### Title: Passenger Car/Light Duty Truck OBD Inspection and Maintenance Flowchart

**File name:** PC-LDT OBD IM Flowchart ver8.3

**Description:** This document describes the Inspection and Maintenance process for a passenger car or light duty truck with the OBD II emission system.


<table>
<thead>
<tr>
<th>Year</th>
<th>Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>1996+</td>
<td>October 01, 2009</td>
</tr>
</tbody>
</table>

---

**A Vehicle Evaluation Guidance:**

**Passenger Car and Light Duty Truck OBD I/M (Gasoline) Flowchart**
START: CONFIRM VEHICLE

Start PC/LDT Gasoline OBD I/M Test

Assumptions:
VIN & Year, Make, Model (YMM) information are previously collected by host application

Confirm Vehicle Applicability to I/M Test Check

Query Inspector
(or use previously collected data): Is vehicle 1996 or newer model year vehicle?
Yes: Perform PC-LDT Diesel I/M Flowchart
No: Invalid vehicle for OBD testing

YES

Query Inspector
(or use previously collected data): Is vehicle engine fueled by Gasoline or Diesel?
Gasoline: Proceed to next step
Diesel: Exit PC/LDT Gasoline OBD I/M Test

Exit PC/LDT Gasoline OBD I/M Test
START: CONNECT TO SAE J1962 DLC

1. Turn ignition key or run switch off for 10 seconds

2. Connect TAS data link cable to vehicle 16 pin DLC.

Query inspector:
- Is data link cable connected?
  - <keyboard input>
  - YES
  - NO

Query inspector:
- Can data link cable be connected?
  - <keyboard input>
  - YES
  - NO

State's option:
1. Prompt inspector to enter reason cannot be connected and take appropriate action from there:
   - DLC tampering has occurred
   - DLC is inaccessible due to OEM design
   - DLC is inaccessible due to Aftermarket equipment
   - DLC cannot be located
   - Disabled motorist
2. Default to back-up test;
3. Fail car;
4. Other
Read voltage on pin 16 of SAE J1962 data link connector.

Is voltage on pin 16 equal to or greater than 11 volts?

- **YES**
  - State's option:
    - (1) Fail vehicle
    - (2) Default to back-up test;
    - (3) Other

- **NO**
  - State's option:
    - (1) Fail vehicle
      - No voltage or open circuit on pin 16 of SAE J1962 DLC
      - High resistance or open circuit on pin 4 or pin 5 of SAE J1962 DLC
      - Check:
        - Fuse for pin 16 of SAE J1962 DLC
        - Electrical wiring circuit for pin 16
        - Electrical wiring circuit for pin 4
        - Electrical wiring circuit for pin 5
    - (2) Default to back-up test;
    - (3) Other
ON-BOARD DIAGNOSTIC MIL CHECK

Set MIL_KOEO_Visual_Result=Empty
Set Comm_Result=Empty

Prompt inspector to turn key to key ON, engine OFF (KOEO), visually watch dashboard MIL.

Query inspector: Does MIL illuminate (momentarily or constant)?

<Keyboard Input>

YES

Set flag to:
MIL_KOEO_Visual_Result=Pass

Start engine and let idle (KOER).

NO

Prompt inspector to remove key from ignition

Use software timer to wait 30 seconds

Prompt inspector to insert key, turn key to key ON, engine OFF (KOEO), visually watch dashboard MIL.

Query inspector: Does MIL illuminate (momentarily or constant)?

<Keyboard Input>

YES

Set flag to:
MIL_KOEO_Visual_Result=Pass

NO

Set flag to:
MIL_KOEO_Visual_Result=Fail

Visual MIL Bulb Check

PC-LDT OBD Inspection Maintenance Test (Gasoline) - Flowchart Guide Version 8.3
Did external test equipment receive a Mode $01$, PID $00$ response from any OBD II ECU?

