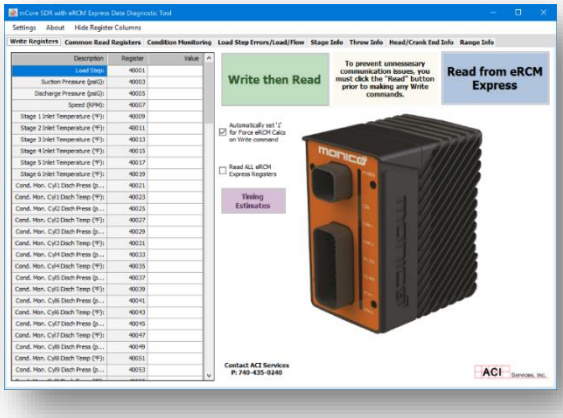
eRCM Express™

ACI's eRCM Express is an add-on device that calculates extensive compressor performance (load, flow, unit safety, etc.), provides useful information needed to control the compressor (Next Step Up, Next Step Down, allowable pressure and speed ranges, etc.), and provides for certain condition monitoring tasks (such as estimating cylinder leakages from valves and rings, based on actual discharge pressures and temperatures).

To complete the millions of calculations required to generate a full set of compressor data, for all cylinder ends, for all throws, for all stages, and for all load steps, the unit requires some time. Most unit models can be completed within 150 ms, while others may take upwards of 350 ms to complete.

- Data synchronization is critical. Most data input items, such as pressures, temperatures, and speed will vary very little from call to call, and as a result the calculated performances will generally vary slightly. Thus, while being out of synch for a half second to a second is rarely an issue, a change in load step can result in significant changes to the calculated performance data.
- **Allen-Bradley**: The mCore unit will set the ACI_PLC_ACK[1] tag to a 1 to let you know when calculations are complete, and thus the resulting data can be used.
- **Modbus**: Make sure that the PLC (or gateway) waits long enough after writing data to the unit before reading data back to use.
    - When waiting is not practical, make sure that the PLC waits at least a few write/read cycles before acting on results – especially with a load step change.
    - ACI's eRCM Express Diagnostic Software is a great free program to have installed on a PC. It provides valuable feedback which can significantly minimize efforts when checking Modbus communications and data validity.
    - Download software from the ACI website: www.ACIServices.com.



- The mCore unit allows previewing dynamic data to/from device directly via the web UI.

Topic Sections

eRCM Express™

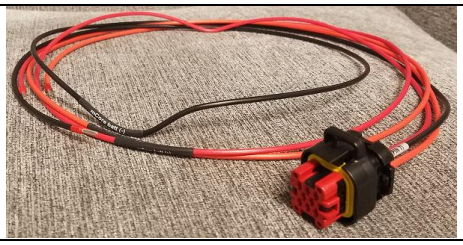
Package

Contents

Each eRCM Express unit arrives in a Monico box with foam protection, with a power cable, with a communications cable, this manual and a checklist sheet.

	
Monico Product Box (Closed)	Monico Product Box (Opened)

	
One (1) eRCM Express mCore SDR Unit	One (1) eRCM Express Manual, and One (1) Checklist Sheet

	
One (1) prewired Key Connector A. POWER CABLE About 6-foot long.	One (1) prewired Key Connector B, with RJ45 Port Connector also wired. COMMUNICATIONS CABLE

eRCM Express™

Contact and

Support

Please Contact:

ACI Services, Inc.

125 Steubenville Avenue
Cambridge, Ohio 43725

P: 740-435-0240, x538

Website: www.ACIServicesInc.com

For issues, questions, concerns, etc. related to:

- Compressor Models
- eRCM Express Product Features
- Software Diagnostics
- Integration of product features into PLC's Control Logic

Monico Inc.

18530 Klein Church Road
Spring, Texas 77379

P: 281-350-8751, Option 1

Website: www.MonicoInc.com

For issues, questions, concerns, etc. related to:

- Hardware Configuration
- Hardware Installation
- Electrical, Wiring, Cabling, and Grounding
- CDL, Communications Setup, etc.
- Certifications and Rating

eRCM Express™

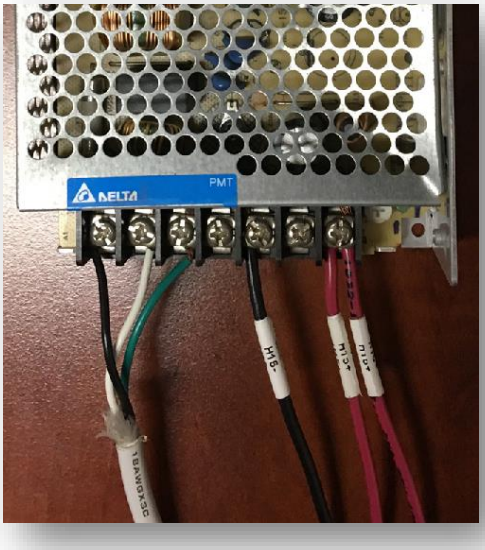
Unit

Installation

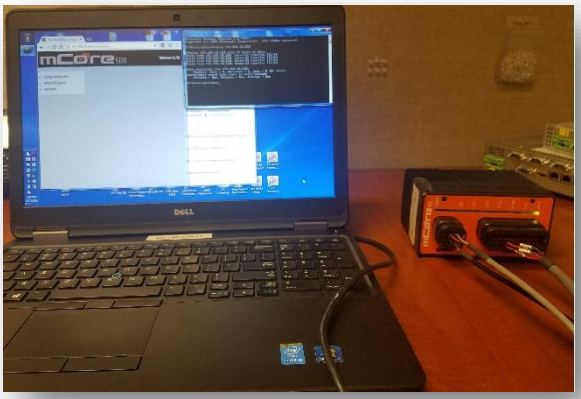
For **local workstation testing**, most users will connect the eRCM Express to a local power supply, then connect their PC to the unit to setup protocol to use, initiate communications, change IP address, and/or upload the most up-to-date compressor modeling files.

The eRCM Express unit needs connected to power and communications.

- Package includes a standard power cable with its **Key Connector A** already wired. Snap in the **Key Connector A** into its appropriate port, and then connect the three (3) wires to power supply:
    1. **Red** (H16+): Positive Terminal
    2. **Black** (H16-): Negative Terminal
    3. **Orange** (H15+): Positive Terminal
        - ***If orange wire is not connected to positive, the unit will not boot.***



- Snap in the **Key Connector B** (Communications) into its appropriate port, and then connect a PC to the eRCM Express via the Ethernet connector.
    1. Ping unit with static IP number or local device name, or
    2. Use eRCM Express Diagnostic Software, or
    3. Access with other software.
- Connect to mCore.

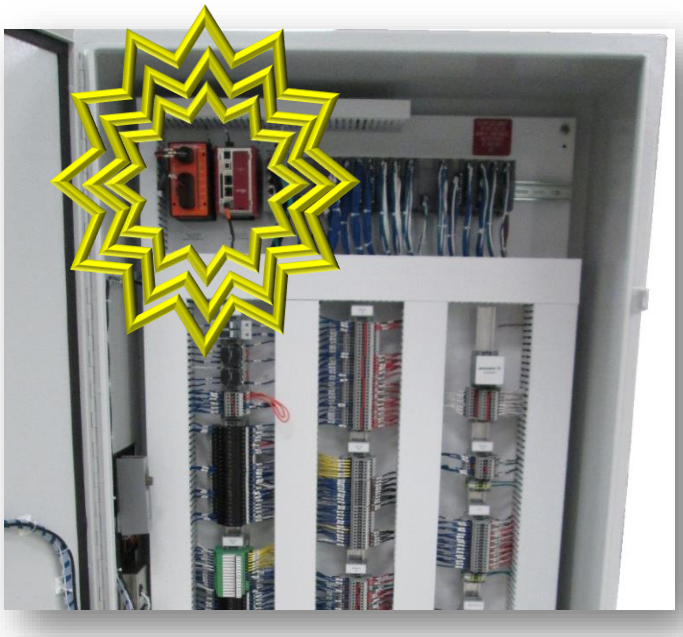


For **field installation**, most users will mount the eRCM Express into a Unit Control Panel (UCP), typically via a DIN-rail mount. Thereafter, the eRCM Express unit needs connected to power and communications.

Please reference the mCore SDR Operations Manual (Addendum V) for full hardware installation directions, including wiring, power, electrical, shielding, restrictions, and cabling.

- Package includes a standard power cable.
- Connect wires to appropriate 24 VDC power.
- When appropriate, connect ground to mCore's Grounding lug (back of device).
    1. **Red** (H16+): Positive Terminal
    2. **Black** (H16-): Negative Terminal
    3. **Orange** (H15+): Positive Terminal
        - ***If orange wire is not connected to positive, the unit will not boot.***

- Package includes a standard Ethernet connector.
- Snap in the **Key Connector B** (Communications) into its appropriate port, and then connect the PLC (or gateway device such as ProSoft Card or Red Lion) to the eRCM Express via the Ethernet connector.
- Connect to device.



eRCM Express™

mCore User Interface

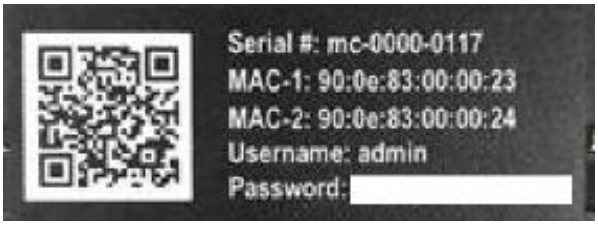
Initial Setup

Logging into a New mCore Unit

The mCore has a web-based UI (User Interface) for configuring the unit that can be accessed using a web browser. Unless the new mCore has been provisioned with a unique configuration, Ethernet 1 and 2 will be set to DHCP. On the bottom of the mCore will be a sticker with useful information and a QR code.

