PFCS Vendor Specifications

Table of Contents

Document Revisions

Doc. Rev. Date	Doc. Rev. Number	Doc Rev. Description	Page No. Revised
6/1/98	1	Updated document to conform to Chrysler Book of Implementation standards; corrected grammar	ALL
1/1/2000	2	Section 1.2.1 In the figure the message type was printed with "00003", corrected it to "0003"	5
1/1/2000	2	Section 1.3, Added new request type "0006" into spec's	7
1/1/2000	2	Updated Section 1.5 SPD Vendor Information. SPD record layout all in one section, including PFCS header.	11,12
1/1/2000	2	Updated Section 1.6.2 From where to obtain NIC part number for PC's. In Software Requirements, few minor changes were made.	15
1/1/2000	2	Updated Section 3.3.2, Corrected Start and End values for the last two rows of the table.	29
6/1/2000	3	Updated Section 1.4.1. Error codes table updated and new error codes (from PFD) table added.	9
6/1/2000	3	Cover page added	i
6/1/2000	3	Note added	6
6/1/2000	3	Chapter 1.8 PFD Buyoff Checklist removed	20
6/1/2000	3	Chapter 4 Testing Vendor's Equipment added	30
6/1/2000	3	1.6.2 TCP/IP Device Connectivity Specifications updated. PFCS DLL agreement.	16
10/24/2001	4	Complete Re-write	ALL

This page intentionally left blank.

1 PFCS Device Categories and Certification Requirement

This section seeks to define the PFCS Device Categories and to give guidelines on the best method of connecting the device to PFCS. All OEM devices expecting to communicate to DaimlerChrysler Plant Systems through PFCS will be required to complete the PFCS Certification Process, at a designated DaimlerChrysler facility, before the OEM equipment can be installed in the plant. The PFCS Certification Process is covered in *Section 5* of this document.

1.1 PFCS Device Categories

PFCS Devices fall into three general categories: Process Equipment, Multi-Spindle Devices, and Single Spindle Devices.

1.1.1 Process Equipment

- All process equipment will be Ethernet connected using the TCP/IP protocol.
- Process equipment is described as a device with multi-function operation designed to perform a specific set of tasks. This may include torque, fluid fills, functional testing, sub-assembly, etc.
  - PC, PLC, or custom processor based.
  - Typically fix mounted – not portable.
  - Download of vehicle data supported.

1.1.2 Multi-Spindle Torque Tools

- All multi-spindle torque tools will be Ethernet connected using the TCP/IP protocol.
- Multi-spindle torque tools are described as a devices incorporating multiple torque spindles to rundown several fasteners simultaneously.
  - PC, PLC or custom processor based.
  - Typically designed for a specific function.
  - Typically fix mounted – not portable.
  - Download of vehicle data supported.

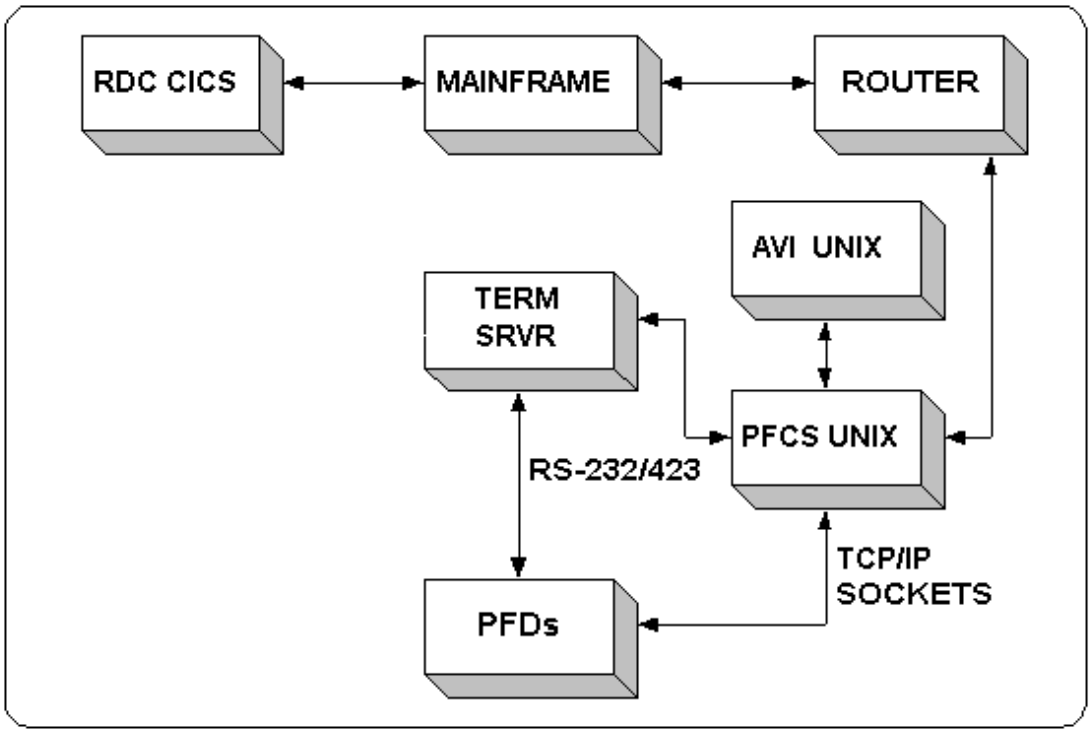
1.1.3 Single Spindle Torque Tools

- Single spindle torque tools can be Ethernet or RS232 connected.
- Single spindle torque tools are described as devices incorporating a single nut-runner used to attach single or multiple fasteners.
  - PC, PLC or custom processor based.
  - Typically a generic design. Tool can be used in several different functions.
  - Typically a hand held device.
  - Typically portable. Easily moved to other locations.
  - Vehicle specification information not available.

2 Vendor Communication

This document describes protocols required for a vendor-supplied plant floor device (PFD) to communicate with the Plant Floor Communication System (PFCS). These protocols affect the two-way conversational link used to send and receive data between PFDs and PFCS. Vendor devices may be connected to PFCS using the following options:

- RS-232 connecting (a single spindle torque tool) to a DaimlerChrysler-supplied terminal server using:
  - Data format: 8 bit ASCII
  - Speed: 9600 BPS
  - Parity: None
  - Stop Bits: 1
  - Flow Control: None
- Ethernet connection between PFD's and PFCS using TCP/IP sockets



As seen in the data flow diagram above, the major parts of the communication link between PFCS and the vendor system are:

- *RDC-CICS:* Regional Data center/CICS regions in which DaimlerChrysler online applications reside — for example, PFS, Broadcast and Receiving systems.
- *MAINFRAME:* Front-end processor required for MVS-TCP/IP.
- *AVI:* A UNIX platform residing in the plant. The AVI (Automatic Vehicle Identification) System interfaces with other corporate computer systems to gather and report specific vehicle build and style information. This extensive vehicle information is maintained on a real-time basis within an in-plant database. AVI is also closely integrated with plant floor manufacturing machines, to which it delivers this vehicle data. AVI can also gather data from the plant floor. It may store this information in its in-plant database for download to other plant floor machines, and it may report the data to other corporate systems. The AVI system is supported by the APIC (Automatic Product Identification & Control) Systems Group
- *PFCS:* A UNIX platform residing in the plant. PFCS (Plant Floor Communications System) is the corporate communication standard for all Chrysler Assembly plants in the U.S., Canada, Mexico, and Europe. The PFCS system utilizes a client/server architecture incorporating applications residing on an IBM RS6000 server at each plant location, and integrated applications on the corporate mainframe computers. All PFCS applications are developed and maintained by the PFCS team. The system is a critical component required to collect quality data from tooling and process equipment, and deliver this data to upper level mainframe applications. In addition, the PFCS system supplies specific vehicle style and build information to floor devices, and outside just-in-time suppliers. PFCS is an integral requirement for torque monitoring, automated line stops, terminal sequencing, and error proofing.
- *TERM-SRVR:* A terminal server is a protocol converter, which converts from RS232 to Ethernet TCP/IP.
- *PFD:* Plant Floor Devices such as Torque Monitors, Pedal Pushers, Wheel Aligners, Pin Stampers, SPD suppliers, etc.

2.1 PFCS Protocol Overview

The PFCS system is capable of sending and receiving data from a wide variety of plant floor devices.

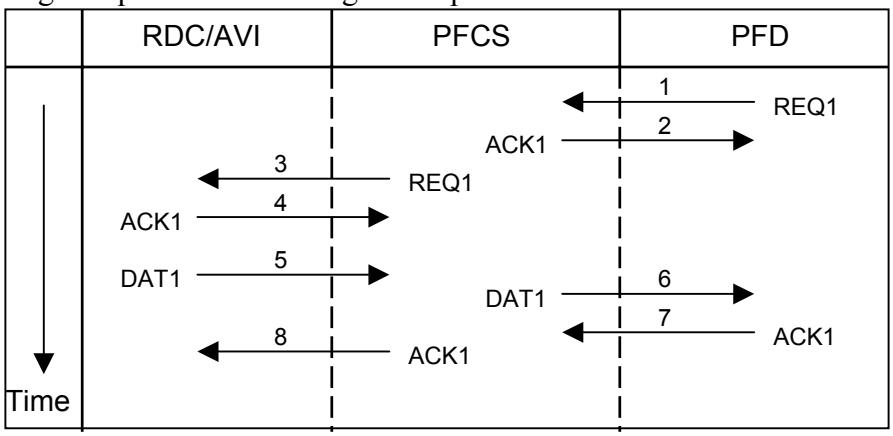
Result data from the PFD to the PFCS system:
The PFD sends data to the PFCS system as each operation is completed or at the end of a specific cycle. The trigger used by the PFD to send the result data packet to the PFCS system is dependent upon the specific physical process within the plant. PFCS will never request the result data from the PFD.

Vehicle build information sent from PFCS to the PFD:
Some PFD's require specific vehicle build information in order to complete an operation. The PFCS system can provide this information automatically when the vehicle reaches a specific point (unsolicited vehicle data), or as the result of a request for vehicle data from the PFD. It is preferred to have the PFD request the specific data for each vehicle (*Section 2.4*)

All messages must contain only printable ASCII characters.

2.1.1 Requests for Vehicle Data

Vehicle Data requests are initiated by PFD's. This section outlines the process of both generating a request and receiving the response.



- Sending Vehicle Data Request to PFCS

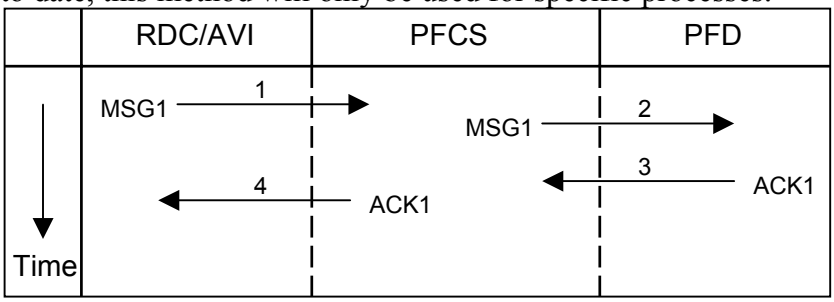
The PFD sends a request to PFCS. The message packet sent to PFCS includes a header, request, and delimiter. PFCS validates the message and sends an ACK (acknowledgement) or a NAK (negative acknowledgement) to the PFD.

- Receiving Vehicle Data from PFCS

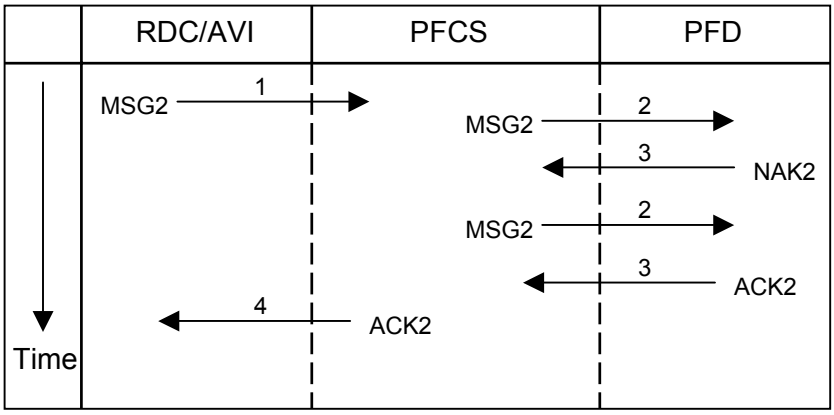
The ACK sent to the PFD indicates that the request was received and will be forwarded to the RDC or AVI application program for processing. At this point, the PFD must wait for a response to the request. After receiving the response data, the PFD must send either an ACK or a NAK, thereby completing the data transfer. PFD's must process NAK's as described in *Section 2.2.2*.

2.1.2 Unsolicited Vehicle Data

Specific vehicle data can be automatically sent to a PFD based upon a predefined status point in the assembly process. As each vehicle passes a specific point, a vehicle data record for the PFD can be created and sent to the device. This requires the PFD to have a local vehicle database, a separate RS232 port or Ethernet socket, and a method to manage the additions, updates, and deletions to the database. Due to the complexity of managing several PFD databases on the plant floor and the difficult task of keeping all the vehicle data up to date, this method will only be used for specific processes.



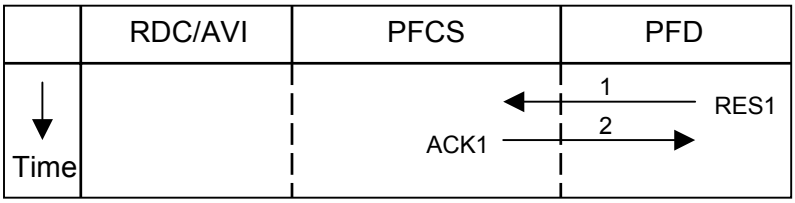
If an ACK is received by PFCS, communications will continue with the next message. Because this is sent unsolicited, the PFD must always be in the correct state to receive data.



If a NAK is received by PFCS, the message may be re-transmitted to the PFD (depending on the error code). This repeated transmission is intended as a means of guarding against garbled data on the link between PFCS and the PFD. PFCS programs would subsequently re-send the message packet to PFD's. If the NAK exceeds a configured limit, then PFCS will log an error, discard the packet, and transmit the next message to the PFD.

2.1.3 Sending Test Results to PFCS

When a PFD has finished its operation, it must send the results of that operation up to PFCS.