Attempt to establish communication to OBD II system using Mode $01$, PID $00$ request on approved protocols:


YES

NO

Set loop counter $X=0$
Set loop counter $Y=0$
Set loop counter $V=0$

Communication Check

Prompt Inspector to Check:
- Is Ignition Key/Run Switch On, Engine Running?
- Is cable firmly connected to vehicle SAE J1962 DLC?

Set flag value:
Comm_Result=Fail

SET VALUE:
Protocol_ID={ID}

Increment $X=X+1$

Does $X=3$?

YES

Set flag value:
Comm_Result=Pass

NO

14

3

15
Communication may have erronously been established only with transmission ECU. Retry Mode $01, PID $00 request to see if communication with additional modules can be established.

Example of responding OBD II ECUs:
1st-$10 (engine)
2nd-$18 (transmission)
or
1st-$41 (Chrysler transmission)
2nd-$40 (Chrysler engine)
or
1st-$12 = ECU 1 (engine control unit, bank 1)
2nd-$13 = ECU 2 (engine control unit, bank 2)
3rd-$22 = EML (power/idle control module)
4th-$32 = TCM (transmission control module)

Example of created list of OBD II ECUs:

<table>
<thead>
<tr>
<th>Module ID</th>
<th>Total_PID_Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>$10</td>
<td>24d</td>
</tr>
<tr>
<td>$18</td>
<td>6d</td>
</tr>
</tbody>
</table>

Note: "$" = hex number, "d" = decimal number

Total_PID_Count: One method to calculate this value: increment a counter by one for each PID $01-$20 (up to and including $20) indicated as supported (i.e., bit set to 1) in the Mode $01, PID $00 response message. If and only if PID $20 is supported, also request Mode $01, PID $20 and continue to increment the same counter for each PID $21-40 indicated as supported. Continue as appropriate for PIDs $40, $60, etc. if and only if the subsequent PIDs are identified as supported. Other calculation methods can be done or may be better.

Example of PID Count Look-up Table:

<table>
<thead>
<tr>
<th>YMM: 96 Chry Caravan</th>
<th>PID Cnt #1</th>
<th>PID Cnt #2</th>
</tr>
</thead>
<tbody>
<tr>
<td>ECU ID #1:</td>
<td>$40</td>
<td>18</td>
</tr>
<tr>
<td>ECU ID #2:</td>
<td>$41</td>
<td>2</td>
</tr>
</tbody>
</table>

Example of PID Count Look-up Table:

Communication may have erronously been established only with transmission ECU. Retry Mode $01, PID $00 request to see if communication with additional modules can be established.

Communication has been established with an incorrect module because a non-OBD vehicle ECU illegally responded to an OBD initialization request. Stop communication on this protocol (e.g., ISO 14230) and try to re-establish communication with one of the other protocols (e.g., ISO 9141).
Verify engine is running by sending functional message requesting engine speed (Mode $01, PID $0C) from any one of the ECUs that responded to Mode $01, PID $00 and indicated support for PID $0C

Is engine speed > 250 rpm?

YES

Query Inspector: “Engine speed below minimum expected to verify engine is indeed running. Do you want to continue the test anyway?”

<Keyboard Input>

YES

NO

Prompt Inspector to make sure engine is running (or restart engine)

Option to State to handle vehicle:
(1) Reject from OBD I/M test and treat like non-OBD vehicle;
(2) Include in OBD I/M test with special handling (e.g., ignore readiness for 1996-1997 Cadillacs, etc.)

Verify vehicle is OBD II compliant by sending functional message requesting OBD requirements (Mode $01 PID $1C) from all ECUs that responded to Mode $01, PID $00 and indicated support for PID $1C

Receive functional message response (Mode $01 PID $1C)

Determine if vehicle is OBD II compliant by confirming that PID $1C data byte is equal to $01, $03, $07, $09, $0B, or $0D. Does data byte match?

YES

Vehicle may be:
1) Non-OBD II Federal vehicle >8,500 lbs
2) Non-OBD II California vehicle >14,000 lbs
3) Non-US certified vehicle (EOBD or JOBD)
4) Not OBD compliant

NO

Optional OBD Requirements Check

Is Vehicle Under Test MY 2005 or later?

