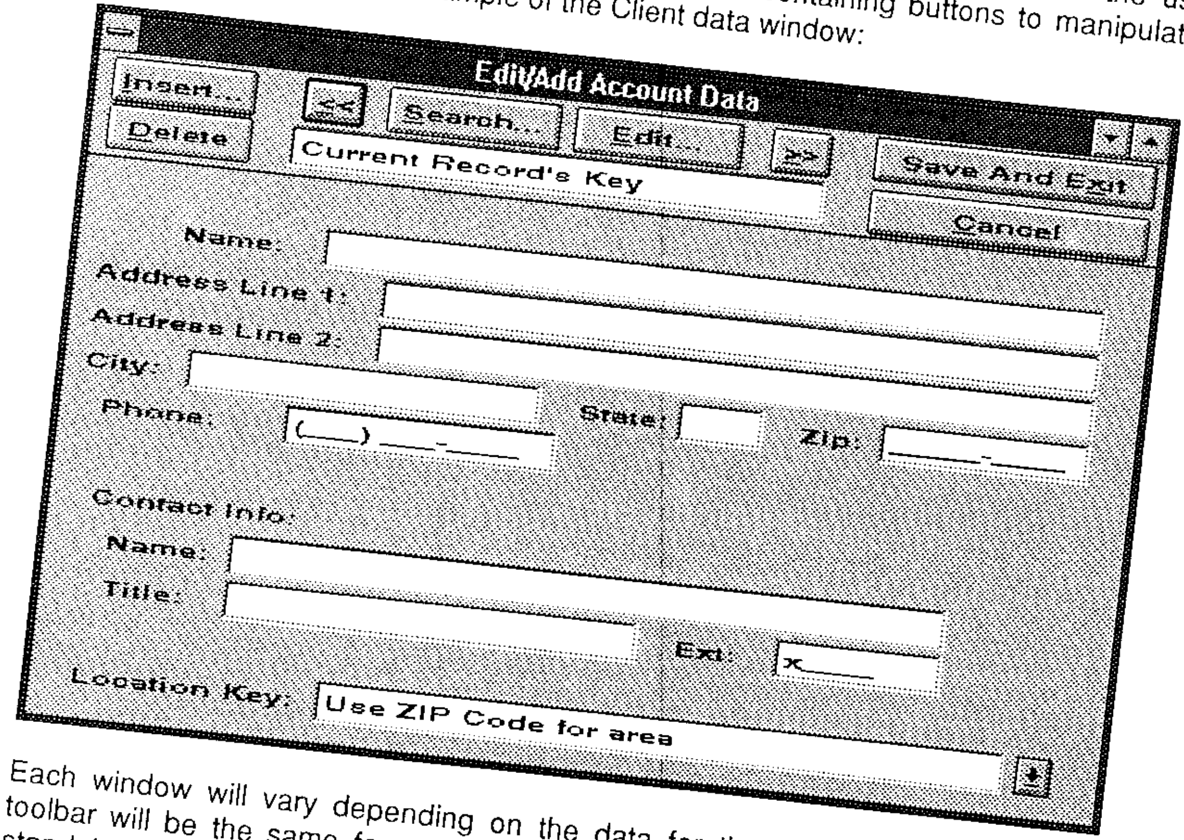
V. TCP-RAD Internals

A. Data Organization

Data will be organized into the following structures:

- Truck data (Truck ID, Communications address, trailerID, driver ID, TCP program data, etc. (Note: the Truck data will compose the files accessed via File menu operations New, Open, Save and Save As...)
- Client data (Customer information: ID, name, address, etc.)
- Tank data (client ID, Tank ID, capacity, etc.)
- Driver data (name, license data, etc.)
- Stop data (unique stop number, status, custody transfer date/time, client ID (lessee), tank ID, truck & trailer ID, driver, volume totals, etc.)