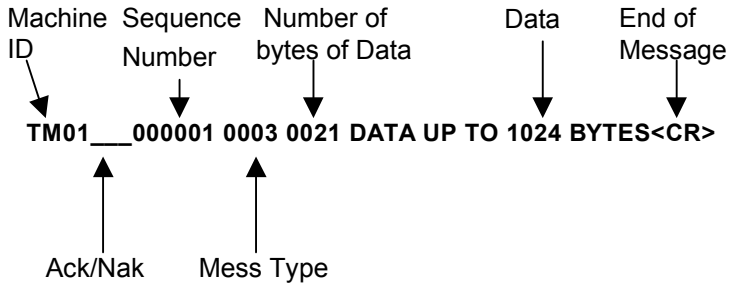
The PFD sends a result packet to PFCS. The packet includes a header, result(s), and a delimiter. PFCS validates the message and sends an ACK or a NAK to the PFD.

2.2 Message Structures

All PFCS messages have the same structure. This section identifies the structure and format of data, responses sent from PFCS, and error types.

2.2.1 Headers, Data, and Delimiters

The message structure is the same for all types of messages.
Messages are sent in the following format:

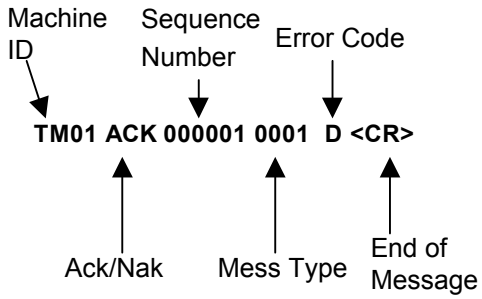


Section	Byte Position	Description
Header	1 – 4	Machine ID e.g.. TM01
	5 – 7	ACK/NAK area
	8 – 13	Message Sequence Number
	14 – 17	Message Type e.g. (0001, 0002, 0003, …)
	18 – 21	Byte count of data in data area of message.
Data	22 – n	Variable data area up to 1024 bytes of data
	n + 1	Message terminator. Carriage return x'0d'

Note: All data is in ASCII format.

2.2.2 ACK and NAK

The ACK sent to the PFD by PFCS says the vendor request was received and will be processed. If the PFD is sending operations results, then it can continue. If the PFD is requesting data, then the PFD should wait for the response packet corresponding to the request. After receiving the response packet, the PFD must send either an acknowledgment (ACK) or a negative-acknowledgment (NAK).
ACK and NAK messages are sent in the following format:



Message	Byte Position	Description
ACK/NAK	1 - 4	Machine ID e.g.. TM01
	5 - 7	ACK/NAK area
	8 - 13	Message Sequence Number
	14 - 17	Echo of Message Type.
	18	Error Code for ACK/NAK.
	19	ACK/NAK record terminator. x'0d'

Note: All data is in ASCII format.

2.3 Machine ID

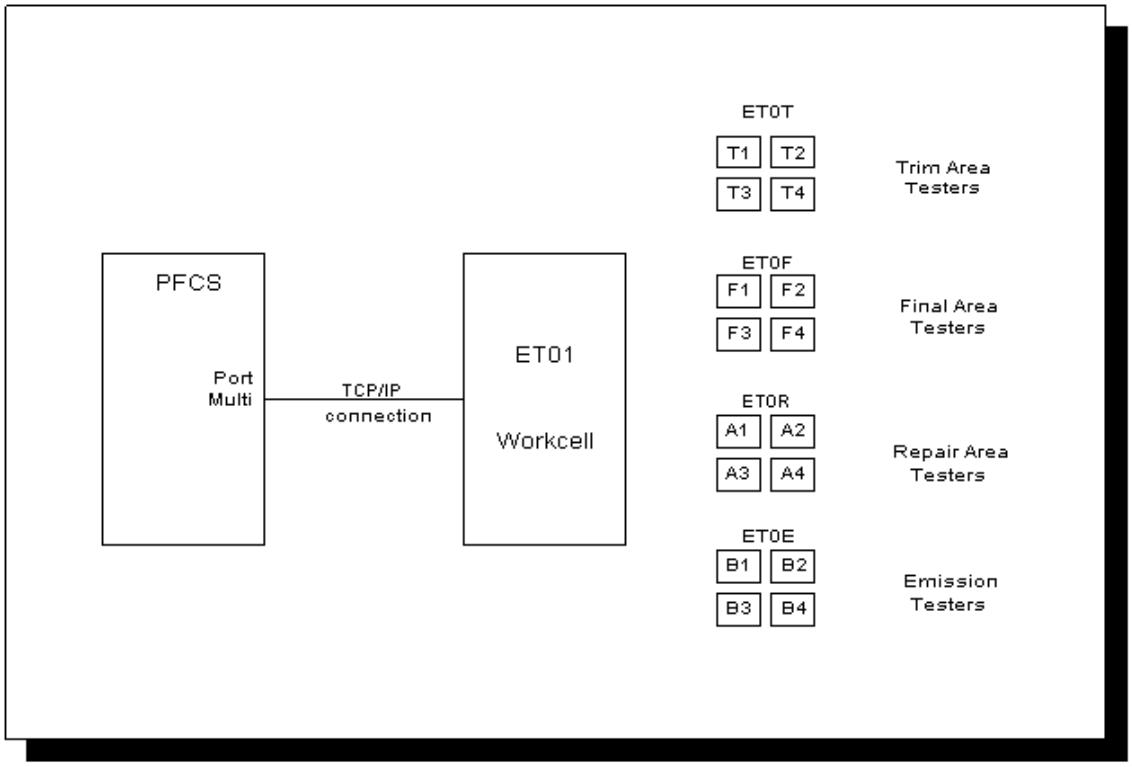
Plant ITM will assign a unique 4 character Machine ID to be used by a PFD for all PFCS communication. This ID must be a configurable option on the controller. Simple torque operations will generally require one Machine ID per connection to PFCS, however, in more complex cases, multiple Machine ID's may need to communicate to PFCS on a single port. Some examples of tools using multiple Machine ID's on a single port are:

- Multi-channel torque controllers
- PFD's that act as a server for more than one tool on the plant floor
- Any process that requires multiple machine ID's

Multiple Machine Requirements:

- The vendor must provide a configuration screen on the controller to individually set up each four-character machine ID.
- One of the machine ID's on the port must be configured as the "main" machine ID.
    - The main machine ID must be used to establish connections to PFCS, and send keep alive messages (message type 9999).
    - If the device requires vehicle build data, the main machine ID is the only device able make requests for and receive vehicle data (message type 0001)
    - The main machine ID can also send test results (message type 0002) to PFCS
- All machine ID's can send test results to PFCS.
- PFCS can also accept keep-alive messages from all machine ID's, if required by the process.
- If operation requires unsolicited data download from PFCS (type 0003), it must receive this download on a separate port.
- While waiting for vehicle data (type 0001) test results cannot be sent
- Due to the complexity of multiple machine ID's on a single port, it is important that Corporate and Plant ITM be involved in the process

Below is a diagram of a typical multi-machine workcell.



The workcell makes VIN specific requests (type 0001) using machine ID ET01, the main machine ID. PFCS ACK's these requests, returns the broadcast data to ET01, and ET01 sends ACK's to PFCS when the data is received.

When sending test results (type 0002) from the Trim Area to PFCS, the workcell uses machine ID ET0T, and the actual tester ID (T1-T4) is sent as part of the test results.

When sending test results (type 0002) from the Final Area to PFCS, the workcell uses machine ID ET0F, and the actual tester ID (F1-F4) is sent as part of the test results.

When sending test results (type 0002) from the Repair Area to PFCS, the workcell uses machine ID ET0R, and the actual tester ID (A1-A4) is sent as part of the test results.

When sending test results (type 0002) from the Emission Area to PFCS, the workcell uses machine ID ET0E, and the actual tester ID (B1-B4) is sent as part of the test results.

2.4 Message Types

The following table describes the valid PFCS Message Types

Message Type	Description
0001	Request for vehicle data from host to PFD (i.e. get data from host). Requested data will also be returned from PFCS with this message type. No other message can be sent while waiting for requested vehicle data.
0002	Test Results to PFS from PFD
0004	Test Results to Broadcast from PFD
0006	Messages from ALSVS to PFCS
0003	Unsolicited vehicle data from host to PFD.
9999	Keep alive message from PFDs.

2.5 Error Recovery Processing

If either PFCS or a PFD detects an error, the error messages described in the table in *Section 2.5.1* are used to identify the error so that appropriate actions can be taken.

2.5.1 Error Description Table

The following error codes are generated by PFCS and sent to the PFD to indicate possible errors in ACK/NAK records.

Sent With	Error Code	Error Description	Action to be taken by PFD
NAK	A	Bad data received from PFD, message type not configured in PFCS, or data can not be processed by PFCS.	Retry message as configured (default 3 times) at 5 sec. intervals.
	B	Machine ID not valid.	PFD must use the first four bytes from NAK as machine ID.
	E	Sequence number not numeric.	PFD should reset its sequence number to "000001" and re-send
	H	Invalid message type.	Correct message type and re-send.
	J	Invalid byte count of data, must be in the range 0000-1024.	Correct data length and re-send.
	I	Input exceeds 1024 bytes	Reformat the message and re-send
ACK	D	Duplicate message sequence number.	Process like ACK. Go to the next message.

The following table describes the error codes that must be generated by the PFD when an error condition occurs.

Sent With	Error Code	Error Description	Action to be taken by PFCS
NAK	A	Bad data received from PFCS.	PFCS will retry sending the data. (default 2 times)
	E	Sequence number not numeric.	PFCS will reset the sequence number to "000001" and re-send the message.
	H	Invalid message type.	PFCS will log NAK for use in error detection and alarm notification
	I	Input exceeds 1024 bytes.	PFCS will log NAK for use in error detection and alarm notification.
	F	Unexpected sequence number	PFCS will re-send the message using the sequence number given in the PFD's NAK. (default 2 times)
	J	Invalid byte count, must be in the range 0000-1024.	PFCS will re-send the data. (default 2 times)
ACK	D	Duplicate message sequence number	PFCS will process like ACK and send the next message.
	G	Delayed data delivery	PFCS will process like ACK
	K	Invalid Vehicle Data	PFCS will process like ACK & log for use in error detection & alarm notification.

2.5.2 Duplicate messages

Duplicate messages are detected using sequence numbers (bytes 8-13). If a PFD receives a duplicate from PFCS, it must send an ACK with error code '**D**'. PFCS will do the same if it receives a duplicate message from a PFD. Upon receipt of an ACK-D, the PFD should discard the message and continue processing.

2.5.3 Unrecognizable Characters

If the PFD receives unrecognizable characters from PFCS (control characters, etc.) in any portion of the message, it should log an error and send a NAK-A message to PFCS.

2.5.4 Sequence Number Processing

This section discusses the use of the PFCS sequence number for error detection and the actions to be taken by PFCS or the PFD to resolve or log the event.

2.5.4.1 General Message Sequence Number Rules

General Message Sequence Number Rules for both the PFD and PFCS are as follows:

- A message sequence number will start at "000000" and will increment by one until the number "999999" is reached.
- When the message sequence number reaches "999999", the number will "wraparound" or reset to the number "000000" and continue as a normal sequence number from that point forward.
- The PFD must maintain a separate message sequence number for each port used in PFCS communications.

2.5.4.2 Messages from PFCS

Rules governing the use of sequence numbers for messages from PFCS to the PFD are as follows:

- A PFD must send a NAK with an error code of 'F' for any unexpected sequence number received from PFCS. A message sequence number is unexpected when the current sequence number is not equal to the previous sequence number plus one. PFCS will re-send the message with the expected sequence number taken from the NAK-F message
- If a PFD receives a type "0003" message with sequence number "000000", then the PFD's sequence number must be re-set to "000000".

2.5.4.3 Messages from PFD

Rules governing the use of sequence numbers for message from a PFD to PFCS are as follows:

- If PFCS receives an unexpected sequence number from the PFD:
    - PFCS will log the event into its transaction log.

    - If no other problem is found with the message, PFCS will send an ACK to the PFD using the received sequence number. .
- If the connection from the PFD to PFCS terminates or restarts, the sequence number will be reset to "000000".
- For a PFD that requests Vehicle Build Data through PFCS (Message Type "0001")
    - If a PFD determines that it has a delayed response for vehicle build data, it must send an ACK-G message to PFCS.
    - A delayed response is determined by the PFD either by message content or a request time-out cycle. See *Section 2.5.6 Time-out (Received ACK from PFCS and Waiting for Vehicle Data)*

2.5.5 Time-out (Waiting for ACK)

PFD equipment must time-out in *N* seconds (typically N=5 seconds) while waiting for an ACK/NAK from PFCS and retry the message up to 3 times. If all retries are unsuccessful, PFD should close the connection and reconnect. This is designed to handle lost messages between process equipment and PFCS.
Note: The length of the time-out should be determined by the process cycle time and must be a configurable option in the PFD.

2.5.6 Time-out (Received ACK from PFCS and Waiting for Vehicle Data)

PFDs must time-out in *N* seconds (typically N=5 seconds) while waiting for a response to a request. The request must be logged and discarded. If the process allows, a new request may be submitted to PFCS as long as the sequence number is incremented.
Note: The length of the time-out should be determined by the process cycle time — normally, the length of time before the PFD prompts the operator to manually perform the entire operation — and must be a configurable option in the PFD.

2.5.7 Machine-ID Error

The PFCS machine ID(s) must be a configurable option in the PFD. PFCS returns a NAK-B for any invalid Machine ID. The PFD must respond to this message in one of the following ways:

1) If the PFD's process only requires a single machine ID on its port, then the PFD can get the expected machine ID from a NAK-B by sending an invalid machine ID (i.e., dashes) in its first message.

2) If the PFD's process requires multiple machine ID's on its port, then the PFD can get the expected machine ID from a NAK-B by sending an invalid machine ID (i.e., dashes) in it's first message. This machine ID should be used as the tool's main machine ID (The main machine ID is the ID used for sending keep-alives and requesting vehicle data).

2.5.8 Keep Alive Message

All PFDs must send *Keep Alive Messages* at regular, two-minute intervals whenever they are in an idle state (not sending or receiving messages).

- An ACK is sent by PFCS in response to the *keep alive message* to let the PFD know PFCS is alive.
- If PFCS fails to respond to a *keep alive message*, then the PFD must go into its error recovery routine as defined in *Section 2.7.2.1 Error Recovery*
- This message is typically used to diagnose communication problems.