YES

Verify engine is running by sending functional message requesting engine speed (Mode $01, PID $0C) from any one of the ECUs that responded to Mode $01, PID $00 and indicated support for PID $0C

Is engine speed > 250 rpm?

YES

A

B

G

F

Is Vehicle Under Test MY 2005 or later?

NO
Send functional message requesting Mode $01, PID $01 information from all ECU_s that indicated support for PID $01.

Receive functional message response(s) for Mode $01, PID $01, and increment ECU_PID01_Msg_Response_Counter by one for each responding ECU.

Does ECU_PID01_Supported.Counter = ECU_PID01_Msg_Response_Counter?

YES

FAIL TEST
Display on report:
1) Communication failure - PID $01 message count does not match ECU count.

NO

Does Y=3?

NO

Increment Y=Y+1

YES

Not all PID $01 response messages have been received. Restart process of requesting Mode $01, PID $00

Did only one ECU respond and is Data B, bit 2 = 1 (CCM_SUP = YES) the only monitor supported (the only bit in Data B bits 0-2, Data C bits 0-7 that is = 1)?

YES

Communication may have erroneously been established only with transmission ECU. Retry Mode $01, PID $00 request to see if communication with additional modules can be established.

NO

Did only one ECU respond and did it have both:
(1) a device (module) ID of $FD;
and (2) Mode $01, PID$01, Data C, bit 4 = 1 (ACRF_SUP = YES) and Data D, bit 4 = 1 (ACRF_RDY = NO)?

YES

I/M equipment is plugged into EASE simulator instead of an actual vehicle.

NO

Did only one ECU respond and did it have both:
(1) a device (module) ID of $FD;
and (2) Mode $01, PID$01. Data C, bit 4 = 1 (ACRF_SUP = YES) and Data D, bit 4 = 1 (ACRF_RDY = NO)?

YES

Special Note:
The EASE simulator is specifically mentioned because implementers were concerned with a means to detect that simulator module and EASE agree to voluntarily configure its simulator to specifically output these criteria. This logic will not detect other simulators that may exist or be developed in the future.

NO

Special Note:
While it would be desirable to fail the car here as it likely has not received all of the OBD information that it should, there are some legitimate diesel vehicles and maybe Canadian vehicles that only have CCM supported.

Increment V=V+1

YES

Exit PC/LDT Gasoline OBD I/M Test

NO

Does V=3?

Increment V=V+1

NO

Example of responding OBD II ECU Mode $01, PID $01 monitor information:

$10 (ENGINE):
- Data B bits 0-2:
  - MIS_SUP: YES;
  - FUEL_SUP: YES;
  - CCM_SUP: YES;
- Data C bits 0-7:
  - CAT_SUP: YES;
  - HCAT_SUP: NO;
  - EVAP_SUP: YES;
  - AIR_SUP: NO;
  - ACRF_SUP: NO;
  - O2S_SUP: YES;
  - EGR_SUP: YES;
- Data D bits 4-6:
  - MIS_RDY: YES;
  - FUEL_RDY: YES;
  - CCM_RDY: YES;
- Data D bits 0-3:
  - CAT_RDY: YES;
  - HCAT_RDY: NO;
  - EVAP_RDY: NO;
  - AIR_RDY: NO;
  - ACRF_RDY: NO;
  - O2S_RDY: YES;
  - EGR_RDY: YES;

$18 (TRANSMISSION):
- Data B bits 0-2:
  - MIS_SUP: NO;
  - FUEL_SUP: NO;
  - CCM_SUP: YES;
- Data D bits 0-7:
  - CAT_SUP: NO;
  - HCAT_SUP: NO;
  - EVAP_SUP: NO;
  - AIR_SUP: NO;
  - ACRF_SUP: NO;
  - O2S_SUP: NO;
  - EGR_SUP: NO;
- Data D bits 4-6:
  - MIS_RDY: YES;
  - FUEL_RDY: NO;
  - CCM_RDY: YES;
- Data D bits 0-3:
  - CAT_RDY: YES;
  - HCAT_RDY: NO;
  - EVAP_RDY: NO;
  - AIR_RDY: NO;
  - ACRF_RDY: NO;
  - O2S_RDY: YES;
  - EGR_RDY: YES;

EUC_PID01_Msg_Response_Counter = 2

Option to State to handle result:
(1) Set internal flag to be used for audit/enforcement;
(2) Immediately abort and fail test;
(3) Other
All Mode $01, PID $01 information has been received from each OBD II ECU.