The Truck information will be the exception in that this will be a binary file containing the configuration file information for the TCP-CO. The remainder of the files are described by the following: Each list of entries of the above structures will be maintained as a flat text file on disk, and a linked list in memory. Each type of structure will be associated with a unique window allowing the viewing, addition, changing, and deletion of record entries of that structure type. Each of the above structures will also contain a pointer to a free-form text entry for additional descriptive information. The application code to interface each of these unique windows and allow the user to interact with them will be via a common toolbar containing buttons to manipulate the record. as shown in this sample of the Client data window:



Location Key: Use ZIP Code for area

Each window will vary depending on the data for that particular structure, but the toolbar will be the same for each window. Each window will be able to be invoked standalone; that is, if the function requiring interaction with the record is known beforehand, (i.e., the user is adding a new record to a database) the window can be presented with only OK/Save and Cancel buttons available in the toolbar, and the action will be taken upon the OK button being pressed. If the menu choice is more generic, (i.e., Edit Client Database...), the window can be presented with the full functionality of the toolbar enabled. In this case, all actions will be taken immediately upon the leaving of a record. Changes and deletions will require the user to respond positively to a confirmation dialog box before the action is completed (This verification prompt will be subject to a configuration choice allowing the user to disable prompting before deletion or modification).

The Client record and its associated window will take advantage of a feature that will be included in the TCP-RAD, the ability to handle 'user-modifiable records'. There will be a method for the user to add additional fields to the Client record by specifying a label and maximum size of the field (limited by application constraints). The Client window will adapt to present these new fields to the dispatcher when working with the client database. The user has no ability to position fields within the window. These user-defined fields will always be alphanumeric data fields, and no other field types will be supported. This feature will be designed in a general manner, so that any other database record where this feature may be useful could incorporate the functionality.

B. Database Record Definitions

The following records will be those used within the database .TBL files. Each record will have one or more key data fields that will be required for TCP-RAD/TCP-CO operation. In addition, records will contain user-defined fields to allow for more flexibility between various end users of the TCP-RAD/TCP-CO product.

Trailer Data Record

Field	Label	Data Type	Length{/Range}
1	ID	Alpha	20
.	User Defined	User Defined	User Defined
n	User Defined	User Defined	User Defined

Driver Data Record

Field	Label	Data Type	Length{/Range}
1	Name	Alpha	30
.	User Defined	User Defined	User Defined
n	User Defined	User Defined	User Defined

Client Data Record

Field	Label	Data Type	Length{/Range}
1	Client ID (lessee)	Alpha	20
2	Client Name	Alpha	30
5	User Defined	User Defined	User Defined
.	User Defined	User Defined	User Defined
.	User Defined	User Defined	User Defined
n	User Defined	User Defined	User Defined

Tank Data Record

Field	Label	Data Type	Length{/Range}
1	Tank ID (unique)	Alpha	20
2	Tank Leaseholder (Client ID)	Alpha	30
3	Max Capacity	Numeric	6/1-999999
4	User Defined	User Defined	User Defined
.	User Defined	User Defined	User Defined
n	User Defined	User Defined	User Defined

Defined Stop Data Record

Field	Label	Data Type	Length{/Range}
1	LeaseHolder (Client ID)	Alpha	20
2	Tank ID	Alpha	20

Completed Stop Data Record

This record is used to communicate between TCP-CO and -RAD and is specified in The TCP-CO External Communications specification. See appendix A for a reference.

C. File Formats

The following standard header will be used for binary files except where noted otherwise:

Standard Header: 1 word - file type identifier unique to each file type
1 word - file revision information
30 bytes - description of file (i.e. TCP-RAD Configuration File)
30 bytes - copyright information
©1997 FMC Smith Meter, Inc.

Text fields in header will be null padded to the specified size.

The various files required for the TCP-RAD operation are listed below, along with the format used within the files. Not all files required for TCP-RAD operation are listed, only those specific to the data unique to this application.

.TRK files - Truck Definition/TCP Configuration Files (Truck Info, TCP-CO data)

This file will be in a binary format. After the header will be a Truck definition record, containing the ID, license info, and other information specific to the truck. Following this record will be a series of program entries directly representing the values programmed into the TCP-CO. Each entry will consist of a structure, the first element being the program code identifier, and the second element being the actual value of the program code.

.REC files - Data Table Record Definition Files

Proprietary binary format - After the standard header there will be a structure defining the particular table (table name, whether duplicate keys are allowed, whether multiple keys are to be used, etc). Following the table definition structure will be a variable number of field definition structures defining each field in the table's record.