Format: MMMM LLLLLL99990020VENDOR MODEL v1.1.1**<CR>**
(1) (2) (3) (4) (5) (6) (7) (8) (9)

(1) Machine-ID (4 bytes)
(2) Blanks (3 bytes)
(3) Last sequence number processed (6 Bytes)
(4) Request type, constant 9999
(5) Data length 20 (4 bytes)
(6) Vendor Name (8 bytes)
(7) Hardware Model (6 bytes)
(8) Software Version (6 bytes)
(9) Message terminator carriage return (x'0d')

PFDs can send text messages using the *Keep Alive Message* structure. These messages are logged in PFCS in log files only and can be used for debugging purposes.
For example:

Format: BF01 000010999990026VENDOR MODEL v1.1.1Manual**<CR>**
(1) (2) (3) (4) (5) (6) (7) (8) (9) (10)

(1) Machine-ID (4 bytes)
(2) Blanks (3 bytes)
(3) Last sequence number processed (6 Bytes)
(4) Request type, constant 9999
(5) Data length 26 (4 bytes)
(6) Vendor Name (8 bytes)
(7) Hardware Model (6 bytes)
(8) Software Version (6 bytes)
(9) Text message (Up to 50 bytes)
(10) Message terminator carriage return (x'0d')

2.6 SPD Vendor Information

When a vehicle passes through a broadcast station at frame, paint, and trim information about the parts needed for that vehicle is sent to the Sequenced Parts Delivery (SPD) system. SPD identifies the sequence in which parts must be shipped in containers so that parts are received in correct sequence on the plant floor.

2.6.1 SPD Message Format

This table identifies the SPD record layout including record header

Start	End	Len	
1	4	4	Machine Id defined by DaimlerChrysler
5	7	3	ACK/NAK area: Spaces
8	13	6	Message Sequence Number
14	17	4	Request Type:0003
18	21	4	Data Byte Count:(Position 22 to the end, excluding <CR>)
22	25	4	Record Identification BCST - Pay-As-Built and supplier on-line BCCS - Corporate Suppliers BCSB - Pay-As-Built only batch BCSS = Supplier only batch
26	26	1	Model Year
27	27	1	Plant Code
28	33	6	VIN (last 6)
34	35	2	Pay-As-Built Status
36	37	2	Broadcast Status
38	44	7	Supplier Number
45	47	3	Operation Number
48	59	12	Vehicle Order Number
60	66	7	Numeric Broadcast Sequence Number
67	77	11	Broadcast Date and Time (YYDDDHHMMSS) DDD – Julian date
78	78	1	Recovery Flag (Y, N or E = Error)
79	174	96	Part Information - up to eight occurrences of the following can be included: part number (10) and usage count (02)
175	175	1	Message Terminator: x 0d

Note: All Data is in ASCII format.

2.6.2 SPD Hardware Requirements

For primary communications between DCX systems and the supplier's systems DCX will provide a 56kb digital frame relay circuit to the supplier's location and will terminate the circuit with a 56KB DSU and a multi-port Router. DCX will, in addition, provide two (2) dial-in modems. Modem 1 will be used for the DCX Network Control Center to dial in for diagnostics, configuration and to initiate manual dial backup if the Router is not configured for auto-dial backup (Auto-dial backup is the default configuration option). Modem 2 will be used by the Router to initiate dial backup; the router will maintain the dial connection using Modem 2 until the primary frame relay circuit is returned to service.

The supplier is responsible for providing standard 115VAC power receptacles for the DSU, the Router, Modem 1, Modem 2 and at least 1 spare receptacle in the event DCX needs to use diagnostic equipment brought on-site at the supplier location. These five (5) power receptacles are the minimum power requirements for circuit connectivity.

In addition, the supplier is required to provide two standard business dial up telephone lines for use by Modem 1 and Modem 2 above in the event that dial backup is required.

SPD is an asynchronous application and the supplier is responsible for providing a serial I/O port on their SPD system and a cable connecting to this port for receipt and acknowledgement of SPD data sent by DCX to the suppliers system. The software the suppliers use for this purpose must conform to the coding requirements as outlined in other sections of this document.

The connecting cable provided by the supplier must terminate on the DCX end with a DB25 Male connector. The connector on the supplier's end must be of a type compatible with the supplier's serial I/O port. The cable must be wired as a straight through cable as the Router port will present a DCE interface to the supplier's system.

The modems and the supplier's I/O serial port must be configured for: 8 data bits, no parity, 1 stop bit and the I/O serial port should be set to operate at 9600BPS. Flow control may be used on this connection. DCX supports Xon/Xoff flow control, if flow control is required.

Modem configuration

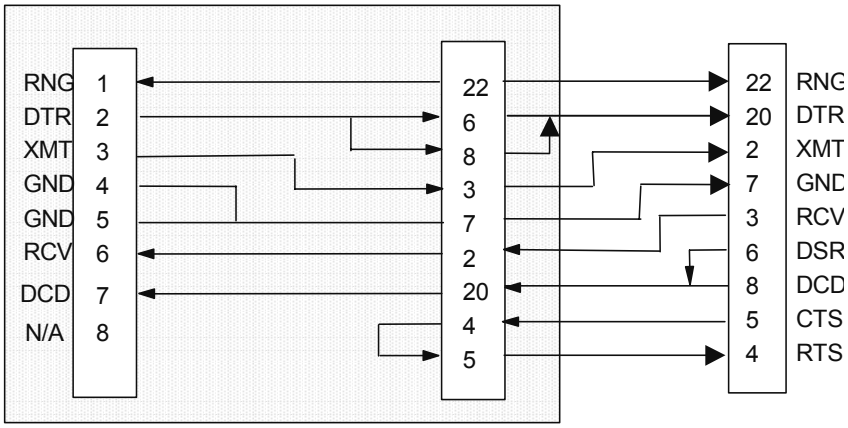
Data Format	8 bit ASCII
Baud Rate	9600 BPS
Parity	None
Stop Bits	1
Flow Control	Xon/Xoff (if required)

2.7 PFCS Connectivity Specifications

2.7.1 RS232 Device Connectivity

All vendor devices intending to connect to DaimlerChrysler computer equipment for transferring data electronically must fully conform to the industry standard for serial communications. This standard is defined by the EIA RS-232C serial interface specification and can be obtained from the EIA (Electronics Industries Association). The document is fairly large and contains protocols other than RS-232C. The RS-232C portion of the EIA standard can be broken down into two main parts. They are *synchronous* and *asynchronous* communications. This document (B2_0) will pertain only to the asynchronous communications portion of the EIA. Outlined here are the requirements for affecting a two-way asynchronous communications link with DaimlerChrysler machines. **Note**: This is in no way meant to displace or replace the EIA document as the ultimate source of information regarding RS-232C communications. **Note**: DaimlerChrysler will provide an RS-232 to RS-423 (RJ45) adapter to connect the CAT-5 cable drop to a PFD. This will enable extended cable length to a maximum of 500 feet.

Connector Wiring Diagram



The vendor must supply a serial port on their equipment for connection of the RS-232C cable to be used for data transfer. This port must present a 25-pin (DB25) **female** connector configured as DTE. **Note**: More than one port may be required on the vendor machine. In addition, DaimlerChrysler will supply the RS-232C connecting cable (s). The RS-232C communication specification for serial data transfer using DB25 signaling refers to a full-pinned RS-232C cable (a cable wired with 25 copper conducts). This document (B2_0) will concern itself only with those pins required for asynchronous transmission. The table in *Section 2.7.1.1* will describe these pins and their functions.

2.7.1.1 Female Connectors (Pin Assignments & Descriptions)

Name	PIN	Direction	Description
TXD	2	=>	Transmit Data. Data to be sent to DaimlerChrysler machine will be presented on this pin.
RXD	3	<=	Receive Data. Data received by the vendor machine from DaimlerChrysler will be presented on this pin.
RTS	4	=>	Request to send. This pin is raised high by the vendor machine at power up and should remain high as long as power is applied to the vendor machine.
CTS	5	<=	Clear to send. This pin will be raised high by the DaimlerChrysler machine when Pin 4 (RTS) is detected high presented by the vendor machine.
DSR	6	<=	Data set ready. This pin will be set high by the DaimlerChrysler machine
SG	7	<=>	Signal ground. Must be present to transmit or receive asynchronous signals.
DCD	8	<=	Data carrier detect. This pin will be set high by the DaimlerChrysler machine.
DTR	20	=>	Data terminal ready. This pin will be set high by the vendor machine and should remain high as long as the PFD is operational.

Note: Pin 4 (RTS) should be set to permanent high. Permanent high (or permanent 'request to send') is the typical setting in the asynchronous environment. RTS/CTS flow control is not supported.
Pins 6 and 8 (DSR and DCD) are indications that the DaimlerChrysler equipment is ready for data communication and can be used to determine if DaimlerChrysler equipment is online. Lack of signal on these two leads indicates DaimlerChrysler equipment is offline.
The voltage to be placed on Pin 4 and Pin 20 by the vendor machine should be in accordance with the EIA RS-232C serial interface specification. However, this voltage is typically +3 to +12 volts high and -3 to -12 volts low. +3 to -3 volts indicates 'off'.
All vendor devices that are to receive unsolicited data from DaimlerChrysler IS-PFCS, must conform to the following requirements:

Minimum Pin Requirements 2, 3, 7 & 20. Pin 20 must be high only when the PFD is operational. This pin is used by PFCS to determine the connection state of a PFD.

2.7.1.2 Error Recovery

Vendor programs must detect connection errors and be able to recover

- Typical errors
  - Peer disconnect
  - No response from PFCS for up to 3 retries. (Including keep-alive messages)
  - Unrecognizable data received on port

- Recovery process
  If the PFD detects a loss of communication to PFCS:
  - Close the port (bring DTR low)
  - Reset internal counters and buffers
  - Immediately re-establish a connection to PFCS (bring DTR high)
  - If the initial attempt to reconnect was unsuccessful, the PFD must continue to attempt to reconnect with each new transaction to PFCS (Including keep-alive messages)

2.7.2 TCP/IP Device Connectivity Specifications

All vendor devices using TCP/IP to communicate with PFCS must conform to the following requirements:

- Hardware Requirements

Vendors must provide the NIC (Network Interface Card) to connect to the plant network and support TCP/IP interface.

Cable Type	10BaseT or 10Base2 (dependant on plant infrastructure)
Network Interface Card	Vendor can get current part number and specifications from DaimlerChrysler Launch Coordinator. Launch coordinator may contact DaimlerChrysler Hardware Planning or DaimlerChrysler SPAS department for part number.

- Software Requirements

Vendor programs must perform a client role to establish a TCP/IP connection to PFCS. Plant ITM will provide the PFCS IP Address and Port Number for the PFD to establish a connection to PFCS. The lone exception to this rule is when the PFD is a PLC communicating via TCP/IP. PFCS will establish the connection to the PFD, in this case. There may be some differences between systems like MS WINDOWS and UNIX as far as TCP stack implementation.

The Corporate PFCS group can also provide a DLL (including source code, developed in Visual C++) to be used as an example for PFCS communication and is not responsible for any damage caused by using this code. This DLL is provided as shareware code and is not specifically supported by PFCS

2.7.2.1 Error Recovery

Vendor programs must detect connection errors and be able to recover.

- Sample errors
  - Peer disconnect.
  - Socket address in use.
  - No Response from PFCS for up to 3 retries. (Including keep-alive messages)
  - Unrecognizable data on port

- Recovery process
  If the PFD detects a loss of communication to PFCS:
  - Close the socket
  - Reset internal counters and buffers
  - Immediately re-establish a connection to PFCS
  - If the initial attempt to reconnect was unsuccessful, the PFD must continue to attempt to reconnect with each new transaction to PFCS (Including keep-alive messages)

2.7.3 PLC Connectivity - USD Driver

The APIC Group provides the Universal Serial Driver (USD) for PLC communications to PFCS. This driver is able to communicate to PFCS via either TCP/IP or RS232.

2.8 Receiving Vehicle Data & Sending Operation Results

2.8.1 Vehicle Data by Request

The data exchange for PFD's that make individual build data requests for every vehicle is described below. This must be clarified with the plant coordinators.

1. PFD is required to have 1 connection to PFCS.
2. The PFD must establish logical socket connections to PFCS or dedicate an RS232 port on the device to PFCS communication.
3. PFD must request a VIN or AVI Barcode by sending a "0001" request to PFCS. If the request times out (either waiting for ACK or waiting for vehicle build data), the PFD must inform the operator to override manually or mark the vehicle for re-work/repair.
4. After the operation is completed, the results should be sent to PFCS. If the communication to PFCS is down, the PFD must store the operation results for future transmission (if required by the process).

Automatic/Manual Mode Operation

The following issues are items the process equipment vendors must consider in PFD communication implementation. These issues are to be viewed as process issues, not communication issues.

Events to consider in Automatic Mode:

- Scanner no-read event followed by a manual VIN/AVI Barcode entry and style select by the operator. If test is performed, results must be sent.
- Scanner no-read event followed by a manual VIN/AVI Barcode entry and style select by the operator. The operator then aborts the test. The process equipment must send a fail.
- Scanner no-read and without manual selections of both the VIN/AVI Barcode and style, the process equipment must not allow the test to be performed.
- The process equipment must clear the VIN/AVI Barcode after the operation for that particular vehicle is completed.

Events to consider in Manual Mode:

- Manual VIN/AVI Barcode entry and style select by the operator. If test is performed, results must be sent.
- Manual VIN/AVI Barcode entry and style select followed by an abort by the operator. The process equipment must send a fail.
- Normal scan event followed by an abort by the operator. The process equipment must send a fail.
- Normal scan event followed by test being performed. Valid results must be sent.

2.8.2 Storing Operation Results

In the previous section, it is mentioned that if the connection to PFCS is down, the PFD must store the operation results for future transmission (if required by the process). Some general guidelines regarding the storing of these results are:

- If the PFD is sending results that DO NOT contain a vehicle identifier (VIN or AVI barcode label), the result data must not be stored when the connection to PFCS is down. In this case the PFD must ONLY send the latest data to PFCS

- If the PFD is sending results that DO contain a Vehicle Identifier (VIN or Barcode), a configurable option must be available to store the result data when the connection to PFCS is down. The tool must be able to store all result data for at least the last 30 minutes worth of production time.

3 Torque Monitoring

Section 3.0 includes information on torque monitoring performed by different types of single and multi-nutrunner tool controllers. Before identifying the features of each type of controller, it is important to understand key terms and the normal sequence of operations on the plant floor.

3.1 Sequence of Operations

For the purposes of this section, a typical station contains:

- at least one PFS (corporate quality system) terminal
- one or more torque tools associated with a PFS terminal in that station
- actual station setups may vary by plant and implementation

1. The plant floor operator verifies that the system is synchronized with the plant floor by locating the vehicle identity and matching it to the identity shown on the PFS screen.
2. The operator performs the required operation.
3. The operator determines that the results of the operation are reflected on the PFS Screen. (Pass / Fail)

3.2 Key Definitions

The following definitions are important to understanding plant floor equipment function.