**PROCESSING PID $01 READINESS INFORMATION**

For every ECU that responds with a Mode $01, PID $01 response message, increment **Commanded_MIL_Status_Counter** by one for each ECU that indicates the MIL is commanded ON (data A, bit 7 equal to 1).

- **H**
  - Set flag value: **Commanded_MIL_Status_Counter > 0?**
    - YES: Set flag value: **Commanded_MIL_\_Result = Fail**
    - NO: Set flag value: **Commanded_MIL_\_Result = Pass**

- For every ECU that responds with a Mode $01, PID $01 response message, increment **Not_Ready_Monitor_Counter** by one for each monitor that is reported both as supported (data B, bit 0 or data C, bits 0-7 equal to 1) and incomplete (data B, bit 4 or data D, bits 0-7 equal to 1).

- Go to Readiness Look-up Table to determine value for **Allowed_Not_Ready** appropriate for model year of vehicle being inspected.

- Is **Not_Ready_Monitor_Counter > Allowed_Not_Ready?**
  - YES: Set flag value: **Readiness_Test_Result = Fail**
  - NO: Set flag value: **Readiness_Test_Result = Pass**

- For every ECU that responds with a Mode $01, PID $01 response message, add the number of stored DTCs (data A, bits 0-6) to the **Number_of_Stored_DTC_Counter**.

- For Example of decoding complete vehicle total responding OBD II ECU:
  - **Year**:
    - 1996: 2
    - 1997: 2
    - 1998: 2
    - 1999: 2
    - 2000: 2
    - 2001: 1
    - 2002: 1
    - 2003: 1
    - 2004: 1
    - 2005: 1
    - 2006: 1
    - 2007: 1
    - 2008: 1
    - 2009: 1
    - 2010: 1

---

PC-LDT OBD Inspection Maintenance Test (Gasoline) - Flowchart Guide Version 8.3

Page 10
EVALUATE VEHICLE READINESS MONITOR INFORMATION

Example 1 presents a case where the Vehicle has using 3 responding ECUs – ECM 1, ECM 2, and TCM

**Step 1** – determine Supported Readiness Monitors from all responding emission ECUs. Do this by reading data byte B, bits 0-2, and data byte C, bits 0-7:

<table>
<thead>
<tr>
<th>PID $01 Response Bytes</th>
<th>Monitor</th>
<th>ECU</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data B bits 0-2</td>
<td>MIS_SUP</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td>FUEL_SUP</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td>CCM_SUP</td>
<td>Yes</td>
</tr>
<tr>
<td>Data C bits 0-7</td>
<td>CAT_SUP</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td>HCAT_SUP</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td>EVAP_SUP</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td>AIR_SUP</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td>ACRF_SUP</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td>O2S_SUP</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td>HTR_SUP</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td>EGR_SUP</td>
<td>Yes</td>
</tr>
</tbody>
</table>

**Step 2** – Using data from only Supported Monitors, determine which Readiness Monitors are “Ready”. Do this by reading data byte B, bits 4-6, and data byte D, bits 0-7:

<table>
<thead>
<tr>
<th>PID $01 Response Bytes</th>
<th>Monitor</th>
<th>ECU</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data B bits 4-6</td>
<td>MIS_RDY</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td>FUEL_RDY</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td>CCM_RDY</td>
<td>Yes</td>
</tr>
<tr>
<td>Data D bits 0-7</td>
<td>CAT_RDY</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td>HCAT_RDY</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td>EVAP_RDY</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td>AIR_RDY</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td>ACRF_RDY</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td>O2S_RDY</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td>HTR_RDY</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td>EGR_RDY</td>
<td>Yes</td>
</tr>
</tbody>
</table>

Note: Yes or No indicates monitors that do not count because they are “Not supported” in Step 1.