The label will contain the following:

- Serial Number
- MAC-1 Address
- MAC-2 Address
- Username
- Password



1. Plug the mCore's Ethernet 1 or 2 directly into a computer or network switch using an Ethernet cable.
2. Locate the Serial Number written on the label sticker on the bottom of the mCore.
3. Open a browser and enter connected mCore unit's serial number.
    - Example: If the serial number is mc-1234-5678, enter:
        i. http://mc-1234-5678.local
        ii. Or, https://mc-1234-5678.local
    - If unit was previously changed to support a static IP address, enter that IP address by itself (or use the serial number method as shown above).

The following screen appears in the browser once a connection is made:

The username is: admin
The default password is: *(see sticker on the bottom the mCore unit)*

For more details about mCore hardware, including how to Factory Reset the unit, please consult the mCore Manual.

Page intentionally left blank for end-user's notes about actual eRCM Express installation.

eRCM Express™

Hardware and

Communications

Please reference the mCore SDR Operations Manual (Addendum V) for full hardware and full communications specifications and support.

eRCM Express (mCore v1.1+) supports Allen-Bradley and Modbus communications.

When Modbus is used:

- eRCM Express is a Modbus slave,
- Use Function 16 to write to Modbus Registers, and
- Use Function 03 to read from Modbus Registers.

⚠ For **Modbus over IP**, only read a Maximum of **100** (16-bit) Registers at a time. Reading more registers than the maximum will usually cause the Modbus protocol to stop responding.

6.3 03 (0x03) Read Holding Registers

This function code is used to read the contents of a contiguous block of holding registers in a remote device. The Request PDU specifies the starting register address and the number of registers. In the PDU Registers are addressed starting at zero. Therefore registers numbered 1-16 are addressed as 0-15.

The register data in the response message are packed as two bytes per register, with the binary contents right justified within each byte. For each register, the first byte contains the high order bits and the second contains the low order bits.

Request

Function code	1 Byte	**0x03**
Starting Address	2 Bytes	0x0000 to 0xFFFF
Quantity of Registers	2 Bytes	1 to 125 (0x7D)

Response

Function code	1 Byte	**0x03**
Byte count	1 Byte	2 x **N***
Register value	**N*** x 2 Bytes	

***N** = Quantity of Registers

Error

Error code	1 Byte	**0x83**
Exception code	1 Byte	01 or 02 or 03 or 04

Here is an example of a request to read registers 108 – 110:

Request		**Response**	
Field Name	*(Hex)*	*Field Name*	*(Hex)*
Function	**03**	Function	**03**
Starting Address Hi	**00**	Byte Count	**06**
Starting Address Lo	**6B**	Register value Hi (108)	**02**
No. of Registers Hi	**00**	Register value Lo (108)	**2B**
No. of Registers Lo	**03**	Register value Hi (109)	**00**
		Register value Lo (109)	**00**
		Register value Hi (110)	**00**
		Register value Lo (110)	**64**

The contents of register 108 are shown as the two byte values of 02 2B hex, or 555 decimal. The contents of registers 109–110 are 00 00 and 00 64 hex, or 0 and 100 decimal, respectively.

⚠ **Write and read data at most two (2) times per second.**

eRCM Express™

Sending/Reading

Data Rates

Your eRCM Express box is powerful, but complex compressor and thermodynamic modeling can require millions of calculations. It will take some time for those calculations to be performed. This delay can be from 150 ms to 350 ms. It is important that new writes are not being implemented during this time, and that reading of compressor performance data is not done until after all data has been fully updated. Otherwise, results read back may be a mixture of previous point performance and current point performance.

To guarantee calculated results are synched to inputs, follow these rules.

- If using **Allen-Bradley** protocol: (*see Addendum VI for more details*)
  1. Write operating inputs to eRCM Express with the correct compressor model loaded.
  2. Write a **1** to **ACI_PLC_REQ[0] (REQUEST)** to tell mCore to run calculations.
     a. **ACI_PLC_ACK[0] (BUSY)** will go to a **1**. Calculations running. Please standby.
     b. **ACI_PLC_ACK[1] (DONE)** will go to a **1.** Calculations done.
        i. Upon a 1 in **ACI_PLC_ACK[1]**, write a **0** to **ACI_PLC_REQ[0] (RESET)**.
  3. Repeat above steps to re-run calculations.
     a. Most end users calculate compressor performance one per second.

- If using **Modbus** protocol:
  1. Write operating inputs to eRCM Express with the correct compressor model loaded.
  2. Wait long enough for all results to be calculated (**at least 150 ms**).
     a. 150 ms is the minimum wait time, with around 250 ms being common, and 350 ms should be about the max wait – it really depends on the complexity of the compressor model, number of load steps, etc.
        i. See Addendum IV for determining reasonable calculation times for models being reviewed.
  3. Read back desired results.
  4. Repeat above steps to re-run calculations.
     a. Most end users calculate compressor performance one per second.

- Throughout this manual, items will be reference by "general name `(AB tag name; Modbus Reg)`". For example:
  - **SetViewerFile** `(ACI_Inputs[50]; REG#40101)`
  - **NumStgs** `(ACI_Misc[10]; REG#40345)`

eRCM Express™

Diagnostics

Software

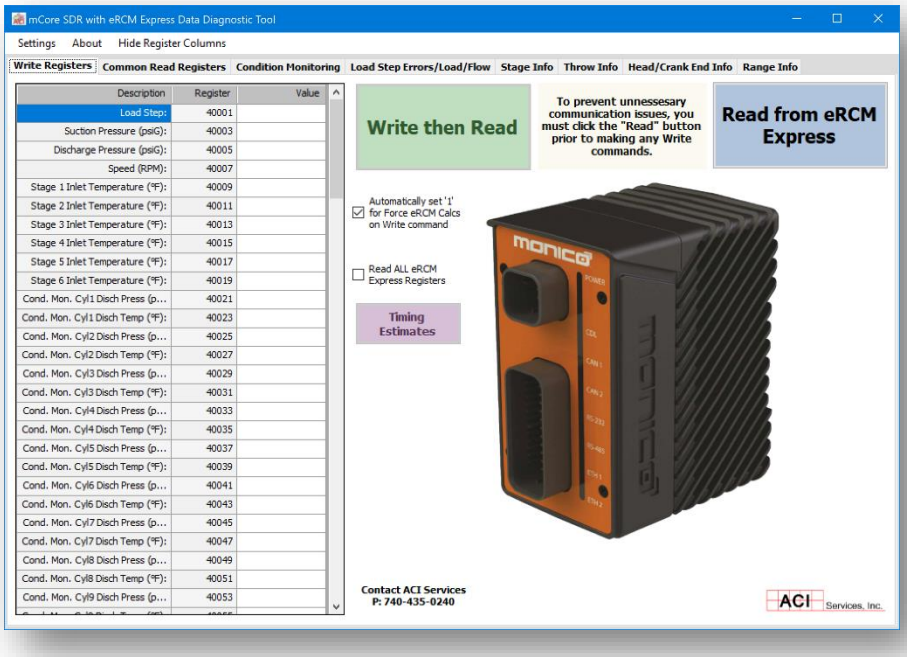
The latest version of the eRCM Express Diagnostic Software can be downloaded from the ACI Services, Inc. website.

<u>During diagnostics, the mCore must be set to act as a Modbus Slave device.</u>

With the eRCM Express unit set up (powered on, and with an active Ethernet communications line), run the eRCM Diagnostic Software on a Windows®-based PC connected to the eRCM Express unit.

Select the mCore SDR with eRCM Express item.

Click the **Read from eRCM Express** button to initiate connections and read back unit information. Now, enter values via data cells on the displayed *Write Registers* tab, then click the **Write to eRCM Express** button. If the unit is connected and communicating correctly, all appropriate registers on the remaining tabs will be populated with calculated and returned values.



NOTE: Determine roughly how long (ms) it takes to calculate full compressor performance for any model in the eRCM Express by using the **Timing Estimates** feature in the Diagnostic Software.

eRCM Express™

Uploading

Modeling File(s)

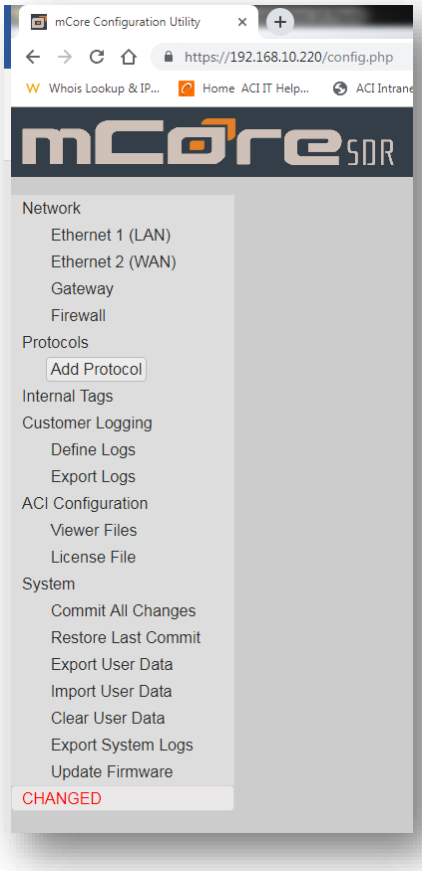
Steps to Install new eRCM Viewer™ File(s)

- Make sure desired eRCM Viewer files (*.rvf) are available for eRCM Express unit.
  a. Models should reflect correct compressor hardware, gas composition, unit staging, unloading devices, operating ranges, and unloading steps.