- Vehicle Operation – An operation performed on a vehicle at a given station using one or more tools, i.e., lug nut operation; left and right side tools used to fasten front and rear lug nuts of vehicle.
- Rundown Message – Message containing data from one or more rundowns sent by the tool controller.
- Rundown Data – Data sent about each rundown (each bolt) from the tool to the tool controller. Part of a rundown message.
- Undo/Redo Rundown – An additional rundown that results from an error or failing initial rundown. Re-doing an operation causes additional rundown data for the same vehicle.
- Channel – A channel is a single connection between the tool and the tool controller.
- Machine ID – The machine ID is a four character identifier in the front of the rundown message sent to PFCS.
- Spindle Number – The spindle number is a two character number identifying each spindle in the rundown message.

3.3 Nutrunner Tooling

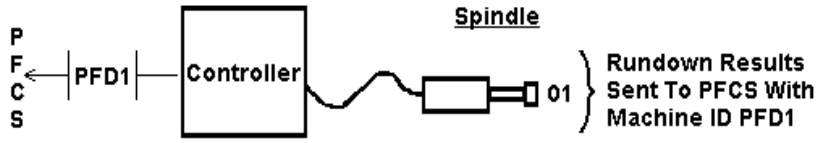
3.3.1 Single Spindle – Single Channel Controller

The single spindle - single channel controller has only one tool with a single spindle that sends rundown data through a single channel to the tool controller. The vehicle operation consists of a single or multi-bolt rundown using the single spindle tool. The controller sends a message to PFCS every time a rundown is performed. The spindle number in the rundown message is always '01'.

- **Single Machine ID**

With this setup the machine ID in the rundown message is the same for every rundown.

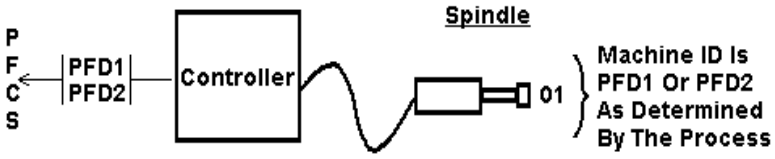
EXAMPLE
Single Spindle
Single Channel
Single Machine ID



- **Machine ID Determined By Process**

It is possible for more than one machine ID to be assigned to a single spindle tool. With the setup the process must have a method of determining which rundown the tool is currently performing. The controller then uses the machine ID appropriate to that fastener to label the rundown message.

EXAMPLE
Single Spindle
Single Channel
Machine ID Determined
By Process

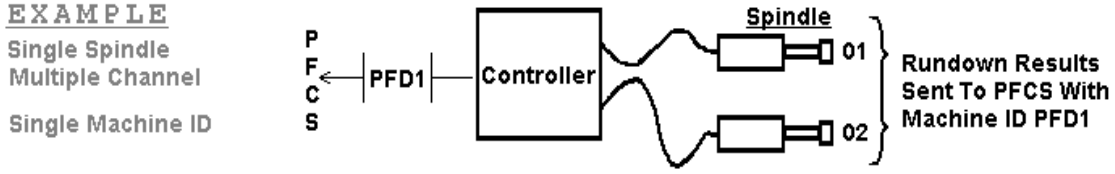


3.3.2 Single Spindle – Multiple Channel Controller

The Single Spindle – Multiple Channel Controller uses multiple tools with one spindle each.

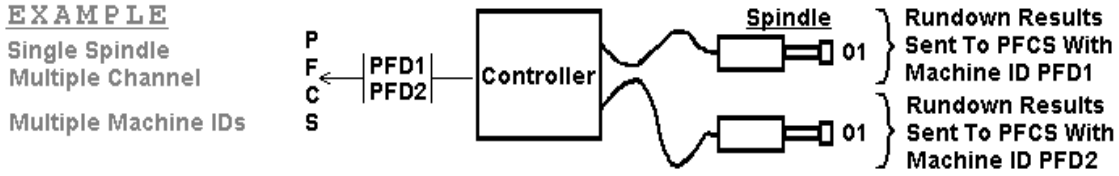
- **Single Machine ID**

A single machine ID setup has a single machine ID for more than one spindle. Each spindle has a unique spindle number starting with '01'. Rundown Results are sent upon completion of the rundown.



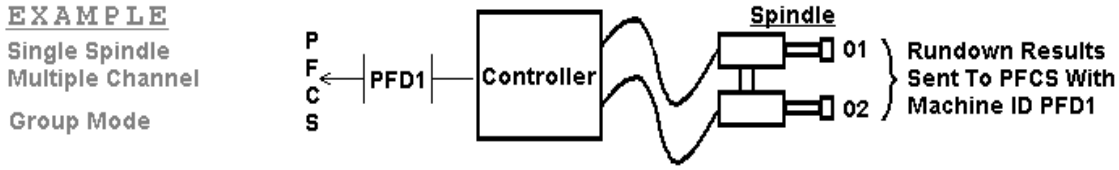
- **Multiple Machine IDs**

A multiple machine IDs setup has a machine ID for each spindle. The spindle number for each rundown message is always set to '01'. Rundown results are sent upon completion of a rundown.



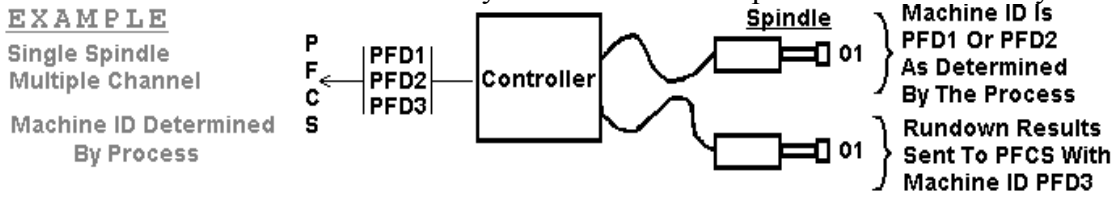
- **Group Mode**

With the group mode setup, the spindles are designed to run at the same time and are often physically joined by a fixed mount or bracket. When the rundowns are completed they are sent to PFCS together. The first spindle number is set at '01', the second spindle number set to '02' and so on.



- **Machine ID Determined By Process**

With this setup the controller must have a method of determining which fastener is being worked on. Upon completion of a rundown the controller sends the rundown message to PFCS with a machine ID determined by that method. The spindle number is always '01'.

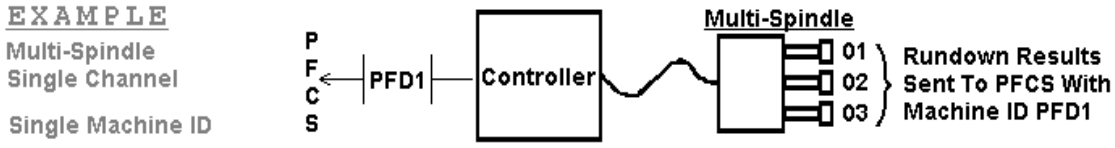


3.3.3 Multi-Spindle – Single Channel Controller

This controller has only one multi-spindle tool. These tools are customized for their operation and contain the exact number of spindles to rundown all bolts simultaneously. The rundown results for all of the spindles are sent to PFCS in a single rundown message. Each spindle is assigned a spindle number starting with '01' then increasing by one for each additional spindle.

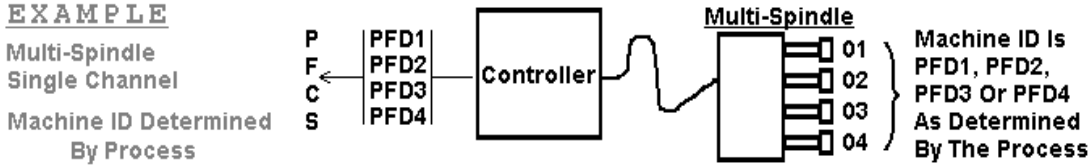
- **Single Machine ID**

This setup has the controller sending the rundown message with the same machine ID every time the multi-spindle tool completes a rundown.



- **Machine ID Determined By Process**

With this setup the machine ID for the rundown message is different depending on which fasteners the multi-spindle tool is currently working on. The controller must have a method of determining which fasteners it is about to perform the rundown on for this setup to work.

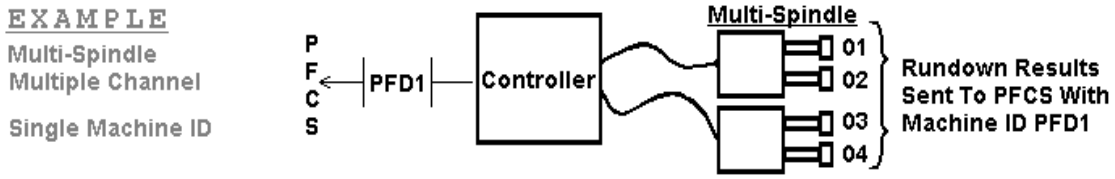


3.3.4 Multi-Spindle – Multiple Channel Controller

When one controller handles more than one multi-spindle tool, the controller is called a Multi-Spindle - Multiple Channel Controller. The rundown data for each spindle in a multi-spindle tool is sent in a single rundown message upon completion of a rundown by that tool.

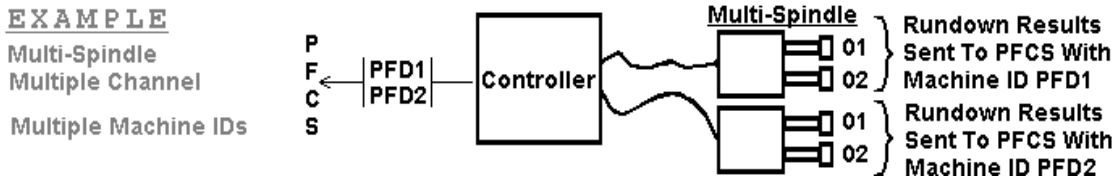
- **Single Machine ID**

A single machine ID setup has a single machine ID for more than one multi-spindle tool. Each spindle on the multi-spindle tools has a unique spindle number starting with '01'.



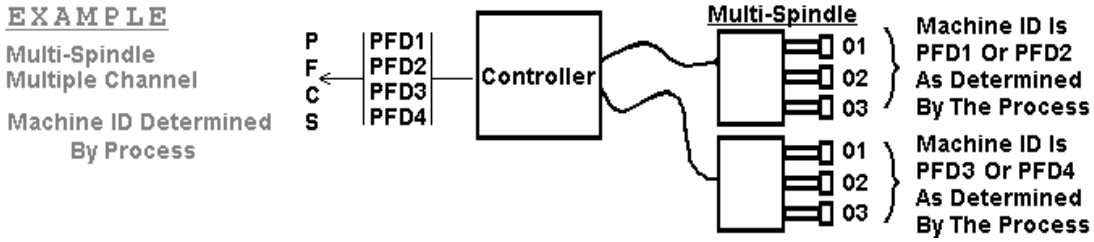
- **Multiple Machine IDs**

A multiple machine IDs setup has a separate machine ID for each multi-spindle tool. For each individual PFCS Machine ID the spindles are assigned spindle numbers starting with '01'.



- **Machine ID Determined By Process**

With this setup the controller must have a method of determining which fasteners it is currently working on. The controller then uses the machine ID appropriate to those fasteners to label the rundown message.



CONTROLLER TYPE	FEATURES	SEND RUNDOWN RESULTS WHEN	EXAMPLE (First six characters of each result)
Single Spindle Single Channel	Single Machine ID	Single Rundown Is Completed	PFD1 → P0101P..
	Machine ID Determined By Process	Single Rundown Is Completed	PFD1 → P0101P.. PFD2 → P0101P..
Single Spindle Multiple Channel	Single Machine ID	Single Rundown Is Completed	PFD1 → P0101P.. PFD1 → P0201P..
	Multiple Machine ID	Single Rundown Is Completed	PFD1 → P0101P.. PFD2 → P0101P..
	Group Mode	As Required by Process	PFD1 → P0101P..P0201P..
	Machine ID Determined By Process	Single Rundown Is Completed	PFD1 → P0101P.. PFD2 → P0101P..
Multiple Spindle Single Channel	Single Machine ID	As Required by Process	PFD1 → P0101P..P0201P..P0301P..
	Machine ID Determined By Process	As Required by Process	PFD1 → P0101P..P0201P..P0301P.. PFD2 → P0101P..P0201P..P0301P..
Multiple Spindle Multiple Channel	Single Machine ID	As Required by Process	PFD1 → P0101P..P0201P.. PFD1 → P0301P..P0401P..
	Multiple Machine ID	As Required by Process	PFD1 → P0101P..P0201P.. PFD2 → P0101P..P0201P..
	Machine ID Determined By Process	As Required by Process	PFD1 → P0101P..P0201P.. PFD2 → P0101P..P0201P..

4 Data Formats

Section 4.0 identifies the data format for the following: -

- Requesting Vehicle build data.
- Receiving Vehicle build data.
- Sending Operation result data.

4.1 Requesting Vehicle Build Data

The Vendor must support the following two formats for requesting build data:

4.1.1 Requesting Vehicle Build Data Using VIN

The following format is used by the PFD to *request* vehicle data from PFCS:

	Start	End	Len	Description
Header	1	4	4	Machine ID
	5	7	3	ACK/NAK area: Spaces
	8	13	6	Message Sequence Number
	14	17	4	Message Type: 0001
	18	21	4	Data Byte Count: 0018
Data Area	22	23	2	Type Code: Spaces=get specific vin, GN=get next
	24	29	6	Device Name from PES *
	30	31	2	Trigger Status: Spaces
	32	39	8	Last 8 of VIN
	40	40	1	Message Terminator: x 0d

Note: Process Equipment Specification (PES) is a sub-system of Broadcast

* Device name from PES is normally Machine ID plus 2 spaces

4.1.2 Requesting Vehicle Build Data Using AVI Barcode

(AVI- Automatic Vehicle Identification System)
The following format can be used by the PFD to *request* vehicle data from PFCS, if AVI barcode is available:

	Start	End	Len	Description
Header	1	4	4	Machine ID
	5	7	3	ACK/NAK area: Spaces
	8	13	6	Message Sequence Number
	14	17	4	Message Type: 0001
	18	21	4	Data Byte Count: 0018
Data Area	22	23	2	Type Code: GT
	24	29	6	Device Name from PES *
	30	31	2	Trigger Status : Spaces
	32	39	8	8 char VMS/AVI Barcode
	40	40	1	Message Terminator: x 0d

Note: Get Next (GN) requests are not supported for this option.