.TBL files - Data Table Files

ASCII text format (No standard header)

a) ***Client table***

Series of Client (lessee) record enties, whitespace between fields, and a <CR><LF> pair separating each record. The first line will contain the field names.

b) ***Tank table***

Series of Tank record enties, whitespace between fields, and a <CR><LF> pair separating each record. The first line will contain the field names.

c) ***Driver table***

Series of Driver Record entries, whitespace between fields, and a <CR><LF> pair separating each record. The first line will contain the field names.

d) ***Trailer table***

Series of Trailer Record enties, whitespace between fields, and a <CR><LF> pair separating each record. This list will echo the available defined trailers for which there are .TRL files created. Adding a new trailer to this table will create a new .TRL file initialized to default values. The first line will contain the field names.

.RTE files - Truck Stop List files

ASCII text format (No standard header)
Series of Defined Stop Record entries, whitespace between fields, and a <CR><LF> pair separating each record. The first line will contain the field names.

.LOG files - Completed Stop Data Archive Files

ASCII text format (No standard header)

Series of Competed Stop Record entries, whitespace between fields, and a <CR><LF> pair separating each record.

.RPT files - Configurable Report Definition files

This file will be in binary format. After the header will be a report header containing the report name and pertinent flags/info followed by a series of report entry definitions, followed by a table of null-terminated user-defined text entries. These entries may refer to stop record data, TCP program code data, TCP program code descriptions, or user-defined text entries.

VI. Summary

The TCP-CO hardware and TCP-RAD together will comprise a system capable of tracking a number of trucks during the custody transfer steps of pickup (and offloading) of crude oil collected from client tanks in the field. The TCP-RAD will handle configuration of TCP-CO parameters and synchronization of a client database, as well as the dispatching of a list of stops on a periodic basis, be it a shift, a day, or one at a time. Upon return by the truck from the stop(s), volume and other accumulated data will be offloaded by the TCP-RAD system computer. This data will be stored in a flat text file for a period of time determined by the user, and available for import into other applications at the discretion of the user.

VII. *Appendix 1: Route Stop Record Format*

Field	Label	Data Type	Max Length {Range}
1	Truck ID	Alpha	20
2	Transaction Number	Numeric	12
3	Lease ID	Alpha	20
4	Tank ID	Alpha	20
5	Recommended LACT ID	Alpha	20
6	Load Date	Date	10
7	Seal Open Time	Time	8
8	Seal Closed Time	Time	8
9	Load Average Temp	Numeric	8
10	Load Avg Dens	Numeric	8
11	Load Avg BS&W	Numeric	8
12	Volume - Indicated	Numeric	12
13	Volume - GRS	Numeric	12
14	Volume - GST	Numeric	12
15	Volume - NSV	Numeric	12
16	Mass	Numeric	12
17	Unauthorized NSV	Numeric	12
18	CTL	Numeric	12
19	CSW	Numeric	12
20	CCF	Numeric	12
21-40	Prompt #1 - #20 Response	Alpha	User Defined
41	Report To Use	Numeric	2 {01-16}
42	Comment Field	Alpha	Variable
42+n	User Defined Field n	Alpha	Variable

Note: User-Defined Fields are any TCP-RAD database record fields, fixed or user-defined, NOT ALREADY INCLUDED IN THE STOP DATA RECORD that are selected by the dispatcher to be downloaded as part of the stop definition. There will not be a limit to the number of entries, but the overall size will limited to the maximum size of a stop record (1K).

The stop record will be transmitted as a series of null-delimited strings. Leading and trailing whitespace in each field will not be transmitted, but whitespace is permissible within each field otherwise. Unused or empty slots (or whitespace-only) in the stop record will still contain a placeholder null. For example, an empty field would be represented as follows (prevField is the data in the field prior to the empty field and nextField is the data in the field after the empty field) :

...<NULL>prevField<NULL><NULL>nextField<NULL>...

FMC **Smith Meter Inc.**
An FMC Corporation subsidiary

TCP-CO
External Communications Specification

July 21, 1997
Revision 0
Project # RS050197

Prepared by: ______________________
James M. Pettinato - Associate Software Engineer

Approved by: ______________________
Robert M. Smith - Project Engineer

Approved by: ______________________
Constance L. Gibbons - Sr. Software Engineer

Approved by: ______________________
Dale A. Bohman - Product Marketing Manager

Approved by: ______________________
David P. Resch - Manager
Electronic Products and Automation

Communications Stack Layers

Hardware Layer

This layer may be manifested by several means. In the TCP application, if there is a hard-wired connection between the TCP-CO and the host, this layer is simply the cable connecting the UARTs of each device. For a radio transceiver, there will be additional hardware required to implement the modulation/demodulation of the RF signal. This layer, in other words, is responsible for converting a serial data packet into a radio signal (IR, etc.) that can be transmitted to and received by another hardware layer device. Therefore the hardware layer could also consist of this alternate hardware that will be used, in conjunction with the packet protocol layer, for RF communications.