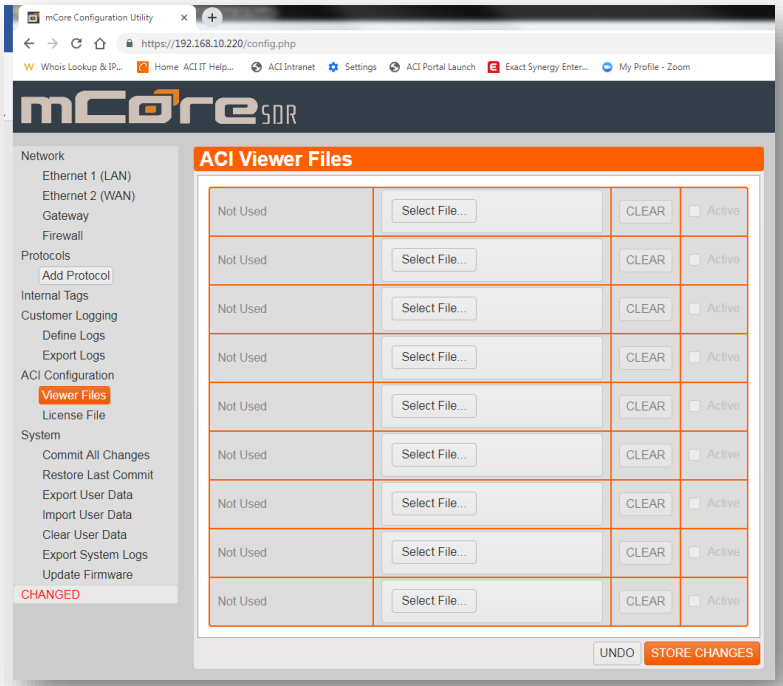
- Connect a PC to the mCore unit and then the main mCore screen will be presented.

The default account is **Admin**.
The default password is: ***(indicated on mCore stickers)***.

Once logged in, then the main panel will be shown:



- From this page, via the **ACI Configuration** menu, select **Viewer Files** to display the adjacent screen.



- To clear the unit of any previous eRCM Viewer models, click the **CLEAR** button adjacent to each model.

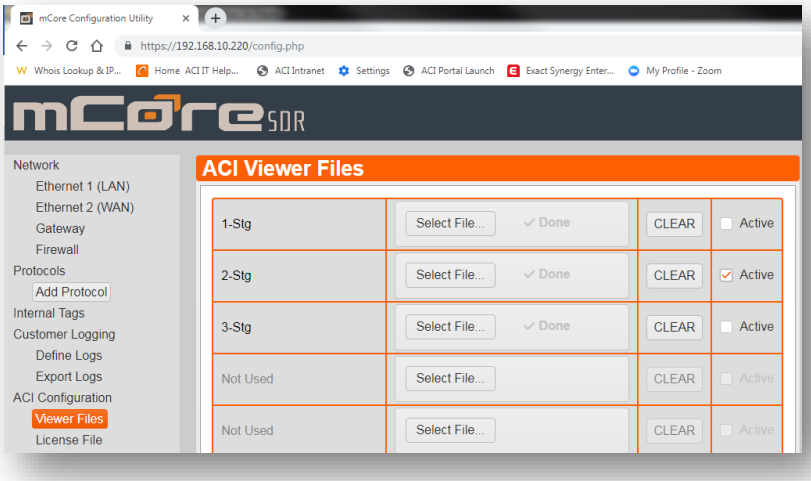
⚠

   a. **Clearing old models is recommend for users. Then upload all new models that need to be on the eRCM Express.**

- To specify new files to load onto the eRCM Express, select one of the **Select File…** buttons and select desired file. Repeat until all desired files are uploaded to eRCM Express.

⚠

   a. **One file must be set to ☑ Active before exiting web app.**



b. **NOTE**: Click on a file's filename link to download a copy of that file from the eRCM Express to PC.
    i. This is important if questions arise about "the actual unit modeling file" in the eRCM Express. Using this feature, a copy of the actual file used can readily be obtained.

- The one file set to **Active** will be the modeling file loaded during startup of the eRCM Express.
    a. *If no file is Set Active, then the eRCM Express may not function correctly.*

WARNING: All eRCM Viewer files (*.rvf) uploaded to the eRCM Express must:

- Begin with a digit "1" through "9", inclusive.
- Only use the characters 'A' - 'Z', '0' - '9', '_', and '-'.
- Use no more than 24 characters in name.
- Do not use spaces in filenames.

WARNING: Uploading more than one modeling file that starts with the exact same digit will create problems. The UI helps by preventing this.

Troubleshooting

Certain settings, or security settings, on web browser being used to interface with your mCore unit may cause undesired results, such as selected file not being uploaded to unit. Using a different browser usually eliminates these issues.

mCore actions and interactions have been checked and work with (using browser's general settings) the following common browsers: Chrome, Edge, Firefox, and IE.

eRCM Express™

Uploading a

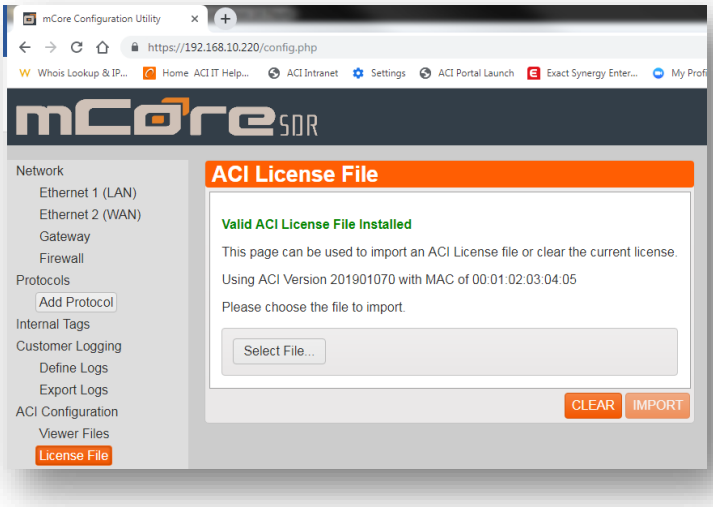
License File

Steps to Upload a New eRCM Express License File

Most end users will **never** need to do this.

1) mCore units are shipped with no license file, a valid eRCM Condition Monitoring license file, or a valid eRCM Express license file.
    a. If a device is upgraded from no license, or from a Condition Monitoring license to a full eRCM Express unit, then a new license file will need to be uploaded to access all eRCM Express features.
    b. To accomplish this, after purchase of an upgrade license, a license file will be sent (e.g. emailed). This license file is unique for each hardware unit and cannot be used on other mCore units. Thus, make sure it is installed on the appropriate unit.

2) Connect PC to the mCore unit. From the main mCore screen, select **License File** from the **ACI Configuration** menu.



WARNING: ***If the license file is cleared, and a new one is not uploaded, then the eRCM Express unit will not function as desired.***

3) From this prompt, select the **Import** button and carefully follow all prompts.

In general, end users should not remove license files unless requested by Tech Support.

eRCM Express™

Updating Unit

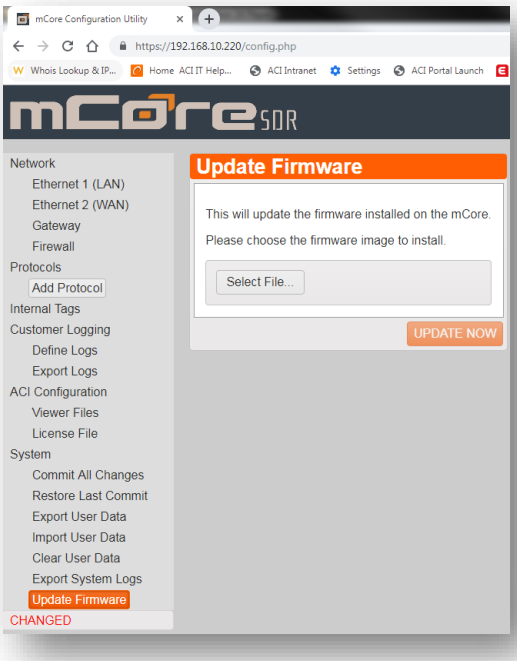
Firmware

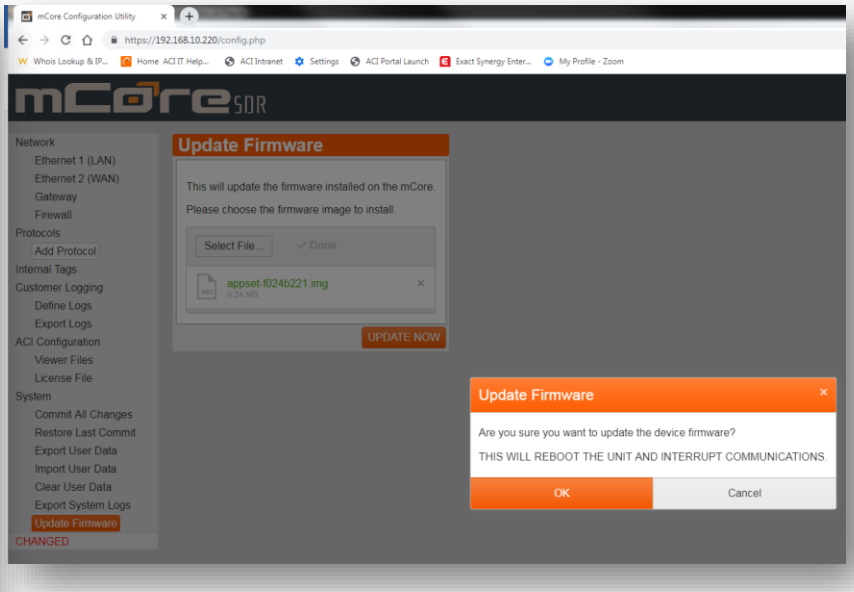
Steps to Install New eRCM Express Software

When available, Monico will provide appropriate Firmware update/upgrade files.

1) Connect PC to the mCore unit. The main mCore screen will be presented. Log in.

2) With the correct Firmware update/upgrade file(s) downloaded to a PC, select **Update Firmware** under the **System** menu, **Select File…** button, browse to file, and then select **UPDATE NOW** button and then follow any additional prompts.





It is recommended that all eRCM Viewer model(s) are re-installed for this unit after any firmware updates/upgrades are finished.

It is recommended that end users review firmware changes before installing them. Certain product changes may require coding changing in the PLC and/or gateway.

Certain firmware updates may require multiple uploads and/or multiple reboots.