Note: Process Equipment Specification (PES) is a sub-system of Broadcast

* Device name from PES is normally Machine ID plus 2 spaces

4.2 Receiving Vehicle Build Data

4.2.1 Receiving Vehicle Build Data as Response to a Request

The following format is used by the PFD to *receive* vehicle build instructions from PFCS as response to a request:

	Start	End	Len	Description
Header	1	4	4	Machine ID
	5	7	3	ACK/NAK area: Spaces
	8	13	6	Message Sequence Number
	14	17	4	Message Type: 0001
	18	21	4	Data Byte Count (position 22 to the end, excluding Message Terminator)
Data Area	22	23	2	Type Code (Used by Broadcast / PFCS)
	24	31	8	Last 8 of VIN
	32	39	8	AVI Barcode
	40	41	2	Status: Vehicle Status, NF=not found
	42	48	7	Track Sequence Number
	49	+n		n= Specific variable data
	1+n		1	Message Terminator: x 0d

4.2.2 Receiving Vehicle Build Data at Broadcast Status Point

	Start	End	Len	Description
Header	1	4	4	Machine ID
	5	7	3	ACK/NAK area: Spaces
	8	13	6	Message Sequence Number
	14	17	4	Message Type: 0003
	18	21	4	Data Byte Count (position 22 to the end, excluding Message Terminator)
Data Area	22	23	2	Type Code (Used by Broadcast / PFCS)
	24	31	8	Last 8 of VIN
	32	39	8	AVI Barcode
	40	41	2	Status: Vehicle Status
	42	48	7	Track Sequence Number
	49	+n		n= Specific variable data
	1+n		1	Message Terminator: x 0d

4.3 PFD Results Record Formats

Section 4.3 provides *sample* record formats of results sent by the following plant floor devices:

4.3.1 Identifies the header used by all PFD's
4.3.2 Torque Monitor
4.3.3 EVAC/FILL (for Radiator, Power Steering, Brake, Washer devices)
4.3.4 Glass Installations
4.3.5 Door Tester
4.3.6 Rolls Test Equipment
4.3.7 Electrical Test

Note: Because device configurations vary by plant, it is important that the target plant verify record formats.

4.3.1 Record Headers

- **PFD Record Headers Using VIN/TRACK**

All results sent from PFDs must include the following header information:

Start	End	Len	Description
1	4	4	Machine ID defined by DaimlerChrysler, i.e., (XX01)
5	7	3	ACK/NAK area: Spaces
8	13	6	Message Sequence Number
14	17	4	Message Type normally: 0002 (Data send to PFS is 0002) (Data send to Broadcast is 0004)
18	21	4	Data Byte Count (position 22 to the end, excluding <CR>)
22	25	4	Number of Records in block: 0001
26	27	2	Filler: spaces
28	33	6	Last 6 of VIN
34	39	6	Last 6 of Track Seq #, if VIN is used insert zeroes
40	51	12	Date and Time (YYMMDDHHMMSS)
52	52	1	Overall Flag for accept or reject. P = Pass F = Fail
53	+n		Results format (Determined by Plant AME & PFS)

- **PFD Record Headers Using AVI Barcode**

All results sent from PFDs must include the following header information:

Start	End	Len	Description
1	4	4	Machine ID defined by DaimlerChrysler, i.e. (XX01)
5	7	3	ACK/NAK area: Spaces
8	13	6	Message Sequence Number
14	17	4	Message Type normally: 0002 (Data send to PFS is 0002) (Data send to Broadcast is 0004)
18	21	4	Data Byte Count (position 22 to the end, excluding <CR>)
22	25	4	Number of Records in block: 0001
26	27	2	Identifier: GT
28	35	8	AVI barcode
36	39	4	Filler: spaces
40	51	12	Date and Time (YYMMDDHHMMSS)
52	52	1	Overall Flag for accept or reject P = Pass F = Fail
53	+n		Results format (Determined by Plant AME & PFS)

4.3.2 Data Portion for Torque Monitors

Torque monitor equipment sends results in the following format:

	Start	End	Len	Description
Results of 1st rundown	53	53	1	Pass/Fail of first measurement
	54	55	2	Spindle Number *
	56	57	2	Bolt Count :01
	58	58	1	Torque Status (Pass or Fail)
	59	63	5	Torque High Limit (XXX.X)
	64	68	5	Torque Low Limit (XXX.X)
	69	73	5	Torque Reading (XXX.X)
	74	74	1	Angle Status (Pass or Fail)
	75	79	5	Angle High Limit (XXXXX)
	80	84	5	Angle Low Limit (XXXXX)
	85	89	5	Angle Reading (XXXXX)
	90	126	37	Results of the 2nd rundown
	127	163	37	Results of the 3rd rundown
	164	+n		Results of the nth rundown (max 25)
	n+1		1	<CR> (Message Terminator)

* Refer to *Section 3*

4.3.3 Example Data Portion for Evac and Fill

Evac and Fill are performed by a number of devices on the plant floor. The following table displays an example record format of results for Fill equipment.

Start	End	Len	Results
53	53	1	Final Vacuum (Pass or Fail)
54	57	4	Final Vacuum Value (0-9)
58	58	1	Overflow (Pass or Fail)
59	63	5	Vacuum Time (0-9)
64	64	1	Fill Pressure (Pass or Fail)
65	68	4	Fill Pressure Value (0-9)
69	69	1	Pressure Decay (Pass or Fail)
70	74	5	Pressure Decay Value (0-9)
75	75	1	Flowback (Pass or Fail)
76	80	5	Flowback Value (0-9)
81	81	1	Volume (Pass or Fail)
82	86	5	Volume Value (0-9)
87	91	5	Cycle Time (0-9)
92	92	1	<CR> (Message Terminator)

4.3.4 Example Data Portion for Glass Installations

The following record format is obtained from glass verification stations:

Start	End	Len	Results
53	53	1	Front glass match or mismatch (Pass or Fail)
54	54	1	Back glass match or mismatch (Pass or Fail)
55	55	1	Rear right glass match or mismatch (Pass or Fail)
56	56	1	Rear left glass match or mismatch (Pass or Fail)
57	57	1	Center right glass match or mismatch (Pass or Fail)
58	58	1	Center left glass match or mismatch (Pass or Fail)
59	59	1	Scanner read (Pass or Fail)
60	60	1	Vision read (Pass or Fail)
61	61	1	AVI communications (Pass or Fail)
62	63	2	Padding spare
64	64	1	<CR> (Message Terminator)

4.3.5 Example Data portion for Door Testers

The following record format is obtained from door test equipment:

Start	End	Len	Results
53	53	1	Body Model □AN□ PZ
54	54	1	BR-RKE P/Z
55	55	1	BR+RLE P/Z
56	56	1	Left door P/Z
57	57	1	Right door P/Z
58	58	1	Speakers P/F
59	59	1	Premium Sound P/F/Z
60	60	1	Power Mirrors P/F/Z
61	61	1	Power Locks P/F/Z
62	62	1	Power Windows P/F/Z
63	63	1	Security P/F/Z
64	64	1	Manual Entry P/F/Z
65	65	1	<CR> (Message Terminator)

4.3.6 Rolls Test Equipment

Because the record format for Rolls Test varies from plant to plant, PFS/Broadcast departments must be contacted to obtain the correct format.

4.3.7 Electrical Test

Because the record format for Electrical Test varies from plant to plant, the PFS/Broadcast departments must be contacted to obtain the correct format.

5 Vendor Certification

This section documents the PFCS Vendor Certification Process as conducted at the DaimlerChrysler Lab Facility with a representative from the PFCS Group and a representative from the Vendor's Organization.

The Lab Certification of the PFD Communications needs to be successfully concluded, before the PFD is allowed on the plant floor. The Vendor is to provide a working PFD Device for Certification and Testing that uses the same hardware and firmware configurations that will be used on the Plant Floor.

The PFCS Certification Process only certifies the PFD's ability to support the PFCS Communication Specifications. The vendor is responsible for any production issues that may arise due to the device's operation on the plant floor.

The PFCS Certification Process is required for all new PFD's. The PFD must be re-certified with PFCS anytime that hardware, software, or firmware changes are made.

5.1 Pre-Lab Documentation

The Pre-lab Documentation section of the PFCS Test Checklist (*Appendix B*) documents the physical aspects of the process that the PFD is intended to support. The purpose of this documentation requirement is to allow the PFCS Group go get a better understanding of the process and to address any system related setup issues prior to the PFD being installed at the plant.

This documentation must be completed prior to the Vendor receiving PFCS Certification.

The rest of this section will detail the instructions needed to fill out the PFCS Test Checklist.

5.1.1 Equipment Description

- Document the operation description as Process, Torque, or both. Process operations are distinguished by the sending of the VIN Number with the results.
- Document a short description of the Vendor hardware and method(s) used to communicate to the PFCS Server. Possible hardware configurations include PLC(USD), PC (possibly using the PFCS DLL), Custom controller, or Unix box. **NOTE:** Whenever possible, the PFD should talk directly from the tool to PFCS. Inserting a "Black Box" between the tool and PFCS is not a recommended method of PFCS communication.
- Document the type of connection(s) used by the equipment. The two possible connection types are Ethernet and RS232.
- Document if the equipment uses multiple machine IDs.
- Document the message types used by the equipment. Possible message types include 0001, 0002, 0003, 0004, 0006, and 9999.
- Document if the process requires that VIN numbers be sent with the result data.
- Give a brief description of the configuration procedures for the PFD.

5.1.2 Data Description

- Document a sample of the PFS result data.
- If expecting data from PFCS, document the vehicle parameters that are required by the equipment.

5.1.3 Data Recovery Procedures

- If the vehicle is being identified with the use of a scanner then document any back-up procedures in case of a scanner failure.
- If the vehicle identifications are being sent with the result data then document any back-up procedures for storing vehicle data in case of a communications failure.
- If the vendor receives vehicle identifications and stores them in a database, document any recovery procedures should the database become corrupted or lost..

5.2 Lab Certification Procedures

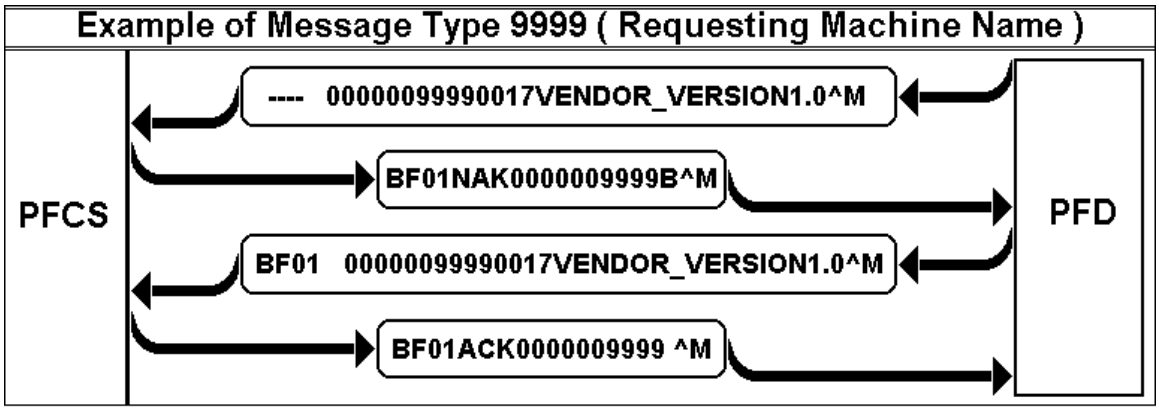
5.2.1 Startup

- Power up PFD.
- If equipment uses TCP/IP hardware then check that the IP Address of the PFCS Server, IP parameters of the PFD, and Socket Port Number(s) are all configurable. There may be multiple ports used in some PC applications, separate ports for result data and unsolicited vehicle data.
- Check all other configurable variables.
  - If the PFD is a multi-machine then verify the configurtation of the Machine ID(s) for each port socket.
  - Verify the configuration of the communication time out values.
  - Verify the configuration of the number of retries. (optional)
- Start up application.
- Look for connection status from PFD to PFCS.
- If applicable, check the vendor equipment's display(s) for the machine ID and/or connection status.
- Wait for first keep alive message. Make sure the first keep alive's message sequence number is 000000 and check for the version number of the vendor software in the data portion.

5.2.2 Requesting Machine ID (If Applicable)

- PFD sends a "keep alive" message with "----" in place of the machine ID.
- PFD receives a NAK "B" which contains the correct machine ID.
- Using the machine ID from the NAK "B" the PFD re-sends the "keep alive" message.
- PFD receives an ACK from PFCS.

NOTE:
If the PFD is a multi-machine then the machine ID returned by PFCS should be the "main" machine ID. (i.e. requests for vehicle data, "keep alive" messages)

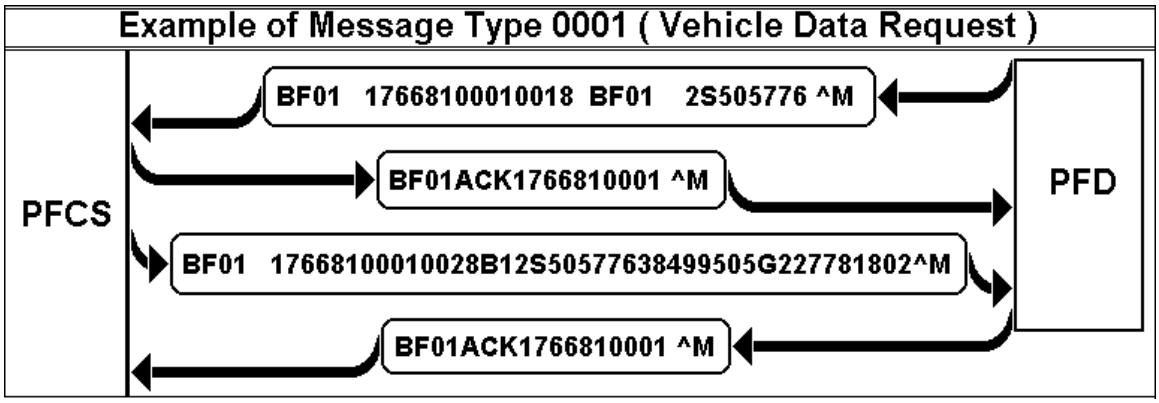


NOTE: ^M indicates carriage return character

5.2.3 Requesting Vehicle Data (Message Type 0001)

- PFD increments the message sequence number by one.
- PFD sends to PFCS a valid data request from vendor PFD using VIN and AVI barcode.
- PFD receives a valid ACK from PFCS.
- PFD receives valid vehicle data from PFCS.
- PFD sends a valid ACK to PFCS.