Protocol Layer (Packet Layer)

In order to efficiently utilize a transport medium such as radiotransmission, where it is assumed there will be high potential for error due to interference or range limitations, the system will provide a robust error-tolerant protocol layer. This layer provides for the insertion of a packetization function which proves useful in such a transport medium in two ways: it minimizes the impact transient interference has on the overall message by limiting the size of required retransmission of data if said data contained an error, and it allows for the inclusion of a repeater transceiver without extensive software interaction. This layer will reside in the driver code resident on the system running the application. The interface (input and output specification) will be designed in such a manner so that (1) the application layer will not need to be aware of the actual protocol in use; and (2) the protocol layer will provide any interaction required with the hardware layer. This functionality will be hidden from the application layer, allowing the packet layer to perform status checking and other functions such as battery monitoring that may be specific to the currently active hardware layer. Any packet protocol may be used in conjunction with any hardware layer-its output will be simply to place characters in a UART transmit buffer. The application, however, must enforce the proper pairing of protocol layer to hardware layer for efficient and sensible operation. *Note: this pairing will initially be the responsibility of the user configuring the TCP-CO; in the future some method of detecting a combination that is undesirable may be added to the application. Note that even undesirable combinations would still function as expected, although not very efficiently.*

Application Layer

The application layer consists of the actual application tasks on the TCP-CO or the host. The unique aspect of this layer is operation will be independent of the transport medium, i.e. radio communications, IR communications, hard-wired, etc. All communications will be achieved via a virtual interface consisting of a fixed set of functions allowing transmitting, receiving, status checking, etc. with the current communications driver. Hence, the application layer need be concerned only with the specific function command to be executed and any arguments it may require, and also must be prepared to handle all possible responses to the command.

Command Language

The external communications interface of the TCP-CO will be defined by a command format that consists of the following functions:

- 'Ping' or request an acknowledge (response should be an ACK)
- Ack
- Request status information (host to TCP-CO only)
- Report Status
- Upload/Download individual program code entries
- Upload/Download complete program code database
- Request Stop X data (response should be a Transmit Stop X data Command)
- Transmit Stop X data (response should be an ACK)
- Delete Stop X data command (response should be an ACK)
- Upload/Download configurable report definitions
- Error Message

For transmissions initiated by the host, it will be up to the application layer on the receiving TCP-CO to be able to determine which type of command has been received and route it to the appropriate application task to handle the command and invoke the command layer to make the appropriate response. For transmissions initiating on the TCP-CO, the host will immediately take control by sending a status request, hence the initial response to any communications initiated by the TCP-CO will be a status request from the host.

Packet Layer (Radio Protocol)

This protocol will be capable of dividing a message up to 65355 bytes in size into packets that will be transmitted individually, and to receive a similarly sized message that has been so divided. The packet protocol will divide the message into sections not to exceed 248 bytes and transmit the sections in order in packets not to exceed 256 bytes. (For RF, the optimal packet size is smaller, so overall packet size will be limited to 40 bytes.) The packet frame will follow ISO 3309 with the following exceptions: (1) A second address byte will *always* be present and will represent the FROM address; its existence will not be dependent on the first (low order) bit of the first address byte as specified in ISO 3309 – § 4.1. (2) The FCS (Frame Checking Sequence) will be extended to 32 bits as allowed in 3309 for future requirements (§3.6 note 1).

Sample Radio Protocol Packet Frame:

STX	TO	FROM	SEQ#	SIZE	DATA	FCS (CRC32)			
0x02	0x01	0x15	0x00	0x05	nnnnn	MSB	.	.	LSB

The To address will always be the address of the unit intended to receive the communication. The From address will always be the address of the unit originating the communication.