Older units may require Monico support to update their firmware.

eRCM Express™

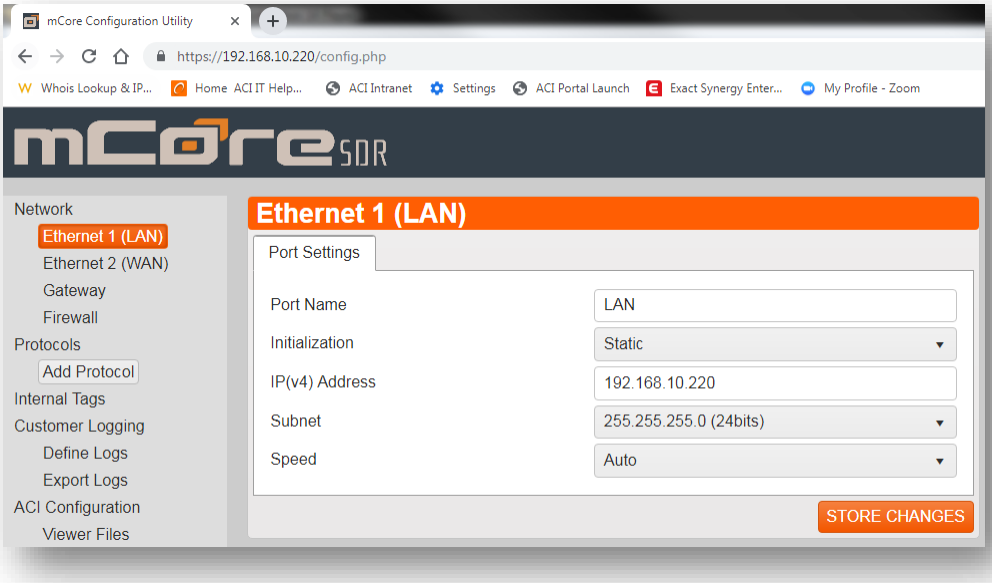
Changing LAN

Settings

Steps to Change IP Settings for eRCM Express

1) Connect PC to the mCore unit. The main mCore screen will be presented. Log in.

2) Now from the **Network** menu, select **Ethernet 1 (LAN)**.



3) Enter new IP information, and then select **STORE CHANGES**.

4) If IP numbers are lost, consult the mCore Manual on how to Factory Reset the unit.

eRCM Express™

General Process

1. When the PLC starts, if there is more than one potential eRCM Viewer model within the eRCM Express, then the PLC needs to make sure that it loads in the correct eRCM Viewer model.
   i. If there is only one (1) eRCM Viewer model, then this step can be skipped.
   ii. Otherwise, the best option is to identify which file to load (1 through 9) via the **SetViewerFile** `(ACI_Inputs[50]; REG#40101)` item – see Using Select Features (Item #1) in this manual.

2. Next, if access to the Discharge Header Pressure (the discharge line pressure the compressor will compress to **after** the bypass valve is closed) is available, then identification of the best safe load step to configure the unit before closing the bypass valve (to prevent rod load and/or pin non-reversal issues during Start Up) can be made. This is done by using the ***CheckSafeStartup*** `(ACI_Inputs[53]; REG#40107)` and the ***SafeLoadStepStartup*** `(ACI_Misc[2]; REG#40329)` registers – see Using Select Features (Item #6) in this manual.

3. If not set by the PLC, the eRCM Express will use the default values assigned to each of the following when the eRCM Viewer model was created. If different values are preferred, set them now.
   i. **SetMinPercChange** (%): Usually 1%-2%. `(ACI_Inputs[58]; REG#40117)`
   ii. **SetMaxPercChange** (%): Usually in the 15-20% range. Set lower if engine cannot handle large step changes in load, set higher if engine can handle larger step changes in load and those changes are needed in areas of the operating map. `(ACI_Inputs[59]; REG#40119)`
   iii. **LSSelectionMode**: There is a variety of these, and the unit needs to be reviewed to decide when one (or ones) are best suited for the unit. *Units with automated VVCPs are generally set to specific modes that **cannot** be changed via the PLC.* `(ACI_Inputs[52]; REG#40105)`
      i. See Addendum I for more details regarding LS Selection Modes.

4. If desired for an HMI screen, read the **Ranges** for key items and store them in the PLC. These do not change until a new eRCM Model is loaded, so this set of registers does not need to be continually read. `ACI_Ranges[0] to ACI_Ranges[24]; REG#s 43605 through 43647`.

5. Read the following set of items, and store in PLC. These do not change until a new eRCM Model is loaded, so this set of registers does not need to be continually read.
   i. `ACI_Misc[3]; REG#40331`, AuxLoad (HP)
   ii. `ACI_Misc[4]; REG#40333`, BHPMax (HP)
   iii. `ACI_Misc[6]; REG#40337`, Elevation (ft)

iv. `ACI_Misc[7]; REG#40339`, AtmPress (psiA)
v. `ACI_Misc[8]; REG#40341`, NumCyls (#)
vi. `ACI_Misc[9]; REG#40343`, NumLSs (#)
vii. `ACI_Misc[10]; REG#40345`, NumStgs (#)
viii. `ACI_Misc[11]; REG#40347`, NumThrws (#)
ix. `ACI_Misc[12]; REG#40349`, OEM_ID (#)
x. `ACI_Misc[14]; REG#40353`, MaxDischF (degF)
xi. `ACI_Misc[15]; REG#40355`, RelHumid (%)

6. After the compressor and driver have warmed up, and ready to close the Bypass Valve, then set the compressor's load step to:
   i. The load step identified in Item #2, **or**
   ii. The load step Operations has identified to use when starting a unit.

7. Close the Bypass Valve

8. When the Bypass Valve is fully closed, then the unit is Online.
   i. At this point, the eRCM Express now can provide useful compressor performance and safety data that can be used to properly control the online compressor.
      i. **NOTE**: *Performance data returned via the eRCM Express before the bypass is closed is not considered valid data and thus it should not be used for alarms, shut downs, etc.*

9. WHEN UNIT IS ONLINE: **(Main Loop)**
   i. Set the Current Load Step in eRCM Express via **CurrLS** `(ACI_Inputs[0]; REG#40001)`
   ii. From unit sensors, read inlet pressure (Stage #1 Suction Pressure **PsG**) and inlet temperature (Stage #1 Suction Temperature, **Ts1F**), the last stage's discharge pressure (Discharge Pressure, **PdG**), and the suction temperature to each stage after the 1st stage **Ts2F..Ts6F**, and the current operating speed **CurrSpeed**.
   iii. Pass this information to eRCM Express via:
      i. **PsG** `(ACI_Inputs[1]; REG#40003)`,
      ii. **PdG** `(ACI_Inputs[2]; REG#40005)`,
      iii. **CurrSpeed** `(ACI_Inputs[3]; REG#40007)`,
      iv. **Ts1F** `(ACI_Inputs[4]; REG#40009)`, and as needed
         1. **Ts2F..Ts6F** `(ACI_Inputs[5]..ACI_Inputs[9]; REG#40011..REG#40019)`.
   iv. Set the Maximum Torque Limit (%) **TorqSP** `(ACI_Inputs[54]; REG#40109)`.

i. Most <u>all</u> users just need to set this to 100%.

v. Force an eRCM Express calculation by putting the value of 1 into
- If using **Allen-Bradley** protocol,
  - Set ACI_PLC_REQ[0] to 1.
    - ACI_PLC_ACK[0] will go to a 1 when calculations have started.
    - ACI_PLC_ACK[1] will go to a 1 when calculations have finished.
    - Set ACI_PLC_REQ[0] to 0.
- If using **Modbus** protocol: **ForceERCMExpressCalculations** `(ACI_Inputs[37]; REG#40273 or ACI_Inputs[136]; REG#40075).`
  - Do not try to retrieve any performance data back from the eRCM Express until the value in **IsKernelBusy** (REG#40275) goes to zero (0). This item changes to one (1) when eRCM Express starts the compressor calculations and changes back to a zero (0) when those calculations are complete. This happens very fast (less than 250 ms for most calls).
    - **<u>NOTE:</u>** *If polling the IsKernelBusy is not easily achievable via the PLC or gateway, then next best option is to just wait about **250 ms** after writing the Write data items. It will generally be finished by then.*

vi. When calcualtions are compelte, then read the following items:

==========

i. eRCMExpressWatchdogPulse `(ACI_Control[1]; REG#40277)`
ii. NextLoadStepUp `(ACI_Control[2]; REG#40279)`
iii. NextLoadStepDown `(ACI_Control[3]; REG#40281)`
iv. NextLSUpPercentChange `(ACI_Control[4]; REG#40283)`
v. NextLSDownPercentChange `(ACI_Control[5]; REG#40285)`
vi. MinSpeedCurrentLS `(ACI_Control[6]; REG#40287)`
vii. MaxSpeedCurrentLS `(ACI_Control[7]; REG#40289)`
viii. MinSuctPressureCurrentLS `(ACI_Control[8]; REG#40291)`
ix. MaxSuctPressureCurrentLS `(ACI_Control[9]; REG#40293)`
x. FindOptimalLoadStep `(ACI_Control[10]; REG#40295)`
xi. CurrentTorque `(ACI_Control[11]; REG#40297)`
xii. IsentropicEfficiency `(ACI_Control[12]; REG#40299)`
xiii. Fuel Rate `(ACI_Control[13]; REG#40301)`

ITEMS TO REVIEW, AND ACTIONS TO TAKE IF NEEDED:

1. **<u>Need for Shut Downs:</u>**
    a. No safe Load Steps present (either method valid):
        i. If NextLoadStepUp = -1 AND NextLoadStepDown = -1 AND FindOptimalLoadStep = -1 then Shut Down. No exceptions.
        ii. If NextLoadStepUp = -1 AND NextLoadStepDown = -1 AND ErrorArray(CurrLS) ≠ 0 then Shut Down. No exceptions.
    b. Communication Issues:
        i. If communications errors exist for more than a couple seconds, or if more than five (5) consecutive errors occur, then Shut Down or go into one of the eRCM Express Offline Modes.
        ii. If WatchdogPulse does not change after setting items when forcing a new calculation and waiting long enough for calculations to finish, then Shut Down or go into one of the eRCM Express Offline Modes.
    c. eRCM Express Offline Modes:
        i. If communication loss happens, then the PLC can try to keep the unit online and running (after issuing appropriate alarms). To do this, PLC needs to record the Ps, Pd, and Ts's at time of lost communications. Then, it must hold Load Step and Speed at current settings. The unit can stay online (and still be safe without its eRCM Express connection) provided:

Method I – Current Operating Pressures and Temperatures Hold Relatively Constant

1. MaxPsChange = (Ps.Max - Ps.Min) / 100 // Units: psi, rpm, °F
2. MaxPdChange = (Pd.Max - Pd.Min) / 100
3. **IF ABS**(CurrentPs – Ps) ≤ MaxPsChange **AND**
   ABS(CurrentPd – Pd) ≤ MaxPdChange **AND** **ABS**(CurrentRPM – RPM) ≤ 8 **AND**
   ABS(CurrentTs1 – Ts1) ≤ 5 **AND** **ABS**(CurrentTs2 – Ts2) ≤ 5 **AND**
   ABS(CurrentTs3 – Ts3) ≤ 5 **AND** **ABS**(CurrentTs4 – Ts4) ≤ 5 **AND**
   ABS(CurrentTs5 – Ts5) ≤ 5 **AND** **ABS**(CurrentTs6 – Ts6) ≤ 5 **THEN** <u>Unit is Safe</u>
   ELSE <u>Need to Shut Down Unit</u>

If the above holds true, then the unit's operating conditions have not changed so much that the unit has drifted in territory where safety is unknown.

Method II – Using Pre-checked Safe Operating Maps

1. **IF** (CurrentPs ≥ OfflineMinSafePs1 **AND** CurentPs ≤ OfflineMaxSafePs1 **AND**
   CurrentPd ≥ OfflineMinSafePd1 **AND** CurentPd ≤ OfflineMaxSafePd1) **OR**
   (CurrentPs ≥ OfflineMinSafePs2 **AND** CurentPs ≤ OfflineMaxSafePs2 **AND**
   CurrentPd ≥ OfflineMinSafePd2 **AND** CurentPd ≤ OfflineMaxSafePd2)
   THEN *Unit is Safe*
   ELSE *Shut Down.*

If the above holds true, then the unit's operating conditions have already been prechecked for safety, and unit is safe to run in these areas.

- *Note: This method is only useful if the eRCM Express unit is set to calculate safe offline operating maps, and this only happens when* ***EnableSafeOfflinePressRangesCalcs*** `(ACI_Inputs[62]; Reg#40125)` *is set to 1.*
  - *Before using this mode, make sure that when it is engaged it does not slow down your eRCM Express.*

2. Adjustments to PLC Setpoints:

a. Set the PLC's Min Allowed Speed to MinSpeedCurrentLS
b. Set the PLC's Max Allowed Speed to MaxSpeedCurrentLS
c. Set the PLC's Min Allowed Suction Pressure to MinSuctPressureCurrentLS
d. Set the PLC's Max Allowed Suction Pressure to MaxSuctPressureCurrentLS

3. Controls:

a. If there is an active call for More Load (or more Flow), then the PLC controls changing loading (sample device preferences given here, but actual device preferences determined by control logic in the PLC):
   i. If Recycle Valve not closed, start to Close Recycle, then
   ii. If available, Unpinch Suction Throttle (up to MaxSuctPressureCurrentLS), then
      1. *Ok for flow, use caution with power.*
   iii. If not at Max Allowed Speed, Increase Speed, then
   iv. If NextLoadStepUp ≠ -1, change load step to NextLoadStepUp.
b. If there is an active call for Less Load (or less Flow), PLC to change loading:
   i. If not at Min Allowed Speed, Decrease Speed, then

ii. If NextLoadStepDown ≠ -1, change load step to NextLoadStepDown.
iii. If available, Pinch Suction Throttle (down to MinSuctPressureCurrentLS), then
    1. *Ok for flow, use caution with power.*
iv. Start to Close Recycle Valve.

==========

xiv. Read in ErrorArray `(ACI_ErrorArray[0..99]; REGs#40445 to 40443+NumLSs*2)`
    1. If NextLoadStepUp = -1 AND NextLoadStepDown = -1 AND ErrorArray(CurrLS) ≠ 0 then Shut Down.

xv. Read in LoadArray `(ACI_LoadArray[0..99]; REGs#40645 to 40643+NumLSs*2)`

xvi. Read in FlowArray `(ACI_FlowArray[0..99]; REGs#40845 to 40843+NumLSs*2)`
    1. Display data from above three arrays on one HMI screen (as a table) to show operators which load steps are valid, and the potential flow rates those load steps can deliver and their associated required loads.
    2. Many end users highlight the current Load Step row, as well as identifying which rows are for Next Load Step Up and Next Load Step Down (when those items are valid, i.e. not -1).

xvii. Read in StageArray `(ACI_StageArray[0..179]; REGs#41045 to 41043+NumStgs*60)`
    1. Display data on HMI

xviii. If/when key Throw Data is desired to be displayed on HMI screen, then:
    1. Read in ThrowArray `(ACI_ThrowArray[0..299]; REGs#41405 to 41403+NumThrws*60)`
        a. Display data on HMI

xix. If/when key HE Cylinder Data is desired to be displayed on HMI screen, then:
    1. Read in HECylinderInfo `(ACI_CylHEArray[0..399]; REGs#42005 to 42003 + NumCylinders*80)`
        a. Display data on HMI

xx. If/when key CE Cylinder Data is desired to be displayed on HMI screen, then:

1. Read in CECylinderInfo `(ACI_CylCEArray[0..399]; REGs#42805 to 42803 + NumCylinders*80)`
    a. Display data on HMI

vii. Go back to Item #9 (WHEN UNIT IS ONLINE section).

eRCM Express™

Pseudo Code

Examples

Modbus Read/Write Pseudo Code Examples

Sending Data to eRCM Express™:

```
  Modbus.Data(1)  = 1     // Current Hardware Load Step
  Modbus.Data(2)  = 206.5 // Current Suction Pressure (psiG) into First Stage
  Modbus.Data(3)  = 318.9 // Current Discharge Header Pressure (psiG) out of Last Stage
  Modbus.Data(4)  = 1158  // Unit's Speed (RPM)
  Modbus.Data(5)  = 74.5  // First Stage Suction Gas Temperature (degF)
  Modbus.Data(6)  = 128.2 // Second  Stage Suction Gas Temperature (degF), when required

  Modbus.NumberOfValues = 6   // 6 is the number of input values (1 Value = two 16-bit regs)
  Modbus.Address = 40001      // 40001 is the starting address for the input values
  Modbus.WriteData()          // Write new input values to the Modbus registers.
```

Force eRCM Express™ to Recalculate New Compressor Performance:

```
  Modbus.Data(1)  = 1         // "1" is flag to recalculate
  Modbus.NumberOfValues = 1   // 1 is the number of input values (1 Value = two 16-bit regs)
  // Reg#40075 is ideal if only using the shorter Quick Writes list (smaller Modbus block)
  Modbus.Address = 40075      // 40075 is the address for ForceERCMExpressCalculations
OR  Modbus.Address = 40273    // 40273 is the address for ForceERCMExpressCalculations
  Modbus.WriteData()          // Write value to the Modbus register.
```

Waiting for eRCM Express™ to Complete Calculations – Using IsKernelBusy Method:

```
  DIM eRCMExpressCount as Integer
  Modbus.NumberOfDataValues = 1
  Modbus.Address = 40275  // The address of the "IsKernelBusy" register.
                          // Will return false (0) when eRCM Express has completed
                          // all of its performance and safety calculations.
                          // This usually takes less than 250 ms.
  eRCMExpressCount = 0
  DO
    Sleep 20                // Sleep 20 ms
    Modbus.ReadData()       // Read data in the "IsKernelBusy" register.
    If Modbus.Data(1) = 0 Then Exit DO     // If 0, then calculations are now complete.
    eRCMExpressCount = eRCMExpressCount + 1
  LOOP UNTIL eRCMExpressCount > 15         // Prevent Infinite Loop

  IF eRCMExpressCount > 15 THEN
    MSG "CRITICAL ISSUE: eRCM Express Unit does not appear to be returning data."
    // NOTE: If a Gateway device is between eRCM Express and PLC, then you may need
    // to wait longer for updates to happen.
  ELSE
    …Now proceed with reading in eRCM Express data…
  END IF
```

Waiting for eRCM Express™ to Complete Calculations – 250 ms Method:

```
  Sleep 250   // Sleep at least 150 ms. eRCM Express will be done by then for MOST models.
              // If model has lots of load steps (20+) or multiple stages (2+) then
              // more time may be required to finish all of the calculations.
              // 250-350 is usually the MOST you need to wait.
  …Now proceed with reading in eRCM Express data…
```