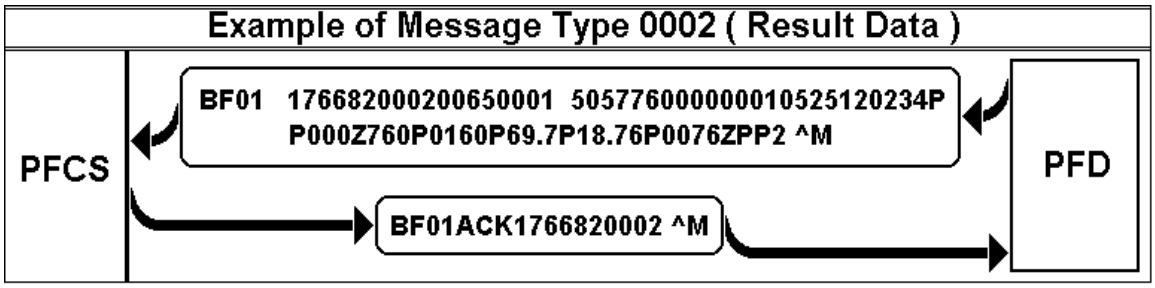
NOTES:
The message type is 0001 for all messages sent and received while requesting vehicle data
The message sequence number of the request will be used in both the PFCS ACK and vehicle data sent by PFCS. It must then be used in the PFD's ACK to receiving the vehicle data.
For multi-machines request must be on "main" machine ID only.
Make sure that the PFD matches the requested VIN/Barcode with the received VIN/Barcode



5.2.4 Test Result Data (Message Type 0002)

- PFD increments the message sequence number by one.
- PFD sends valid test result data to PFCS.
- PFD receives a valid ACK from PFCS.
- Repeat multiple times with a combination of pass and fail results.
- PFD should wait 2 minutes and send a "keep alive" message.
- PFD should receive an ACK from PFCS.

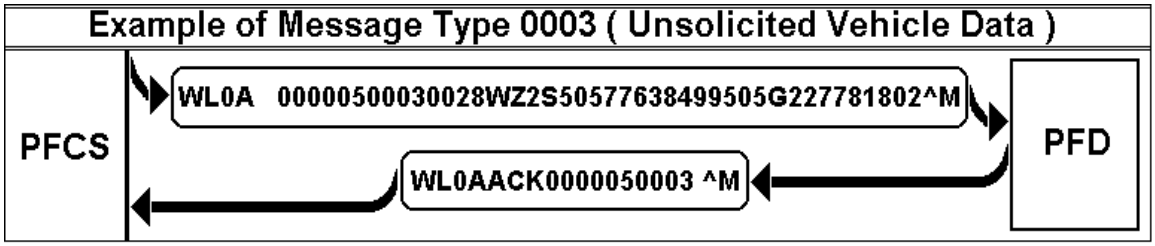
NOTES:
The message type is 0002 for PFS results and 0004 for broadcast results. (rolls specific)
The sequence number of the result message must be one higher than the sequence number of the message that preceded it. The PFD should maintain a separate message sequence number for each port.



5.2.5 Unsolicited Vehicle Data (Message Type 0003)

- PFD receives vehicle data from PFCS with the message sequence number one greater than the previous one.
- PFD increments the message sequence number by one to match that of the message sent by PFCS.
- PFD will validate the VIN data.
- PFD sends an ACK message to PFCS using the sequence number received with the data.
- On the test document, make a note of the type of database storage used (i.e. Flat, CSV, etc).
- PFD should wait 2 minutes and send a "keep alive" message using the same sequence number.
- PFD should receive an ACK from PFCS.
- Perform operation using local vehicle data and verify test results (refer to 4.2.4)

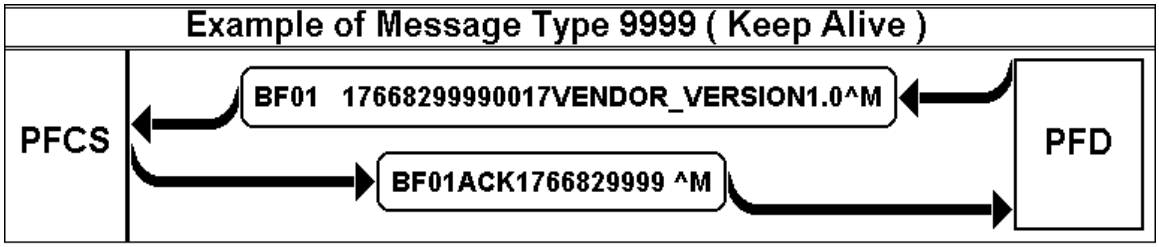
NOTES:
The message type is 0003 for all messages sent and received while receiving unsolicited vehicle data.
Verify that the unsolicited vehicle data was received by the PFD on a separate port socket then the operation results are sent.
Message Sequence Number for the unsolicited vehicle data is incremented by PFCS.
PFD should not make a request for vehicle data on the port that receives unsolicited vehicle data.
PFD increments the message sequence number only after receiving an ack.



5.2.6 Keep Alive Messages (Message Type 9999)

- PFD sends a "Keep Alive" message to PFCS.
- PFD receives a valid ACK from PFCS.
- Verify PFD waited two minutes after the last valid message or previous "keep alive" message before sending the "keep alive" message.
- Verify "Keep Alive" message contains identification and version number of PFD.

NOTES:
All PFCS PORTS shall use Keep Alive functionality.
The "keep alive" message's two minute timeout starts after last valid message or last "Keep Alive" message.
"Keep alive" messages do not increment the sequence number.



5.2.7 Error Recovery Procedures For Requests For Vehicle Data (Message Type 0001)

No Response to a request for vehicle data

- PFD increments the message sequence number by one.
- PFD sends a valid request for vehicle data.
- PFCS does not respond and the PFD should timeout.
- PFD should retry sending the request three times or as set in the configuration. (Timing out after each retry)
- If all retries were unsuccessful then the PFD should disconnect, reset the message sequence number to "000000", reconnect and perform the error procedure as defined by the process.

NOTE:
After failing to send the request to PFCS three times, the PFD should not attempt to send again after regaining communications.

No Vehicle data from PFCS after a request for vehicle data

- PFD increments the message sequence number by one.
- PFD sends a valid request for vehicle data.
- PFD receives a valid ACK from PFCS.
- PFCS does not send vehicle data and the PFD should timeout.
- PFD should indicate that manual operations are required.

NOTE:
If the process allows, a new request may be submitted to PFCS as long as the sequence number is incremented.

NAK-A

- PFD increments the message sequence number by one.
- PFD sends a valid request for vehicle data.
- PFD receives a NAK from PFCS with the error code "A".
- PFD should timeout as set by the configuration.
- PFD should retry sending the request three times or as set by the configuration. (Timing out after each retry)
- If all retries are unsuccessful then the PFD should indicate an error and enter manual operations.

NAK-B

- PFD increments the message sequence number by one.
- PFD sends a valid request for vehicle data.
- PFD receives a NAK from PFCS with the error code "B".
- PFD should retry sending the request three times or as set by the configuration.
- If all retries are unsuccessful then the PFD should indicate an error and enter manual operations.

NOTES:

A NAK "B" is sent if the machine name assigned to the PFD by PFCS does not match the machine name the PFD used in its message. Since this could be a symptom of a greater problem the PFD should not try to correct the machine name by using the machine name returned by the NAK.

ACK-D

- PFD increments the message sequence number by one.
- PFD sends a valid request for vehicle data.
- PFD receives an ACK from PFCS with the error code "D".
- PFD treats ACK-D as a normal ACK and does not repeat request.

NAK-E

- PFD increments the message sequence number by one.
- PFD sends a valid request for vehicle data.
- PFD receives a NAK from PFCS with the error code "E".
- PFD should set the sequence number to "000001" and retry sending the request three times or as set by the configuration.
- If all retries are unsuccessful then the PFD should indicate an error and enter manual operations.

NAK-H

- PFD increments the message sequence number by one.
- PFD sends a valid request for vehicle data.
- PFD receives a NAK from PFCS with the error code "H".
- PFD should correct the invalid message type and retry the sending the request three times or as set by the configuration.
- If all retries are unsuccessful then the PFD should indicate an error and enter manual operations.

NOTES:

A NAK "H" will only be sent by PFCS if the PFD sends a message with a machine name that is not defined by the communications document.

NAK-J

- PFD increments the message sequence number by one.
- PFD sends a valid request for vehicle data.
- PFD receives a NAK from PFCS with the error code "J".
- PFD should correct data length of message and retry sending the request three times or as set by the configuration.
- If all retries are unsuccessful then the PFD should indicate an error and enter manual operations.

NOTES:
NAK "J" is never actually sent by PFCS in the production environment, but is used by the test server program to catch invalid byte count errors sent by the PFD.

PFCS sends Vehicle Data Containing Unrecognizable Characters

- PFD increments the message sequence number by one.
- PFD sends a valid request for vehicle data.
- PFD receives a valid ACK from PFCS.
- PFD receives vehicle data from PFCS containing unrecognizable characters.
- PFD should send a NAK "A" to PFCS and discard the message.

NOTE:
If the process allows, a new request may be submitted to PFCS as long as the sequence number is incremented.

PFCS Sends Vehicle Data With Invalid Message Length

- PFD increments the message sequence number by one.
- PFD sends a valid request for vehicle data.
- PFD receives a valid ACK from PFCS.
- PFD receives vehicle data from PFCS with a message length greater than 1024 bytes.
- PFD should send a NAK "I" to PFCS and discard the message.
- PFD should indicate an error and enter manual operations.

NOTE:
If the process allows, a new request may be submitted to PFCS as long as the sequence number is incremented.

Delayed Response

- PFD increments the message sequence number by one.
- PFD sends a valid request for vehicle data.
- PFD receives a valid ACK from PFCS.
- PFCS does not send vehicle data and the PFD should timeout.
- PFD should indicate that manual operations are required.
- PFD receives valid vehicle data from PFCS
- PFD should send a valid ACK "G" to PFCS.
- PFD should discard the data message.

PFCS Sends Message With Duplicate Sequence Number

- PFD increments the message sequence number by one.
- PFD sends a valid request for vehicle data.
- PFD receives a valid ACK from PFCS.
- PFD receives valid vehicle data from PFCS.
- PFD should send a valid ACK to PFCS.
- PFD receives vehicle data from PFCS with the same message sequence number.
- PFD should send an ACK "D" to PFCS for the second message.
- PFD should discard the message with the duplicate message sequence number.

PFCS Sends Invalid Vehicle Data

- PFD increments the message sequence number by one.
- PFD sends a valid request for vehicle data.
- PFD receives a valid ACK from PFCS.
- PFD receives vehicle data from PFCS with invalid vehicle data.
- PFD should send a valid ACK "K" to PFCS.
- PFD should discard data with incorrect VIN number.
- PFD should perform error procedure as defined by the process.

5.2.8 Error Recovery Procedures For Sending Test Results To PFCS (Message Type 0002)

No Response from PFCS

- PFD increments the message sequence number by one.
- PFD sends result data to PFCS.
- PFCS does not respond and the PFD should timeout.
- PFD should retry sending the test results three times or as set in the configuration. (Timing out after each retry)
- If all retries are unsuccessful then the PFD should discard test results, disconnect, reset the message sequence number to "000000" and reconnect.

NOTE:
If the test results contain vehicle identification information and the PFD is designed to store test results then the PFD should re-send the stored result data when the connection is reestablished.

NAK-A

- PFD increments the message sequence number by one.
- PFD sends result data to PFCS.
- PFD receives a NAK from PFCS with the error code "A".
- PFD should timeout as set by the configuration.
- PFD should retry sending the test results three times or as set by the configuration.
- If all retries are unsuccessful then the PFD should discard test results.

ACK-D

- PFD increments the message sequence number by one.
- PFD sends result data to PFCS.
- PFD receives an ACK from PFCS with the error code "D".
- PFD treats ACK-D as a normal ACK and does not repeat request.

NAK-E

- PFD increments the message sequence number by one.
- PFD sends result data to PFCS.
- PFD receives a NAK from PFCS with the error code “E”.
- PFD should set the sequence number to “000001” and retry sending the test results three times or as set by the configuration.
- If all retries are unsuccessful then the PFD should discard test results.

NAK-I

- PFD increments the message sequence number by one.
- PFD sends result data to PFCS.
- PFD receives a NAK from PFCS with the error code “I”.
- PFD should check length of test results and retry sending the test results three times or as set by the configuration.
- If all retries are unsuccessful then the PFD should discard test results.

NAK-J

- PFD increments the message sequence number by one.
- PFD sends result data to PFCS.
- PFD receives a NAK from PFCS with the error code “J”.
- PFD should verify the overall byte count equals the total number of bytes and retry sending the test results three times or as set by the configuration.
- If all retries are unsuccessful then the PFD should discard test results.

5.2.9 Error Recovery Procedures For Receiving Unsolicited Data (Message Type 0003)

NOTES:
Make sure that the unsolicited data is received on a separate port than the one that receives solicited data. A separate message sequence number should be maintained for this port.

PFCS will normally try to send the unsolicited data to the PFD and wait for an ACK three times. If the data never arrives the PFD may need to enter manual operations as required by the process.

If an unsolicited data message is set for "guaranteed delivery" then PFCS will retry sending the data indefinitely.

PFCS will not increment the sequence number until it receives an ACK from the PFD.

PFCS sends Vehicle Data Containing Unrecognizable Characters
- PFD receives vehicle data from PFCS unrecognizable characters.
- PFD should send a NAK "A" to PFCS using the sequence number in the message from PFCS.
- PFD should discard the message.

PFCS Sends Vehicle Data With Invalid Message Length
- PFD receives vehicle data message from PFCS with a length greater than 1024 bytes.
- PFD should send a NAK "I" to PFCS using the sequence number in the message from PFCS.
- PFD should discard the message.

Wrong Build Information
- PFD receives vehicle data from PFCS with build information not matching PFD limits.
- PFD should send a valid ACK "K" to PFCS with the same sequence number received with the data.
- PFD should discard incorrect data.
- PFD should perform error procedure as defined by the process.

Duplicate Sequence Number
- PFD receives vehicle data from PFCS with the message sequence number incremented by one.
- PFD sends a valid ACK to PFCS with the same message sequence number received with the data.
- PFD stores data in local database.
- PFD receives valid vehicle data from PFCS with the same sequence number.
- PFD should send an ACK "D" to PFCS.
- PFD should discard message with duplicate sequence number.