The sequence number is based solely on the previous transmitted sequence number for this transmitter and is not dependent on TO address or message position. The sequence number will have the range of 0-255, and will wrap back to 0 when incrementing from 255. The receiver will always expect the next sequence number except when looking for the first packet of a message, where any sequence number is valid except that a repeat of the previous sequence number within 3 seconds of a successful packet reception will be considered a repeat packet. The receiver will acknowledge the repeat packet, but not act upon it in any other way. The receiver will wait for the next expected packet at least 8 seconds before timing out during a multiple packet message.

The size byte for the packet indicates the size of the data following the size byte, not including the CRC32. As mentioned above, the maximum size for this value will be 248, making the maximum packet size 256 (after stripping the STX). Again, for radio or other noise-laden medium, this size will be limited to 32, making the overall packet size limit 40.

The data field will contain a message or some portion of a message to be recreated at the receiving unit.

Each packet will be acknowledged by the receiving unit via a response packet with the identical sequence number, to/from addresses swapped, a size of 0, and no data bytes. This packet acknowledge is unique and differs from the Acknowledge command described in the command table, which acknowledges a complete message. If the hardware layer is error-prone (i.e., radio or IR link), in the situation where no packet acknowledge is received, the transmitter will attempt to resend up to 5 times with a random delay of 0.5-1.5 secs between each attempt. If the packet is not acknowledged after the final try the transmitter will consider the transmission of the entire message containing that packet a failure. Regardless of success or failure, for the next packet, the transmitter will always increment the sequence number. Each packet may contain a portion of a message or an entire message.

Message (Command) Format

The message format used to transmit commands and responses will be as follows:

Sample Message:

COMMAND	SIZE (MSB)	SIZE (LSB)	Command Arguments	Command CRC32
0x01-0x63	0xNN	0xNN	(optional)	(optional)

The command byte will be one of the commands defined in the following section, *Command Language*. The size word will be the size of the data starting with the first character after the size word up to AND INCLUDING the CRC32 if it exists for this message. (Note: this differs from the packet protocol size argument, which does not count the CRC32 as part of the size.) The command arguments will be dependent on the type of command, and such are described in the next section. The CRC32 associated with the message here is unique from the CRC32 of the individual packets, and its presence is dependent on the size of the message... any message with a size > 64 bytes will have a CRC32 appended to the message. This CRC32 will be calculated upon the data beginning with the first character after the size word until the end of the data.

Command Language

Command 1 - Individual database element read

Data to follow size word: Register ID - 4 bytes
Byte 1 - data type
Byte 2 - function
Byte 3 - subfunction
Byte 4 - offset
Good response: Command 2 - Individual database write command
Error response: Command 9 - Error Response

Command 2 - Individual database element write

Data to follow size word: Register ID (see cmd 1) then data value of register
Good response: Command 6 - Ack
Error response: Command 9 - Error Message

Note: A Write command issued in response to a Read command should not expect a response.

Command 3 - block read of database values

Data to follow size word: None
Good response: Command 4 - Block Write command
Error response: Command 9 - Error Message

Command 4 - block write of database values

Data to follow size word: Database data as resides in TCP-CO memory
Good response: Command 6 - Ack
Error response: Command 9 - Error Message

Note: No response if this command was issued in response to a Block Read command.

Database will be segregated by read-only status, i.e., all read-only values will be at the end of the physical structure in memory. All writable members will be contiguous, hence a block of data transferred as a BLOB (Binary Large OBject) will allow a complete program definition to be transferred between the PC and the TCP-CO.

Command 5 - 'Ping' (Ack request)

Data to follow size word: None
Response: Acknowledge command

Command 6 - Acknowledge (Ack)

Data to follow size word: None
Response: None.

Command 7 - Status Request

Data to follow size word: None
Response: Command 8 - Status Response

Command 8 - Status Response

Data to follow size word: 20 byte status. Bytes 1-16 represent the status of stops 1-16 respectively, bytes 17-20 are system status bits.
Response: None.