Reading Data from eRCM Express™:

```
  DIM NextLoadStepUp as Double
  …
  DIM LoadStepBHPs(23) As Double

  Modbus.NumberOfDataValues = 1
  Modbus.Address = 40275  // The address of the "IsKernelBusy" register.
                          // Will return false (0) when eRCM Express has completed
                          // all of its performance and safety calculations.
                          // This usually takes less than 150 ms.
  Modbus.ReadData()               // Read data in the "IsKernelBusy" register.
  If Modbus.Data(1) = 0 Then      // If 0, then calculations are now complete.
        // Do the following to read specific items:
        Modbus.NumberOfDataValues = 13 (1 Value = two 16-bit registers)
        Modbus.Address = 40277      // Start of a block of data
        Modbus.ReadData()
        eRCMExpressWatchdogPulse  = Modbus.Data(1)
        NextLoadStepUp            = Modbus.Data(2)
        NextLoadStepDown          = Modbus.Data(3)
        NextLSUpPercentChange     = Modbus.Data(4)
        NextLSDownPercentChange   = Modbus.Data(5)
        MinSpeedCurrentLS         = Modbus.Data(6)
        MaxSpeedCurrentLS         = Modbus.Data(7)
        MinSuctPressureCurrentLS  = Modbus.Data(8)
        MaxSuctPressureCurrentLS  = Modbus.Data(9)
        FindOptimalLoadStep       = Modbus.Data(10)
        CurrentTorque             = Modbus.Data(11)
        IsentropicEfficiency      = Modbus.Data(12)
        FuelRate                  = Modbus.Data(13)

        // Do the following to read an array of values at once:
        // Example: Get the load for load steps 1 through 23
        Modbus.NumberOfDataValues = 23
        Modbus.Address = 40645 // Modbus address for start of LoadArray values (Load Step 1)
        Modbus.ReadData()
        LoadStepBHPs(1) = Modbus.Data(1)    // Load for Load Step 1  (Address 40645)
        LoadStepBHPs(2) = Modbus.Data(2)    // Load for Load Step 2  (Address 40647)
        LoadStepBHPs(3) = Modbus.Data(3)    // Load for Load Step 3  (Address 40649)
        LoadStepBHPs(4) = Modbus.Data(4)    // Load for Load Step 4  (Address 40651)
…
        LoadStepBHPs(23) = Modbus.Data(23)  // Load for Load Step 23 (Address 40689)
…
  END IF
```

eRCM Express™

Using Select

Features

1. **How do I make sure that the correct eRCM Viewer model is loaded into eRCM Express?**
    a. If there is only one (1) eRCM Viewer model used, then it is loaded by default and no check is usually needed.
    b. If there are multiple unit models loaded into eRCM Express (or you want to verify default file for security), then run some code during start up to check:
        i. A default model will load in when eRCM Express starts. This may, or may not, be the desired model.
        ii. If there are multiple eRCM Viewer models on the eRCM Express, then the PLC needs to identify which one it wants to load (and then verify it afterwards). This is done via employing the **File Identification Method**, which is describe in more detail under the "Change Current Modeling File via PLC" section`.
    c. Best to wait at least 500 ms after loading a new file before checking it.
    d. Also, see section on **Sample Pseudo Start Up Code**

2. **How can I verify that eRCM Express is working, via code?**
    a. Check the value in the **eRCMExpressWatchdogPulse** register. If this value changes after your next call to **ChangeOpCondition** then eRCM Express is working as intended. The returned value will always be an integer from 0 to 6 million and should never be the same as the previous number.

3. **How do I know if the eRCM Express is working and returning correct values in the unit's registers?**
    a. Install ACI's eRCM Express Diagnostics & Communications Software (available free from the ACI website). Install that software onto your Windows PC, set up an Ethernet connection from your PC to the eRCM Express unit and run the software. Reference that software's Help for more information about how to effectively use it.
    b. This software can readily help troubleshoot communications issues by eliminating the PLC and gateway devices from the chain – Modbus only!

4. **What do I change to make eRCM Express select load steps using larger (or smaller) increments?**
    a. Set the percent of desired change (for either load or flow) in the register **SetMinPercChange**. Subsequent changes to conditions will lead to determination of **NextLoadStepUp** and **NextLoadStepDown** based on the new percent change value.

5. **How do I force eRCM Express to select load steps based on different criteria?**

a. Please review the section **Addendum I – Load Step Selection Modes**.
b. Models using automated VVCPs may not allow changes to their methods of load step selection.
c. The eRCM Viewer models should be set by default to the most useful load step selection based on the unit, the load steps, and the operation map.
d. LSSelectionMode `(ACI_Inputs[52]; Reg#40105)` is used to set load step selection mode.
e. Setting (and resetting) this item is common when going into (and then back out of) a control panel's Manual Control Mode.

6. **How do I determine which load step to set the unit to before closing the bypass valve during Start Up?**
    a. To determine which load step to set the hardware configuration to, before closing the bypass valve, set the current operating conditions (**PsG**, **PdG**, **CurrSpeed**, **Ts1F**, …), and then set **CheckSafeStartup** to 1 and then force a calculation by setting **ForceERCMExpressCalculations** to 1. When calculations are complete the **SafeLoadStepStartup** register contains the load step to use.
        i. **NOTE**: For this option to be useful, during this mode you must send the discharge line pressure (***after the bypass valve***), and not a cylinder discharge pressure (before the bypass valve), when you set the current operating condition for **PdG**.
        ii. **WARNING**: You must set this register (**CheckSafeStartup**) back to zero (0) after retrieving the safe load step to use during Start Up. If left non-zero, subsequent eRCM Express data may not be appropriate.

7. **What does *MaxLoadPercChange* do?**
    a. The value in **MaxLoadPercChange** is used to prevent eRCM Express from selecting a load step that could result in undesirable consequences to the engine. A significant and sudden change in load (increase or decrease) can lead to engine problems (over-speeding or under-speeding or surging). For many engines, the maximum load change limit is about a 15% change in load, whilst some may go as high as 25-30%. For electric motors, the limit may be higher since electric flows react faster than fuel flows.

8. **What's the logic behind selecting Next Load Step Up (and Down)?**
    a. In general, based on the unit load (or flow) of the Current Load Step, eRCM Express identifies which load steps are at least ***MinLoadFlowPercChange*** percentage points higher (or lower for Next Load Step Down) but no more than ***MaxLoadPercChange*** percentage points away. Then, the closest safe

load steps to that limit are selected. If more than one load step is reasonable, then the load step with the best Load per Unit Flow ratio is selected (as it is the most efficient).

b. However, more detailed information can be found at **Addendum I – Load Step Selection Modes**.

9. I'm not matching the correct bits set in the registers defined by ErrorArray().

a. The bits in ErrorArray() can only be properly identified when their two (2) 16-bit Modbus Integer registers are correctly interpreted as a 32-bit Long.

i. The most common issues are:

1. The two (2) 16-bit Modbus registers are juxtaposed. If so, swap the two 16-bit register locations when creating the 32-bit Long.

2. Data is read as a Floating Point instead of an Integer.

10. How do I use the Live Pressures feature? *(Future Option)*

a. *Check back with ACI for future firmware updates that support this option.*

11. How do I use one of the Special 2-Stage Modes – the mode with Dynamically Adjustable Interstage Pressure, or the mode with Dynamically Adjustable Side Stream Out amounts. (*Only 2-stage models supported*.)

a. If your eRCM model is set for adjustable interstage pressure, then:

i. Set the unit's controlled interstage pressure (psiG) via the **Stage 1 Pd** `(ACI_Inputs[31]; Reg#40063)`. If entered amount is outside of allowed values, then it will be clipped to upper/lower limits. Set to zero (0) to exit this mode and return to modeling unit as a true 2-stage compressor.

ii. Set this value before forcing an eRCM Express update.

b. If your eRCM model is set for adjustable side stream out amounts, then:

i. Set the Side Stream Out amount (MMscfd) via the **SS12** `(ACI_Inputs[70]; Reg#40141)`. Since this is a Side Stream Out, it should be a negative number. Set to 0 for no side stream out. If entered amount is outside of allowed values, then it will be ignored (i.e. side stream out of 0 MMscfd).

ii. Set this value before forcing an eRCM Express update.

12. What version of the eRCM Express Software do I have?

a. Retrieve the value in eRCMExpressVER `(ACI_Misc[16]; Reg#40357)`.

13. What can be done if communications are lost to keep unit up and running?

a. If communications with eRCM Express are lost, PLC then to prevent changes to speed and load step. Then, based on last good data read, if current P_S and

P_D stay within limits (see ACI_Misc[17]..ACI_Misc[24]; Reg#'s 40359..40373), then unit will remain safe. ***Unit operations may not be efficient or optimized, but they will be safe.***

eRCM Express™

Change Current

Modeling File

via PLC

File Identification Method:

At times, multiple eRCM Viewer files will be loaded and ready in the eRCM Express unit. Examples may include:

- One model for 1-stage injection, one model for 2-stage injection, and a third model for 1-stage withdrawal.
- One model for process gas, and another model for purge gas.
- One model for unit with no valve spacers, one for all suction valve spacers in, one for all valve spacers in, and one model for suction valves pulled.

eRCM Viewer files must be specifically identified with a leading digit of "1" through "9" in their filenames.

- **Warning**: Filenames with the same leading digit will lead to confusion. Thus, do not use filenames like "**1**-0_Unit-12.rvf" and "**1**-1_Unit-12b.rvf" as both have the same leading digit "**1**".
    - ***NOTE: Do not put spaces and special characters in filenames.***
    - ***NOTE: Length of filenames must be less than or equal to 24 characters.***

- Sample filenames that Can and Cannot be used:

Filenames that Can be Used	Filenames that Cannot be Used
1Unit7-Ariel_2-Stg.rvf	Unit7-Ariel 2-Stg.rvf *(has spaces, does not start with digit)*
8Superior_MH6_1-Stage.rvf	Superior MH6 Single Stage Unit.rvf *(has spaces, too many characters in filename)*
2.rvf	Rev2.rvf *(does not start with digit)*
6_GE-DS_Service1-CNG.rvf	6GEDS_Service1@CNG.rvf *(has invalid character)*

- To select the file to use, simply set the register ***SetViewerFile*** `(ACI_Inputs[50]; Reg# 40101)` to the special ID number of that file, from "1" to "9".
    - Best to wait at least 500 ms after loading a new file before checking it.

- Sample Pseudo Code for this Method

```
' Load in file "4CooperBessermer_2-Stg.rvf" into eRCM Express
Modbus.NumberOfDataValues = 1
Modbus.Address = 40101
Modbus.Data(1) = 4     // Load in eRCM Model that starts with the digit "4"
Modbus.WriteData()

Sleep 500 ms
' Correct model should now be loaded in eRCM Express.

' Check it (NumStages, NumThrows, NumLSs, etc. to make sure it is correct)
```

  - See section on **Sample Pseudo Start Up Code**

eRCM Express™

Miscellaneous

Notes

Relax eRCM Limits to Station Limits:

When implementing a compressor model in a Unit Control Panel environment, end-user should set certain limits within the model so that those limits do not trigger an invalid condition before the PLC would trigger an invalid condition.