**Step 3** – Determine total Vehicle Readiness Monitor Counter by applying “OR” logic to “Ready” data from each ECU. The Example vehicle below describes the case where there are three (3) Readiness Monitors which are “Not Ready”

<table>
<thead>
<tr>
<th>Monitor</th>
<th>ECU</th>
<th>Vehicle</th>
</tr>
</thead>
<tbody>
<tr>
<td>MIS_RDY</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>FUEL_RDY</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>CCM_RDY</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>CAT_RDY</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>HCAT_RDY</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>EVAP_RDY</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>AIR_RDY</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>ACRF_RDY</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>O2S_RDY</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>HTR_RDY</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>EGR_RDY</td>
<td>Yes</td>
<td>No</td>
</tr>
</tbody>
</table>

Note 1: One or more ECUs reporting not ready for a supported monitor (e.g., CAT, EVAP, EGR) results in that monitor being “not ready” for the vehicle.

Note 2: Even if two ECUs report not ready for the same supported monitor (e.g., CAT), it results in only one monitor being considered not ready for the vehicle.
GET ADDITIONAL ECU PARAMETER INFORMATION

Obtain Optional ECU Parameter Information

Send functional message requesting Mode $01, PIDs $21, $30, $31, $4D, and $4E from all OBD II ECUs that indicated support for aforementioned PIDs

Receive functional message response(s) for Mode $01, PIDs $21, $30, $31, $4D, and $4E from all OBD II ECUs

Store PID $21, $30, $31, $4D, and $4E data

Additional ECU Parameters:
- PID $21 - Distance Driven Since MIL On
- PID $30 - Number of Warmups Since DTC Clear
- PID $31 - Distance Driven Since DTC Clear
- PID $4D - Minutes Engine Run Since MIL Activated
- PID $4E - Time Since DTC Clear

J
Receive functional message response(s) for Mode $09, InfoType $00 from all OBD II ECUs.

Send functional message requesting Mode $09, InfoType $00 information from all OBD II ECUs.

Did a response to Mode $09, InfoType $00 occur?

NO

YES

Set flag value:
Electronic_VIN_Test_Result = N/A
electronic_CAL_ID_Test_Result = N/A
electronic_CVN_Test_Result = N/A

YES

Is Vehicle Under Test MY 2005 or later?

NO

YES

Set flag value:
Electronic_VIN_Test_Result = N/A
electronic_CAL_ID_Test_Result = N/A
electronic_CVN_Test_Result = N/A
EVALUATE ELECTRONIC VIN

Send functional message requesting electronic_VIN using Mode $09, InfoType $01 (if applicable) followed by InfoType $02 if any OBD II ECUs indicated support for Mode $09, InfoTypes $01 (if applicable) and $02.

Receive functional message response(s) for electronic_VIN from all ECUs that support Mode $09, InfoTypes $01 (if applicable) and $02.

Decode response.

Do any ECUs support Mode $09, InfoTypes $01 (if applicable) and/or $02 (VIN)?

YES

Send functional message requesting electronic_VIN using Mode $09, InfoType $01 (if applicable) followed by InfoType $02 if any OBD II ECUs indicated support for Mode $09, InfoTypes $01 (if applicable) and $02.

Receive functional message response(s) for electronic_VIN from all ECUs that support Mode $09, InfoTypes $01 (if applicable) and $02.

Does electronic_VIN match VIN entered by technician for inspected vehicle?

YES

Set flag value: Electronic_VIN_Test_Result = Pass

NO

Prompt inspector to re-enter VIN for inspected vehicle (bar code scanner, manual entry, etc.)

Does electronic_VIN match VIN entered by technician for inspected vehicle?