<u>Status bits for Stop #X</u>

Status byte (x) & 1 = 1	Stop Defined at position X
Status byte (x) & 2 = 2	Stop X completed
Status byte (x) & 4 = 4	Stop X Aborted
Status byte (x) & 8 = 8	Stop X offloaded
Status byte (x) & 16 = 16	reserved for future expansion
Status byte (x) & 32 = 32	“
Status byte (x) & 64 = 64	“
Status byte (x) & 128 = 128	“

<u>Status bits for TCP-CO System</u>

Status byte 1 & 1 = 1	Ready for shift start operations
Status byte 1 & 2 = 2	Ready for shift end operations
Status byte 1 & 4 = 4	Program mode change occurred
Status byte 1 & 8 = 8	reserved for future expansion
Status byte 1 & 16 = 16	“
Status byte 1 & 32 = 32	“
Status byte 1 & 64 = 64	“
Status byte 1 & 128 = 128	“

Status byte 2 & 1 = 1	Alarm condition
Status byte 2 & 2 = 2	reserved for future expansion
Status byte 2 & 4 = 4	“
Status byte 2 & 8 = 8	“
Status byte 2 & 16 = 16	“
Status byte 2 & 32 = 32	“
Status byte 2 & 64 = 64	“
Status byte 2 & 128 = 128	“

Status byte 3 reserved for future expansion
Status byte 4 reserved for future expansion

Command 9 - Error Message

Data to follow size word: 1-byte error code
Response: None.

Error Code Values

0 - No error
1 - Invalid command byte
2 - Invalid command parameters
3 - In Program Mode
4 - Busy (flowing, printing, etc.)
5 - Alarm condition
6 - Insufficient security level access for this function
7 - Requested stop not available
8 - Stop position already in use
9 - Operation out of sequence
10 - Option not available
11 - CRC error in received message

Command 10 - Write Message to TCP-CO display (Not in rev 0)

Data to follow size word: string length word (MSB first, include CRC)
Up to 256 character message string

Good Response: Acknowledge command.
Error response: Command 9 - Error Message

Command 11 - Read Configurable Report Definition

Data to follow size word: None
Good response: Command 12 - Write Configurable Report command
Error response: Command 9 - Error Message

Command 12 - Write Configurable Report Definition

Data to follow size word: up to 65535 byte report definition

Good response: Acknowledge command
Error response: Command 9 - Error Message

Commands 13-15 - Reserved for future expansion

Command 16 + X - Read stop record #X

Data to follow size word: None
Good response: Command 32+X - Write stop record #X
Error response: Command 9 - Error Message

This command instructs the receiving unit to transmit stop information for the stop in the route array in position X where x is 0-15.

Command 32 + X - Write stop record #X

Data to follow size word: Stop record (variable length as defined by user)
Good response: Command 6 - Ack
Error response: Command 9 - Error Message

Note: No response if this command was issued in response to a Read Stop command.

This command will be followed in the message buffer by a valid route stop record as defined in Appendix 1 of this document. The stop will be inserted into the TCP-CO's route information at position X where x is 0-15.

Command 48 + X - Delete stop record #X

Data to follow size word: None
Good response: Command 6 - Ack
Error response: Command 9 - Error Message

This command instructs the receiving unit to remove the stop at position X from the route information, where x is 0-15.

Sample Commands

Sample of individual database element READ command transmission (fits in single packet):

STX	Dest	Src	Sq.#	Size	Command	Sz-Hi	Sz-Lo	Register ID	checksum
0x02	0x01	0x15	0x00	0x07	0x01	0x00	0x04	0x06 0x02 0x01 0x01	CRC32

Note: first (packet) size field is packet size not including CRC32, second size word is command argument size.

Sample of individual database element Write command: (also single packet)

STX	Dest	Src	Sq.#	Size	Command	Sz-Hi	Sz-Lo	Register ID	Data	chksm
0x02	0x01	0x15	0x00	0x08	0x01	0x00	0x05	0x06 0x02 0x01 0x01	0xNN	CRC32

Note: data field format determined by data type - first byte in register ID specifies type

Sample Read Stop #1 command:

STX	Dest	Src	Sq.#	Size	Command	Sz-Hi	Sz-Lo	chksm
0x02	0x01	0x15	0x00	0x03	0x01	0x00	0x00	CRC32

Sample Delete Stop #3 command:

STX	Dest	Src	Sq.#	Size	Command	Sz-Hi	Sz-Lo	chksm
0x02	0x01	0x15	0x00	0x03	0x01	0x00	0x00	CRC32

Note: the commands for stop interaction are as follows:

0x10 plus the zero-based array index to read a stop
0x20 plus the array index to write a stop
0x30 plus the array index to delete a stop