- Example 1: If the PLC alarms at a discharge temperature of 300 °F and shut downs at a discharge temperature of 325 °F, then the maximum allowed discharge temperature in the eRCM Viewer model should be 325 °F, and not 300 °F.
- Example 2: If the unit is allowed to go down to 40% torque, then make sure that the Min Allowed Torque setting in the eRCM Express is 40% or lower.

The goal is to not have the eRCM Express trigger unnecessary shut downs by indicating all load steps as unsafe, when in fact they may still be within the PLC's limits.

Using Compressor to Pack a Discharge Line:

If the unit needs to pack a discharge pipe that has fallen below its normal minimum discharge pressure, then ensure that the following limits are set in the eRCM Viewer model being used in the eRCM Express unit:

- Minimum Compression Ratio = 1.00
- Minimum Discharge Pressure = Minimum Suction Pressure

Running Unit at Low Torque:

If the unit needs to occasionally run in low torque conditions, then make sure that in the eRCM Viewer model being used:

- Minimum Torque = A number from 1% to 60%.
  Typical Default for Motors is 25%, and Typical Default for Engines is 60%.

Running Unit when Discharge Pressure is Less Than Suction Pressure:

If the unit needs to occasionally run when the discharge header pressure is less than the suction header pressure, then:

1) Make sure that this is okayed by the OEM. Operations in blow-through can potentially significantly degrade the life of compressor valves.
2) eRCM Express is not useful for determining unit safety under these conditions (blow-through). So, if that mode is ok with the OEM, then to keep eRCM Express from indicating operating issues, simply set the Discharge Pressure equal to the Suction Pressure+5 psi. Thus, if Ps=390 psiG, and Pd=340 psiG, when sending data to eRCM Express, simply send 390 for Ps, and 395 for Pd.

eRCM Express™

Allen-Bradley Tag Name; Modbus Registers

Commonly Used Lists

*e*RCM™ Express Quick Write List

Modbus Mode *(on default Port 502 – port number can be changed in mCore)*:

eRCM Express acts as a Modbus slave using Ethernet/IP protocol Q414.

- Use Function 16 to write Operating Point Inputs to Modbus Registers 40001- 40273.
- Use Function 03 to read back CPASA Outputs from Modbus Registers 40275- 43653.

- ***NOTE****: All registers (read & write) are 32-bit Floating Point type,* ***except*** *for those between 40445 and 40643 inclusive which form the ErrorArray() and all of these represent 32-bit Integer type. Modbus usually reads items as 16-bit registers, so make sure the two 16-bit registers are transferred correctly to create the final 32-bit Floating Point number (or Integer number).*

- ***NOTE****: To prevent shut downs during operations due to speed (rpm) spikes (from the engine or motor), if a value sent as speed (CurrSpeed) is less than minimum allowed then the minimum allowed speed is used instead, and no errors are generated. Also, if value sent as speed (CurrSpeed) is greater than the maximum allowed then the maximum allowed speed is used instead and no again errors are generated.*

*e*RCM™ Express Quick WRITE List

Most Commonly Used Items

AB Tag	Register	Units	Operating Point inputs	Common Tag Name
ACI_Inputs[0]	40001	#	**Unit's Current Load Step**	**CurrLS**
ACI_Inputs[1]	40003	psiG	**Suction Pressure into Stage-1**	**Ps_psiG**
ACI_Inputs[2]	40005	psiG	**Discharge Pressure out of Last Stage**	**Pd_psiG**
ACI_Inputs[3]	40007	rpm	**Operating Speed**	**CurrSpeed**
ACI_Inputs[4]	40009	°F	**1st Stage Suction Gas Temperature**	**Ts1_F**
ACI_Inputs[5]	40011	°F	**2nd Stage Suction Gas Temperature**	**Ts2_F**
ACI_Inputs[6]	40013	°F	**3rd Stage Suction Gas Temperature**	**Ts3_F**
ACI_Inputs[7]	40015	°F	**4th Stage Suction Gas Temperature**	**Ts4_F**
ACI_Inputs[37]	40075	#	**Set to "1" to force eRCM Express Recalculation**	**ForceERCMExpressCalculations**
ACI_Inputs[54]	40109	%	Set Current Max Allowed Torque, **usually set to 100**(%). If set once after the unit model loaded, this does NOT need to be sent thereafter.	**TorqSP**

Less Commonly Used Items

AB Tag	Register	Units	Operating Point inputs	Common Tag Name
ACI_Inputs[50]	40101	#	**Set which eRCM Viewer Model to use (1/2/3/4/5/6/7/8/9)**	**SetViewerFile**
ACI_Inputs[51]	40103	HP	Set Driver's Max Power, only use if dynamically changes Set to 0 to return to model's original Max Rated Power	**SetDriverMaxHP**
ACI_Inputs[52]	40105	#	**Set to Load Step Selection Mode Number**	**LSSelectionMode**
ACI_Inputs[53]	40107	#	0 = Normal Operations; 1 = Request Safe Start Up Load Step	**CheckSafeStartUp**
ACI_Inputs[55]	40111	0	**Always set to 0**	**LSMode**
ACI_Inputs[56]	40113	°F	Set to current Ambient Temperature **only** if engine ambient rating used in model.	**TambF**
ACI_Inputs[57]	40115	0/1	**0 = Normal Mode;** 1 = Allow Load Steps with non-critical issues	**IgnoreNonCriticalErrors**
ACI_Inputs[58]	40117	%	Set Min % of Rated Power (or percent of current max flow) to consider when determining Next Step Up and Next Step Down	**MinLoadFlowPercChange**
ACI_Inputs[59]	40119	%	Set Max % of Available Power to consider when determining Next Step Up and Next Step Down	**MaxLoadPercChange**
ACI_Inputs[136]	40273	#	**Set to "1" to force eRCM Express Recalculation (same effect as setting 40075)**	**ForceERCMExpressCalculations**

*e*RCM™ Express Quick READ List

AB Tag	Register	Units	Description/Use	Common Tag Name
ACI_Control[0]	40275	#	Returns a one (1) that indicates if the eRCM Express computational engine is currently busy doing calculations. A zero (0) indicates that it is safe to poll for information as calculations are complete.	KrnlBusy
ACI_Control[1]	40277	#	Returns a unique number (from 0 to 6 million) that changes every time new operating conditions are processed by eRCM Express. Helps PLC determine if the full loop of communication/calculations is being implemented.	WDPulse
ACI_Control[2]	40279	#	Relative to Current Operating Conditions and Current Load Step, this register contains the Next Safe Up (increased load). If there is no safe load step with higher load, then a -1 is returned.	NextLSUp
ACI_Control[3]	40281	#	Relative to Current Operating Conditions and Current Load Step, this register contains the Next Safe Down (decreased load). If there is no safe load step with lower load, then a -1 is returned.	NextLSDown
ACI_Control[4]	40283	%	Returns the percent of load change from Current Load Step to Next Step Up, relative to maximum allowed load at the current speed. If there is no safe load step up, then this function returns 999999 as the percent change.	NSU_Perc
ACI_Control[5]	40285	%	Returns the percent of load change from Current Load Step to Next Step Down, relative to maximum allowed load at the current speed. If there is no safe load step down, then this function returns 999999 as the percent change.	NSD_Perc
ACI_Control[6]	40287	Rpm	For the Current Load Step, returns the lowest speed (rpm) for which unit can be adjusted without causing safety issues.	MinRPM
ACI_Control[7]	40289	Rpm	For the Current Load Step, returns the highest speed (rpm) for which unit can be adjusted without causing safety issues.	MaxRPM
ACI_Control[8]	40291	psiG	For the Current Load Step, returns the lowest suction pressure (psiG) for which unit can be adjusted without causing safety issues.	MinPs1
ACI_Control[9]	40293	psiG	For the Current Load Step, returns the highest suction pressure (psiG) for which unit can be adjusted without causing safety issues.	MaxPs1
ACI_Control[10]	40295	#	Returns the ideal Load Step to which to load, to fully load the unit. This is for reference only as the unit's/PLC's goals may not always be to maximize unit's load. Additionally, this register returns a value of -1 if all load steps are invalid.	OptimalLS
ACI_Control[11]	40297	%	For the Current Load Step, returns the Current Torque (%), which is the percent of Required Power relative to Driver's Derated Power.	CurrTorq
ACI_Control[12]	40299	%	Returns unit's efficiency (measure of how efficient the compressor is at current operating point).	IsenEff
ACI_Control[13]	40301	scf/hr	Returns fuel rate based on current load and current speed, only if modeled in the eRCM Viewer model. Otherwise, returns 0.	FuelRate

***NOTE*: If NextLSUp = -1 and NextLSDown = -1 and OptimalLS = -1, then there are NO SAFE LOAD STEPS, and hence unit should Shut Down.**

***NOTE*: If NextLSUp = -1 and NextLSDown = -1 and ErrorArray(CurrLS) $\neq$ 0, then the current load step is unsafe, and there are no safe loads obtainable (too far away), and hence unit should Shut Down.**

AB Tag	Register	Units	Description/Use	Common Tag Name
ACI_Misc[0]	40325	%	Returns Minimum Percent of Load (or flow depending on Load Step Selection Mode) to consider when looking for Next Step Up or Down.	MinLoadFlowPercChange
ACI_Misc[1]	40327	%	Returns the Maximum Percent of Load Change allowed when determining Next Step Up or Down.	MaxLoadPercChange
ACI_Misc[2]	40329	#	**Use CheckSafeStartup to force a calculation of this item. If value= -1, then there is no safe load step for which to start unit.**	LStoSetforSU
ACI_Misc[3]	40331	HP	Returns the Auxiliary Load plus any Frame Friction Load.	AuxLoad
ACI_Misc[4]	40333	HP	Returns the driver's maximum Power.	BHPMax
ACI_Misc[5]	40335	HP	Returns current Max Allowed Load (HP) based on current torque setting, rated load, current speed and T-Amb (if engine is ambient rated).	BHPMax at TorqSP
ACI_Misc[6]	40337	ft	Returns the elevation used for compressor modeling.	Elevation
ACI_Misc[7]	40339	psiA	Returns Atmospheric Pressure.	AtmPress
ACI_Misc[8]	40341	#	Returns number of cylinders modeled in the current model.	NumCyls
ACI_Misc[9]	40343	#	Returns number of load steps defined in the model.	NumLSs
ACI_Misc[10]	40345	#	Returns number of compression stages used in the current model.	NumStgs
ACI_Misc[11]	40347	#	Returns number of throws modeled in the current model.	NumThrws
ACI_Misc[12]	40349	#	Returns ID Number indicating the OEM of the frame/unit being modeled. (See eRCM Software for list of OEM IDs.)	OEM ID
ACI_Misc[13]	40351	#	Returns mechanical efficiency used to convert indicated horsepower to brake horsepower.	MechEff
ACI_Misc[14]	40353	DegF	Returns the average Maximum Discharge Temperature allowed for all modeled cylinders on the unit.	MaxDischF
ACI_Misc[15]	40355	%	Returns Relative Percent Humidity of Initial Inlet Gas being modeled, from 0 to 100 (%).	RelHumid