Sequence Number Mismatch
- PFD receives vehicle data from PFCS with a sequence number out of sequence with the PFD.
- If message sequence number is "000000" then reset PFD's sequence number to "000000".
- Otherwise PFD should send a NAK "F" to PFCS using the expected sequence number, discard the incorrect data, and wait for the next message.

Invalid Message Type
- PFD receives vehicle data from PFCS with an invalid message type.
- PFD should send a NAK "H" to PFCS using the sequence number in the message from PFCS.
- PFD should discard incorrect data.

5.2.10 Error Recovery Procedures For Keep Alive Messages (Message Type 9999)

PFD Receives No ACK From PFCS For "Keep Alive" Messages.

- PFD sends "keep alive" message to PFCS.
- PFCS does not respond and the PFD should timeout.
- PFD should retry sending the "keep alive" message three times or as set in the configuration.
- If all retries are unsuccessful then the PFD should disconnect, reset message sequence number to "000000", indicate an error, and enter manual operations.
- PFD continuously tries to reconnect on each valid message to PFCS until the PFD receives a valid message back from PFCS. (ACK / NAK)

NAK-B

- PFD sends "keep alive" message to PFCS.
- PFD receives a NAK from PFCS with the error code "B".
- PFD should retry sending the "keep alive" message three times or as set by the configuration.
- If all retries are unsuccessful then indicate an error and enter manual operations.

5.2.11 Miscellaneous Error Recovery Procedures

PFCS Closes Connection To PFD

- PFD should reset sequence number to "000000".
- PFD continuously tries to reconnect on each valid message to PFCS until the PFD receives a valid message back from PFCS. (ACK / NAK)
- Check to see that the machine name is still correct.

NOTE:
If the PFD is designed to store test results then when a communications failure occurs the PFD should store the results and re-send the result data when the communication is reestablished.

PFCS Sends Unexpected Message

- PFD receives unknown message from PFCS.
- PFD discards message.

5.2.12 Shutdown

- PFD must close all connections to PFCS.
- If the PFD uses a RS232 connection check that the DTR pin went low.
- Check for peer disconnect in PFCS.

5.3 Integrated Testing

- Thoroughly check communications between PFD and PFCS.
- Identify PFD limitations.
- Perform parallel testing on more than one socket port with vehicle data requests and test results on one port and unsolicited vehicle data on another port.
- Ensure that PFD cannot send a message to PFCS while it is still processing an older message.
- Validate PFD logging and debugging.
- Check any user interface and setup screens.
- Connect PFD to UNIX to ensure PFD can communicate.
- Identify any process issues.
- If the PFD expects unsolicited vehicle data, validate procedures followed when data is not locally available.
- Unplug cable and check for peer disconnect in PFCS.

5.4 PFD Documentation

- PFD should have documentation of all setup procedures.
- PFD should have documentation of all error recovery procedures.

Appendix A: Available Testing Tools

The following is a list of server side testing tools available to the vendor in order to debug their PFCS code at their facility. Final testing must be done with the PFCS group, and the PFD system must be certified with PFCS before the software is installed in the production environment. The vendor must address any (communication) issues that may surface during production.

Vendor device: PC, communicates to PFCS is TCP/IP

Required Software:

- PFCS Test Sever software (available from PFCS, No warranty), runs on MS Windows

Required Hardware:

- RJ45 cable (Vendor device software runs on one system and PFCS test software runs on the other)

Vendor device: (Custom Hardware/PLC) communicates to PFCS using RS232 Port (Option 1)

Required Software:

- PFCS Test Sever software (available from PFCS, No warranty) runs on MS Windows
- Java Serial gateway application (available from PFCS, No warranty) runs on MS Windows

Required Hardware:

- DB9PIN female straight through serial RJ45 adapter (NCP-31089-9F, Netcom Phone: 1800 752 5855) attach to PC
- DB9PIN female cross over serial RJ45 adapter (NCP-31074-9F, Netcom Phone: 1 800 752 5855) attach to vendor device
- Straight thru RJ45 cable, connect it to the above two adapters.

(Vendor device software runs on one system and PFCS test software runs on the other)

(or)

Required Hardware:

- DB9PIN female straight through serial RJ45 adapter (NCP-31089-9F, Netcom Phone: 1800 752 5855) attach to PC
- DB25PIN male cross over serial RJ45 adapter (NCP-31074, Netcome Phone: 1 800 752 5855) attach to the vendor device
- Straight through RJ45 cable, connect it to the above two adapters.

(Vendor device software runs on one system and PFCS test software runs on the other)

Vendor device: (Custom Hardware/PLC) communicates to PFCS using RS232 Port (Option 2)

Required Software:

- PFCS Test Sever software (available from PFCS, No warranty), runs on MS Windows

Required Hardware:

- Lantronix single port terminal server (Model No. UDS-10-01, http://www.lantronix.com/products/ds/uds10) (DCE device)
  Note: PFCS will provide configuration instructions for the terminal server.
- Straight through RJ45 cable (to connect Lantronix single port terminal server to the network/Hub)
- DB25PIN male cross over serial RJ45 adapter (NCP-31074, Netcom Phone: 1 800 752 5855) attach to single port terminal server.
- DB25PIN male cross over serial RJ45 adapter (NCP-31074, Netcom Phone: 1 800 752 5855) attach to device
- Straight through RJ45 cable connect

(or)

Required Hardware:

- Lantronix single port terminal server (Model No. UDS-10-01, http://www.lantronix.com/products/ds/uds10) (DCE device)
  Note: PFCS will provide configuration instructions for the terminal server.
- Cross over RJ45 cable (to connect Lantronix single port terminal directly to PC running PFCS test server)
- DB25PIN female cross over serial RJ45 adapter (NCP-31074F, Netcom Phone: 1 800 752 5855) attach to single port terminal server.
- DB25PIN female cross over serial RJ45 adapter (NCP-31074F, Netcom Phone: 1 800 752 5855)
- Straight through RJ45 cable connect

Vendor device: PLC uses USD, communicates to PFCS using TCP/IP

Required Software:

- PFCS Test Sever software (available from PFCS, No warranty) runs on MS Windows
- RS Linx (Licensed software, Vendors responsibility) runs on MS Windows
- RS Linx Gateway (Licensed software, Vendors responsibility) runs on MS Windows
- PLC gateway software (available from APIC/Rockwell) runs on MS Windows

Required Hardware:

- Straight thru RJ45 cable (Vendor device software runs on one system and PFCS test software runs on the other)
- Transreceiver

Appendix B: PFCS Test Check List

Program/Project: _____ **Plant:** _____ **Area:** _____

System/Device Description: _____

OEM: _____ **Test Date:** _____

Equipment Description

Operation Description: Process: ☐ Torque: ☐ Both: ☐

PFD Type: PLC: ☐ PC: ☐ Custom Controller: ☐ Unix: ☐ Black Box: ☐

Operating System: _____ **Application Software:** _____

Communication: RS232: ☐ Ethernet: ☐ **Hardware Model Number:** _____
Software Version Number: _____

Number of Sockets: _____ Multi-Machine (Y/N): ___ Number of Machine ID's (Multi-Mach): ___

Message Types Used:0001: ☐ 0002: ☐ 0003: ☐ 0004: ☐ 0005: ☐ 0006: ☐ 9999: ☐

VIN Numbers sent with result data (Y/N): ☐

Store VIN Results during a Loss of Communication (Y/N): ☐

Configuration Procedures: _____

Data Description

Sample of Result Data from PFD: _____

Sample of Data from PFCS (if applicable): _____

Data Recovery Procedures

Backup Procedure for Scanner Failure (if applicable):_____

Backup Procedures for Storing Vehicle Data during a Communications Failure (if applicable):_____

Recovery Procedures Should the Database Become Corrupted (if applicable):_____

Startup

P	F	N/A		
☐	☐	☐	Verify that the IP Address of the PFCS Server, Socket Port Numbers, and IP Parameters of the PFD are all configurable:	_____
☐	☐	☐	Verify all Machine IDs used by the PFD are configurable:	_____
☐	☐	☐	Verify all communication timeout values are configurable:	_____
☐	☐	☐	Verify number of retries is configurable (optional):	_____
☐	☐	☐	Verify connection status from PFD to PFCS:	_____
☐	☐	☐	Wait for first Keep Alive Message:	_____

Comments:

Requesting Machine ID (if applicable)

P	F	N/A		
☐	☐	☐	PFD Sends a "Keep Alive" Message to PFCS with "----" in place of the Machine ID:	_____
☐	☐	☐	Using the Machine ID received in the NAK Response the PFD Re-sends the "Keep Alive" Message to PFCS:	_____

Requesting Vehicle data (Message Type 0001)

P	F	N/A		
☐	☐	☐	PFD Sends A Request for Vehicle Data to PFCS:	_____
☐	☐	☐	PFD Receives a Valid ACK from PFCS:	_____
☐	☐	☐	PFD Receives Vehicle Data from PFCS:	_____
☐	☐	☐	PFD Send a Valid ACK to PFCS:	_____
☐	☐	☐	PFD Sends Keep Alive Message 2 Minutes after Receiving Data:	_____

Test Result Data (Message Type 0002)

P	F	N/A		
☐	☐	☐	PFD Sends Test Result Data to PFCS:	_____
☐	☐	☐	PFD Receives a Valid ACK from PFCS:	_____
☐	☐	☐	PFD Sends Keep Alive Message 2 Minutes after Sending Results:	_____
☐	☐	☐	Repeat Multiple Times with Combination of Pass and Fail Results:	_____

Unsolicited Vehicle Data (Message Type 0003)

P F N/A
☐ ☐ ☐ PFD Receives Unsolicited Vehicle Data from PFCS: _____

P F N/A
☐ ☐ ☐ PFD Send a Valid ACK to PFCS: _____

Type of Data Storage Used To Hold Vehicle Data: _____

P F N/A
☐ ☐ ☐ PFD Sends Keep Alive Message 2 Minutes after Receiving Data: _____

P F N/A
☐ ☐ ☐ PFD Performs Operation Using Local Vehicle Data: _____

Keep Alive Messages (Message Type 9999)

P F N/A
☐ ☐ ☐ PFD Sends Keep Alive Message: _____

P F N/A
☐ ☐ ☐ PFD Waited 2 Minutes after last valid message: _____

P F N/A
☐ ☐ ☐ Keep Alive Message Contains Identification and Version Number: _____

Any Other Comments Sent Along with the "Keep Alive" Message: _____

Error Recovery for Requests for Vehicle Data (Message Type 0001)

P ☐ F ☐ N/A ☐ No Response to a Request for Vehicle Data: _____

P ☐ F ☐ N/A ☐ No Vehicle Data from PFCS after a Request for Vehicle Data: _____

P ☐ F ☐ N/A ☐ Received a NAK-A for a Vehicle Data Request: _____

P ☐ F ☐ N/A ☐ Received a NAK-B for a Vehicle Data Request: _____

P ☐ F ☐ N/A ☐ Received an ACK-D for a Vehicle Data Request: _____

P ☐ F ☐ N/A ☐ Received an NAK-E for a Vehicle Data Request: _____

P ☐ F ☐ N/A ☐ Received an NAK-H for a Vehicle Data Request: _____

P ☐ F ☐ N/A ☐ Received an NAK-J for a Vehicle Data Request: _____

P ☐ F ☐ N/A ☐ PFCS sends Vehicle Data Containing Unrecognizable Characters: _____

P ☐ F ☐ N/A ☐ PFCS sends Vehicle Data with Invalid Message Length: _____

P ☐ F ☐ N/A ☐ Delayed Response: _____

P ☐ F ☐ N/A ☐ PFCS Sends Message With Duplicate Sequence Number: _____

P ☐ F ☐ N/A ☐ PFCS Sends Invalid Vehicle Data: _____

Comments:

Error Recovery for Sending Test Results to PFCS (Message Type 0002)

P	F	N/A		
☐	☐	☐	No Response from PFCS:	_____
☐	☐	☐	Received a NAK-A in Response to Test Results:	_____
☐	☐	☐	Received an ACK-D in Response to Test Results:	_____
☐	☐	☐	Received an NAK-E in Response to Test Results:	_____
☐	☐	☐	Received an NAK-I in Response to Test Results:	_____
☐	☐	☐	Received an NAK-J in Response to Test Results:	_____

Comments:

Error Recovery for Receiving Unsolicited Data (Message Type 0003)

P ☐ F ☐ N/A ☐ PFCS sends Vehicle Data Containing Unrecognizable Characters: _____

P ☐ F ☐ N/A ☐ PFCS sends Vehicle Data with Invalid Message Length: _____

P ☐ F ☐ N/A ☐ PFCS Sends Message with Wrong Build Information: _____

P ☐ F ☐ N/A ☐ PFCS Sends Message With Duplicate Sequence Number: _____

P ☐ F ☐ N/A ☐ Sequence Number Mismatch with PFCS Message: _____

P ☐ F ☐ N/A ☐ PFCS Sends Message with Invalid Message Type: _____

Is Loss of Data Acceptable (Y/N): ☐

If not Acceptable Identify Manual Procedures: _____

Comments:

Error Recovery for Keep Alive Messages (Message Type 9999)

P ☐ F ☐ N/A ☐ PFD receives No ACK from PFCS for "Keep Alive" Messages: _____

P ☐ F ☐ N/A ☐ Received a NAK-B: _____

Comments:

Miscellaneous Error Recovery Procedures

P F N/A
☐ ☐ ☐ PFCS Closes Connection to PFD: _____

P F N/A
☐ ☐ ☐ PFCS Sends an Unexpected Message: _____

Comments:

Shutdown

P F N/A
☐ ☐ ☐ Perform PFD Shutdown Closing All Connections to PFCS: _____

P F N/A
☐ ☐ ☐ If using an RS232 Connection Check That DTR Pin is Low: _____

P F N/A
☐ ☐ ☐ Check For Peer Disconnect in PFCS: _____

Comments:

Integrated Testing

P F N/A ☐ ☐ ☐ Thoroughly Check Communications between PFD and PFCS: _____

Identify PFD limitations:_____

P F N/A ☐ ☐ ☐ Perform Parallel Testing with Test Result Data on One Port and Unsolicited Vehicle Data on the Other Port: _____

P F N/A ☐ ☐ ☐ Ensure that the PFD Cannot Send a Message to PFCS while Still Processing an Older One: _____

P F N/A ☐ ☐ ☐ Validate PFD Logging and Debugging Exists: _____

P F N/A ☐ ☐ ☐ Check Any User Interface and Setup Screens on the PFD: _____

P F N/A ☐ ☐ ☐ Connect the PFD to UNIX to ensure the PFD can Communicate: _____

Identify any process issues:_____

P F N/A ☐ ☐ ☐ Ensure the PFD can not send test result data requiring VIN data when the local database is empty: _____

If the PFD expects Unsolicited Vehicle Data what Procedures Should be Followed when the Data is not Locally available?_____

P F N/A ☐ ☐ ☐ Unplug cable and check for peer disconnect in PFCS: _____

Comments:

PFD Documentation

P F N/A

☐ ☐ ☐ PFD Has Documentation on All Setup Procedures: _____

P F N/A

☐ ☐ ☐ PFD Has Documentation on All Error Recovery Procedures: _____

Comments:

_____
_____
_____
_____
_____
_____
_____
_____
_____
_____
_____

This Checklist handles as much testing as is possible in the test environment. When the vendor installs the PFD in the plant it is the vendor's responsibility to address any production issues.