YES

NO

Set flag value: Electronic_VIN_Test_Result = Fail

Set flag value: Electronic_VIN_Test_Result = N/A
**CHECK ELECTRONIC CAL ID AND CVN**

- **L**
  - Decode response.
  - Do any ECUs support Mode $09$, InfoTypes $03$ (if applicable) and $04$ (CAL ID)?
    - **NO**
    - Send functional message requesting `electronic_CAL_ID` using Mode $09$, InfoType $03$ (if applicable) followed by InfoType $04$ if any OBD II ECUs indicated support for Mode $09$, InfoTypes $03$ (if applicable) and $04$.
      - Receive functional message response(s) for `electronic_CAL_ID` from all ECUs that support Mode $09$, InfoTypes $03$ (if applicable) and $04$.
      - Go to CAL ID/CVN Look-up Table to determine if received `electronic_CAL_ID` is valid for inspected vehicle.
    - **YES**
      - **L**
      - Decode response.
      - Do any ECUs support Mode $09$, InfoTypes $05$ (if applicable) and $06$ (CVN)?
        - **NO**
        - Send functional message requesting `electronic_CVN` using Mode $09$, InfoType $05$ (if applicable) followed by InfoType $06$ if any OBD II ECUs indicated support for Mode $09$, InfoTypes $05$ (if applicable) and $06$.
          - Receive functional message response(s) for `electronic_CVN` from all ECUs that support Mode $09$, InfoTypes $05$ (if applicable) and $06$.
          - Go to CAL ID/CVN Look-up Table to determine if received `electronic_CVN` is valid for `electronic_CAL_ID`.
        - **YES**
          - **L**
          - Is `electronic_CAL_ID` a valid CAL ID for inspected vehicle?
            - **NO**
              - Set flag value: `electronic_CAL_ID_Test_Result = N/A`
            - **YES**
              - Set flag value: `electronic_CAL_ID_Test_Result = Pass`
              - Set flag value: `electronic_CAL_ID_Test_Result = Fail`
          - **L**
          - Is `electronic_CVN` a valid CVN for `electronic_CAL_ID` of inspected vehicle?
            - **NO**
              - Set flag value: `electronic_CVN_Test_Result = N/A`
            - **YES**
              - Set flag value: `electronic_CVN_Test_Result = Pass`
              - Set flag value: `electronic_CVN_Test_Result = Fail`

**Example of CAL ID/CVN Look-up Table**

<table>
<thead>
<tr>
<th>Data</th>
<th>ECM ($11$)</th>
<th>TCM ($18$)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cal IDs</td>
<td>1</td>
<td>12586591</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>12579420</td>
</tr>
<tr>
<td>CVNs</td>
<td>1</td>
<td>00002E87</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>00001F1E</td>
</tr>
</tbody>
</table>

**Special Note:**
The CAL ID and CVN checks look for potential "clean screens" or tampered/corrupted software. CAL ID looks for "approved/certified" software to be installed in the inspected vehicle and CVN verifies the software is not corrupted or altered. At this time, it is recommended that I/M programs simply record this information at the time of inspection and use/post process the data later to look for signs of tampering/fraud/etc.
EVALUATING VIN AND OTHER ELECTRONIC INFORMATION

M

PID_Count_Test_Result:
Pass
Fail

electronic_VIN_Test_Result:
Pass or N/A

electronic_CAL_ID_Test_Result:
Fail
Pass or N/A

electronic_CVN_Test_Result:
Fail
Pass or N/A

Set flag value:
electronic_Info_Test_Result = Pass
Set flag value:
electronic_Info_Test_Result = Fail

Query inspector:
Is MIL currently illuminated? <Keyboard Input>
YES
NO

Set flag value:
MIL_KOER_Visual_Result=Fail
Set flag value:
MIL_KOER_Visual_Result=Pass
EVALUATING ON-BOARD DIAGNOSTIC MIL STATUS INFORMATION

Comm_Result:
- Fail
- Pass

MIL_KOEO_Visual_Result:
- Fail
- Pass

MIL_KOER_Visual_Result:
- Fail
- Pass

Commanded_MIL_Result:
- Fail
- Pass

Set loop counter $Z=0$
Set $M03\_Received\_DTC\_Counter=0$

Send functional message requesting Mode $03$ information (DTCs) from all ECUs.