Appendix 1: Route Stop Record Format

Field	Label	Data Type	Max Length {Range}
1	Truck ID	Alpha	20
2	Transaction Number	Numeric	12
3	Lease ID	Alpha	20
4	Tank ID	Alpha	20
5	Recommended LACT ID	Alpha	20
6	Load Date	Date	10
7	Seal Open Time	Time	8
8	Seal Closed Time	Time	8
9	Load Average Temp	Numeric	8
10	Load Avg Dens	Numeric	8
11	Load Avg BS&W	Numeric	8
12	Volume - Indicated	Numeric	12
13	Volume - GRS	Numeric	12
14	Volume - GST	Numeric	12
15	Volume - NSV	Numeric	12
16	Mass	Numeric	12
17	Unauthorized NSV	Numeric	12
18	CTL	Numeric	12
19	CSW	Numeric	12
20	CCF	Numeric	12
21-40	Prompt #1 - #20 Response	Alpha	User Defined
41	Report To Use	Numeric	2 {01-16}
42	Comment Field	Alpha	Variable
42+n	User Defined Field n	Alpha	Variable

Note: User-Defined Fields are any TCP-RAD database record fields, fixed or user-defined, NOT ALREADY INCLUDED IN THE STOP DATA RECORD that are selected by the dispatcher to be downloaded as part of the stop definition. There will not be a limit to the number of entries, but the overall size will limited to the maximum size of a stop record (1K).

The stop record will be transmitted as a series of null-delimited strings. Leading and trailing whitespace in each field will not be transmitted, but whitespace is permissible within each field otherwise. Unused or empty slots (or whitespace-only) in the stop record will still contain a placeholder null. For example, an empty field would be represented as follows (prevField is the data in the field prior to the empty field and nextField is the data in the field after the empty field) :

...<NULL>prevField<NULL><NULL>nextField<NULL>...

Appendix 2: Recommended Application Layer Interface

The following is an example implementation that describes the division in functionality in the protocol layer and the application layer. It is intended merely as an example and is not necessarily representative of the final implementation.

The application will interact with the communications stack via a well-defined interface providing access to all of the required services provided by the communications protocol. The interface will at a minimum provide the following functions to the application layer:

Initialize Communications Stack (select protocol, various communications parameters)
Send Message via current packet protocol (transmit command)
Get Message via current packet protocol (receive command)
Status of Communications Stack (Retrieve status, error information, error text)

The actual interface may be enhanced or hidden by higher level functions on each platform, but at minimum the above requirements will be implemented as functions that will accept the following arguments and return the specified information:

Initialize Communications Stack

int InitComm(int protocol, COM_OPTIONS pOptions);

arguments:

protocol - integer indicating the selected protocol in the available list. For initial release of TCP-CO, the available protocols will be 0-none, 1-Direct Connect and 2-Radio Packet Protocol, different only in maximum size of the data field in the packet (248 for direct connect, 32 for radio).

pOptions - structure with the following information: port to use, baud, data, and parity settings, local address, and other information required to define the communications setup.

Return value:

0 on success, nonzero integer indicating the error on error. Error information may be retrieved by the ComErrMsg() function implementation.

Send Message via current protocol

int SendMessage(char *buf, unsigned int msgSize, int destination_address);

arguments:

buf - character buffer containing message to be sent

msgSize - Unsigned integer value containing number of characters to send

destAddr - integer specifying destination for this transmission.

Return value:
0 on success, nonzero integer indicating the error on error. Error information may be retrieved by the ComErrMsg() function implementation.

Get Message via current protocol

unsigned int GetMessage(char *buf, int maxSize, unsigned int waitTime);

arguments:

buf -	character buffer to hold response
maxSize -	Unsigned integer value containing maximum number of chars to write before returning to avoid exceeding size of *buf*
waitTime -	If zero, return immediately with current status. If nonzero, wait waitTime milliseconds or until a completed message is available before returning.

Return value:
0-no complete message available
1-65534 - size of successfully returned message
65535 - Overrun or other error - use GetComError() for more information

Status of Communications Stack

int GetComError(void);

arguments:
none

Return value:
0-no error present
nonzero - indicates an error

char *ComErrMsg(int errorID);

arguments:

errorID -	Value returned by GetComError. If -1, returns message associated with the last known status of the communications stack.

Return value:
pointer to a character message describing the error associated with *errorID*