ITM:_____

NAME SIGNATURE DATE

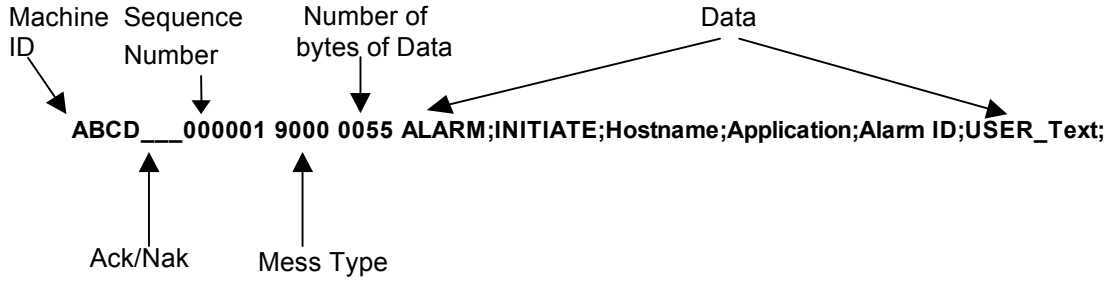
OEM:_____

NAME SIGNATURE DATE

Appendix C: Pager System

1 Message Format

Message Format to start paging: -



Machine ID	**ABCD**	This machine name is supposed to match the machine name configured at FirstPAGE PFCS Server.
Ack/Nak	**_ _ _**	(3 blanks)
Sequence Number	**000000**	Message sequence number, incremented with each new message
Message Type	**9000**	(Pager message type)
Number of bytes of data	**0080**	(data length) the length could be up to 1024 bytes
Data	**ALARM;INITIATE;Hostname;Application;Alarm ID;USER_text;**	

Where:

Hostname is the name of the System (Can be length of 1 to 40 characters, all capital letters with no spaces) i.e. NDC

Application is the name of application (Can be length of 1 to 40 characters, all apital letters with no spaces) i.e. LGV

Alarm ID is the problem ID. The field can hold up to 40 characters and numbers. It is a combination of the Area field (Max of 10 numbers and/or characters, all capital letters with no spaces)+ Machine field (Max of 10 numbers and/or characters, all capital letters with no spaces) + Message Code field (Max of 20 numbers and/or characters, all capital letters with no spaces). Example (BIW.CON4002.AB211) where BIW is the Area, CON4002 is the Machine under that Area, and AB211 is the Message Code under that machine.

Alarm ID = Area + Machine + Message Code

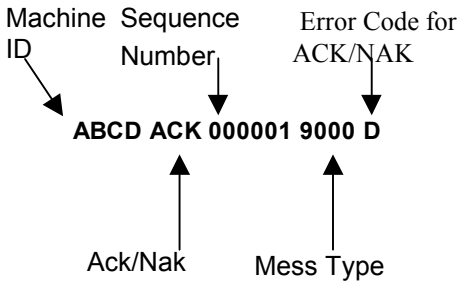
Area is the area of the plant that the device is located in. Each plant can have different areas, i.e. Assembly plants will have different areas than Component plants.
Machine is the name of the machine i.e. LGV
Message Code for example “Battery Low01”

USER_text in this field you can send alarm description (to describe the problem) and location and any other real time data. T This (I.e. VIN Number). This field is a required field to describe the problem.

End of message	**<CR>**	(CR will determine the end of the message)

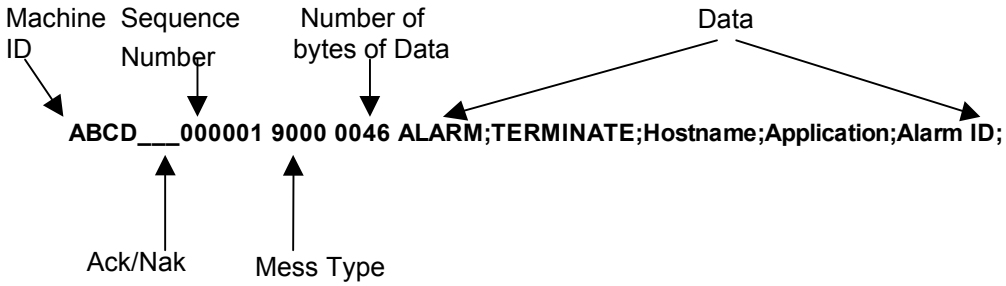
Then your application will get one of the following messages (ACK or NAK)

ACK and NAK messages are sent in the following format



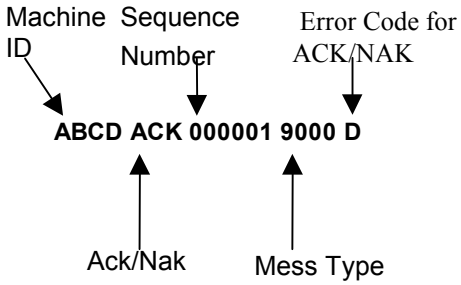
Message format to stop paging: -

Used if your application needs to send Alarm Terminate messages to stop the pager system from escalating

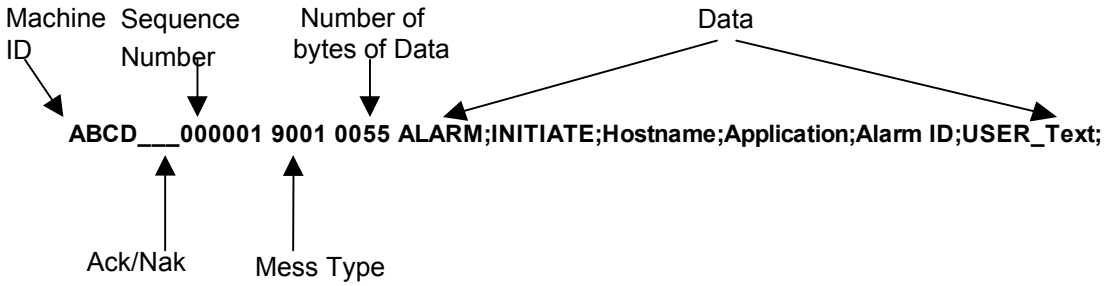


Then your application will get the following message (ACK or NAK)

ACK and NAK messages are sent in the following format

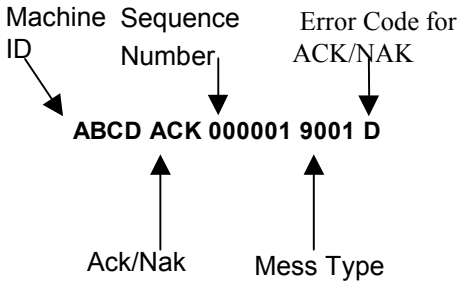


Message format for assured Page delivery: -

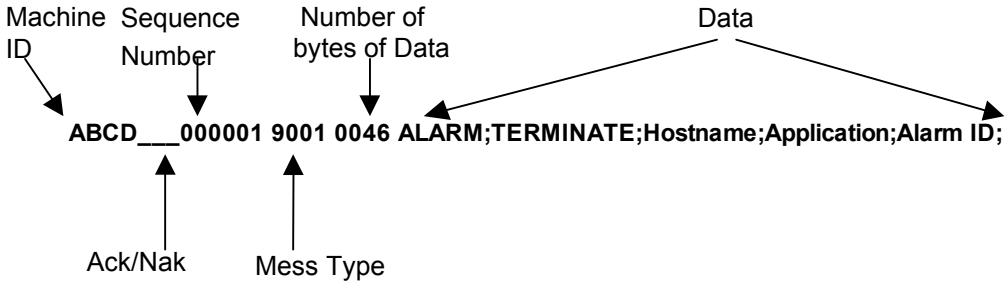


Then your application will get the following message (ACK or NAK)

ACK and NAK messages are sent in the following format

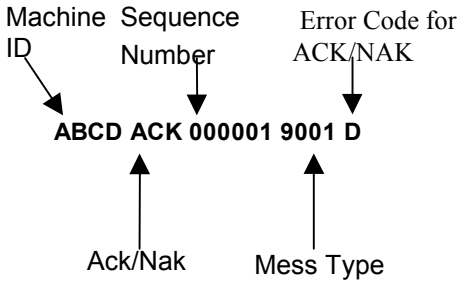


If your application needs to send Alarm Terminate messages to stop the pager system from escalating



Then your application will get the following message (ACK or NAK)

ACK and NAK messages are sent in the following format



2 Pager System Test Check List

Program/Project: _____ **Plant:** _____ **Area:** _____

System/Device Description: _____

OEM: _____ **Test Date:** _____

Equipment Description

PFD Type: PLC: ☐ PC: ☐ Custom Controller: ☐ Unix: ☐ Black Box: ☐

Operating System: _____ **Application Software:** _____

Communication: RS232: ☐ Ethernet: ☐ **Hardware Model Number:** _____
Software Version Number: _____

Number of Sockets: _____ Multi-Machine (Y/N): ___ Number of Machine ID's (Multi-Mach): ___

Paging Message Types Used: 9000: ☐ 9001: ☐ 9999: ☐

Configuration Procedures: _____

Data Description

Sample of Alarm Data from PFD: _____

Startup

P F N/A

☐ ☐ ☐ Verify that the IP Address of the PFCS Server, Socket Port Numbers, and IP Parameters of the PFD are all configurable: _____

P F N/A

☐ ☐ ☐ Verify all Machine IDs used by the PFD are configurable: _____

P F N/A

☐ ☐ ☐ Verify all communication timeout values are configurable: _____

P F N/A

☐ ☐ ☐ Verify number of retries is configurable (optional): _____

P F N/A

☐ ☐ ☐ Verify connection status from PFD to PFCS: _____

P F N/A

☐ ☐ ☐ Wait for first Keep Alive Message: _____

Comments:

_____
_____
_____
_____
_____
_____
_____
_____
_____

Requesting Machine ID (if applicable)

P F N/A

☐ ☐ ☐ PFD Sends a "Keep Alive" Message to PFCS with "----" in place of the Machine ID: _____

P F N/A

☐ ☐ ☐ Using the Machine ID received in the NAK Response the PFD Re-sends the "Keep Alive" Message to PFCS: _____

Test Alarm Message (Message Type 9000 or 9001)

P ☐ F ☐ N/A ☐ PFD Sends Alarm Message to PFCS: _____

P ☐ F ☐ N/A ☐ PFD Receives a Valid ACK from PFCS: _____

P ☐ F ☐ N/A ☐ PFD Sends Keep Alive Message 2 Minutes after Sending Message: _____

P ☐ F ☐ N/A ☐ Repeat Multiple Times and verify that pages are correct format: _____

Keep Alive Messages (Message Type 9999)

P ☐ F ☐ N/A ☐ PFD Sends Keep Alive Message: _____

P ☐ F ☐ N/A ☐ PFD Waited 2 minutes after last valid message: _____

P ☐ F ☐ N/A ☐ Keep Alive Message Contains Identification and Version Number: _____

Any Other Comments Sent Along with the "Keep Alive" Message: _____

Error Recovery for Sending Alarm Messages to PFCS (Message Type 9000 or 9001)

P	F	N/A		
☐	☐	☐	No Response from PFCS:	_____
☐	☐	☐	Received a NAK-A in Response to Alarm Message:	_____
☐	☐	☐	Received an ACK-D in Response to Alarm Message:	_____
☐	☐	☐	Received an NAK-E in Response to Alarm Message:	_____
☐	☐	☐	Received an NAK-I in Response to Alarm Message:	_____
☐	☐	☐	Received an NAK-J in Response to Alarm Message:	_____

Comments:

_____
_____
_____
_____
_____
_____
_____
_____
_____
_____
_____
_____

Error Recovery for Keep Alive Messages (Message Type 9999)

P F N/A

☐ ☐ ☐ PFD receives No ACK from PFCS for "Keep Alive" Messages: _____

P F N/A

☐ ☐ ☐ Received a NAK-B: _____

Comments:

_____
_____
_____
_____
_____
_____
_____

Miscellaneous Error Recovery Procedures

P F N/A

☐ ☐ ☐ PFCS Closes Connection to PFD: _____

P F N/A

☐ ☐ ☐ PFCS Sends an Unexpected Message: _____

Comments:

_____
_____
_____
_____
_____
_____
_____
_____
_____
_____

Shutdown

P F N/A

☐ ☐ ☐ Perform PFD Shutdown Closing All Connections to PFCS: _____

P F N/A

☐ ☐ ☐ If using an RS232 Connection Check That DTR Pin is Low: _____

P F N/A

☐ ☐ ☐ Check For Peer Disconnect in PFCS: _____

Comments:

Integrated Testing

P F N/A

☐ ☐ ☐ Thoroughly Check Communications between PFD and PFCS: _____

Identify PFD limitations:_____

P F N/A

☐ ☐ ☐ Ensure that the PFD Cannot Send a Message to PFCS while Still Processing an Older One: _____

P F N/A

☐ ☐ ☐ Validate PFD Logging and Debugging Exists: _____

P F N/A

☐ ☐ ☐ Check Any User Interface and Setup Screens on the PFD: _____

P F N/A

☐ ☐ ☐ Connect the PFD to FirstPAGE PFD Server on NT Box to ensure that the PFD can communicate and send pages: _____

Identify any process issues:_____

P F N/A

☐ ☐ ☐ Unplug cable and check for peer disconnect in PFCS: _____

Comments:

PFD Documentation

P F N/A
☐ ☐ ☐ PFD Has Documentation on All Setup Procedures: _____

P F N/A
☐ ☐ ☐ PFD Has Documentation on All Error Recovery Procedures: _____

Comments:

This Checklist handles as much testing as is possible in the test environment. When the vendor installs the PFD in the plant it is the vendor's responsibility to address any production issues.

ITM:_____
NAME SIGNATURE DATE

OEM:_____
NAME SIGNATURE DATE