Receive functional message response(s) for Mode $03$ PID and store DTCs for printout.
Increment $M03\_Received\_DTC\_Counter$ by one for each DTC received (not each message but each DTC!).

Special Note: DTCs are transmitted in two bytes. Two bytes of all zeroes should not be interpreted as a DTC of P0000. Two bytes of all zeroes is required when no stored DTCs are available to fill the two bytes in the message response. P0000 is NEVER a valid DTC.

Does $Number\_of\_Stored\_DTC\_Counter=M03\_Received\_DTC\_Counter$?
- NO
- YES

Set flag value: $MIL\_Test\_Result=Pass$

Set flag value: $MIL\_Test\_Result=Fail$

Increment $Z=Z+1$

For every ECU that responds with a Mode $01$, PID$01$ response message, add the number of stored DTCs (data A, bits 0-6) to the $Number\_of\_Stored\_DTC\_Counter$.

Receive functional message response for Mode $01$, PID$01$ from all ECUs.
Resend functional message requesting Mode $01$, PID$01$ information from all ECUs.

Set flag to: $Number\_of\_Stored\_DTC\_Counter=0$
Set flag to: $M03\_Received\_DTC\_Counter=0$

Does $Z=3$?
- NO
- YES

Store special footnote message to be printed on final report underneath the list of DTCs:
"This list may not reflect all DTCs currently stored in the on-board computer"
PASS/FAIL DECISION BASED ON ON-BOARD DIAGNOSTIC INFORMATION

VEHICLE FAILED OBD TEST
Display on report:
1) Vehicle Communication Failed
2) MIL_KOEO_Visual_Result=Pass/Fail
3) MIL_KOER_Visual_Result=Pass/Fail
4) Commanded_MIL_Result=n/a
5) Readiness_Test_Result=n/a
6) Unable to establish communication with vehicle. Problem may be caused by damaged/missing data link connector, vehicle communication hardware failure, or vehicle electrical problem.
7) Protocol ID = {Protocol_ID}

Special Note:
If the MIL is commanded "off" but the inspector failed the car for the bulb check or visual MIL on, it is probably worth having the technician verify the fail again. Many inspectors have falsely failed vehicles for maintenance reminder lights, ABS lights, or other non-MIL lights. Technicians may be unhappy with having to do this but the cars that legitimately fail here should be something like 0.01% of all cars.

VEHICLE NOT READY FOR OBD TEST
Display on report:
1) ECU ID
2) MIL_KOEO_Visual_Result=Pass
3) MIL_KOER_Visual_Result=Pass
4) Commanded_MIL_Result=Pass
5) Readiness_Test_Result=Fail (print out status for each monitor)
6) VIN and/or software in vehicle is not valid for inspected vehicle (print out results of electronic VIN, CAL ID, CVN, and/or PID Count)
7) Protocol ID = {Protocol_ID}

Display standard text about there may be nothing wrong with the vehicle but the vehicle has not had sufficient time to complete its self diagnostics prior to inspection and cannot be inspected at this time.

VEHICLE PASSED OBD TEST
Display on report:
1) MIL_KOEO_Visual_Result=Pass
2) MIL_KOER_Visual_Result=Pass
3) Commanded_MIL_Result=Pass
4) Readiness_Test_Result=Pass
5) Protocol ID = {Protocol_ID}

VEHICLE FAILED OBD TEST
Display on report:
1) MIL_KOEO_Visual_Result=Pass/Fail
2) MIL_KOER_Visual_Result=Pass/Fail
3) Commanded_MIL_Result=Pass/Fail
4) Readiness_Test_Result=Pass/Fail
5) List of DTCs stored (and special footnote if appropriate)
6) Protocol ID = {Protocol_ID}
STOP: FINISH TEST & DISCONNECT FROM SAE J1962 DLC

Turn ignition key or run switch off

Disconnect TAS data link cable from vehicle mounted 16 pin SAE J1962 DLC

Finish PC/LDT Gasoline OBD I/M Test