AB Tag	Register	Units	Description/Use	Common Tag Name
ACI_ErrorArray [0..99]	40445… 40643	#, INT	**Only read as many as needed.** Then look at bits of 32-bit Integers for specific errors encountered per load step. Any item with a non-zero value indicates that that load step is unsafe to use!	ErrorArray()

AB Tag	Register	Units	Description/Use	Common Tag Name
ACI_LoadArray [0..99]	40645… 40843	HP	**Only read as many as needed.** Then look at values of 32-bit Floats to see if any are too high (overloaded) or too low (under loaded) for driver. If any, drop them. Overloading and underloading are driver (engine/motor) related issues rather than compressor-related issues. Thus, if limiting load more than via the Max and Min allowed torque settings allow, then that will need to be handled via the PLC.	LoadArray()

AB Tag	Register	Units	Description/Use	Common Tag Name
ACI_FlowArray [0..99]	40845… 41043	MMscfd	**Only read as many as needed.**	FlowArray()

AB Tag	Register	Units	Description/Use	Common Tag Name
ACI_StageArray[7]	**41059**	**#**	Stage Compression Ratio	**StageArray()**
ACI_StageArray[10]	**41065**	**HP**	Stage Load	
ACI_StageArray[11]	**41067**	**MMscfd**	Stage Flow	
ACI_StageArray[22]	**41089**	**DegR**	Cylinder Discharge Temperature (°R, not °F)	

Add 60 to Modbus registers for each additional stage: **Add 30 to AB Tag** indexes for each additional stage.

AB Tag	Register	Units	Description/Use	Common Tag Name
ACI_ThrowArray[0]	**41405**	**%**	Gas Rod Loads – Compression (Most Low-speed units)	**ThrowArray()**
ACI_ThrowArray[1]	**41407**	**%**	Gas Rod Loads – Tension (Most Low-speed units)	
ACI_ThrowArray[4]	**41413**	**%**	Net Rod Loads – Compression (High-speed units)	
ACI_ThrowArray[5]	**41415**	**%**	Net Rod Loads – Tension (High-speed units)	
ACI_ThrowArray[19]	**41443**	**#**	Degrees of Pin Reversal in Compression	
ACI_ThrowArray[20]	**41445**	**#**	Degrees of Pin Reversal in Tension	

Add 60 to Modbus registers for each additional stage: **Add 30 to AB Tag** indexes for each additional stage.

AB Tag	Register	Units	Description/Use	Common Tag Name
ACI_CylHEArray[3]	**42011**	**DegF**	Cylinder End Estimated Discharge Temperature	**CylHEArray()**
ACI_CylHEArray[6]	**42017**	**%**	Suction Volumetric Efficiency	**CylCEArray() – For Modbus add 800 to register numbers.**
ACI_CylHEArray[14]	**42033**	**HP**	Cylinder End Load	**For AB, use ACI_CylCEArray[]**

Add 80 to Modbus registers for each additional stage: **Add 40 to AB Tag** indexes for each additional stage.

AB Tag	Register	Units	Description/Use	Common Tag Name
ACI_Ranges[0]	**43605**	**psiG**	Min Allowed Suction Pressure	PsRange
ACI_Ranges[1]	**43607**	**psiG**	Max Allowed Suction Pressure	
ACI_Ranges[2]	**43609**	**psiG**	Min Allowed Discharge Pressure	PdRange
ACI_Ranges[3]	**43611**	**psiG**	Max Allowed Discharge Pressure	
ACI_Ranges[4]	**43613**	**rpm**	Min Allowed Speed	SpeedRange
ACI_Ranges[5]	**43615**	**rpm**	Max Allowed Speed	
ACI_Ranges[6]	**43617**	**%**	Min Allowed Torque	TorqueRange
ACI_Ranges[7]	**43619**	**%**	Max Allowed Torque	
ACI_Ranges[8]	**43621**	**DegF**	Min Allowed Ambient Temperature	AmbientTempRange
ACI_Ranges[9]	**43623**	**DegF**	Max Allowed Ambient Temperature	
ACI_Ranges[10]	**43625**	**DegF**	Min Allowed Stage-1 Gas Temperature	Stage1TempRange
ACI_Ranges[11]	**43627**	**DegF**	Max Allowed Stage-1 Gas Temperature	
ACI_Ranges[12]	**43629**	**DegF**	Min Allowed Stage-2 Gas Temperature	Stage2TempRange
ACI_Ranges[13]	**43631**	**DegF**	Max Allowed Stage-2 Gas Temperature	
ACI_Ranges[14]	**43633**	**DegF**	Min Allowed Stage-3 Gas Temperature	Stage3TempRange
ACI_Ranges[15]	**43635**	**DegF**	Max Allowed Stage-3 Gas Temperature	
ACI_Ranges[16]	**43637**	**DegF**	Min Allowed Stage-4 Gas Temperature	Stage4TempRange
ACI_Ranges[17]	**43639**	**DegF**	Max Allowed Stage-4 Gas Temperature	
ACI_Ranges[18]	**43641**	**DegF**	Min Allowed Stage-5 Gas Temperature	Stage5TempRange
ACI_Ranges[19]	**43643**	**DegF**	Max Allowed Stage-5 Gas Temperature	
ACI_Ranges[20]	**43645**	**DegF**	Min Allowed Stage-6 Gas Temperature	Stage6TempRange
ACI_Ranges[21]	**43647**	**DegF**	Max Allowed Stage-6 Gas Temperature	

eRCM Express™

Allen-Bradley; Modbus Registers

Full Write/Read List

List of All Active Writes:

NAME	ADDRESS	AB Tag Name	Notes
CurrLS	40001	ACI_Inputs[0]	These are the basic values the control panel needs to send
PsG	40003	ACI_Inputs[1]	to eRCM Express.
PdG	40005	ACI_Inputs[2]	
CurrSpeed	40007	ACI_Inputs[3]	
Ts1F	40009	ACI_Inputs[4]	
Ts2F	40011	ACI_Inputs[5]	
Ts3F	40013	ACI_Inputs[6]	
Ts4F	40015	ACI_Inputs[7]	
Ts5F	40017	ACI_Inputs[8]	
Ts6F	40019	ACI_Inputs[9]	
Cyl1_Pd	40021	ACI_Inputs[10]	These items are only used when doing Condition
Cyl1_Td	40023	ACI_Inputs[11]	Monitoring.
Cyl2_Pd	40025	ACI_Inputs[12]	
Cyl2_Td	40027	ACI_Inputs[13]	
Cyl3_Pd	40029	ACI_Inputs[14]	
Cyl3_Td	40031	ACI_Inputs[15]	
Cyl4_Pd	40033	ACI_Inputs[16]	
Cyl4_Td	40035	ACI_Inputs[17]	
Cyl5_Pd	40037	ACI_Inputs[18]	
Cyl5_Td	40039	ACI_Inputs[19]	
Cyl6_Pd	40041	ACI_Inputs[20]	
Cyl6_Td	40043	ACI_Inputs[21]	
Cyl7_Pd	40045	ACI_Inputs[22]	
Cyl7_Td	40047	ACI_Inputs[23]	
Cyl8_Pd	40049	ACI_Inputs[24]	
Cyl8_Td	40051	ACI_Inputs[25]	
Cyl9_Pd	40053	ACI_Inputs[26]	
Cyl9_Td	40055	ACI_Inputs[27]	
Cyl10_Pd	40057	ACI_Inputs[28]	
Cyl10_Td	40059	ACI_Inputs[29]	
Stage 1 Ps	40061	ACI_Inputs[30]	For Special 2 Stage mode feature (fixed interstage pressure),
Stage 1 Pd	40063	ACI_Inputs[31]	write the interstage pressure to 40063.
Stage 2 Pd	40065	ACI_Inputs[32]	
Stage 3 Pd	40067	ACI_Inputs[33]	
Stage 4 Pd	40069	ACI_Inputs[34]	
Stage 5 Pd	40071	ACI_Inputs[35]	
Stage 6 Pd	40073	ACI_Inputs[36]	
ForceERCMExpressCalculations	40075	ACI_Inputs[37]	Set this to a "1" to force recalculations of all perf. Items.
Field_Ps	40077	ACI_Inputs[38]	If non-zero, calls routine to determine max allowed useful Ps.
SetViewerFile	40101	ACI_Inputs[50]	Not needed if only **one** modeling file being used.
SetDriverMaxHP	40103	ACI_Inputs[51]	
LSSelectionMode	40105	ACI_Inputs[52]	Load Step Selection Mode. Generally set in eRCM Viewer model.
CheckSafeStartUp	40107	ACI_Inputs[53]	Check SafeLoadStepStartup for the load step to use.
TorqSP	40109	ACI_Inputs[54]	Cannot be set higher than upper limit defined in model.
LSMode	40111	ACI_Inputs[55]	
TambF	40113	ACI_Inputs[56]	Used only for models derating engine on Ambient Temps.
IgnoreNonCriticalErrors	40115	ACI_Inputs[57]	Used during starting or stopping. Limit use to < 240 secs.
SetMinPercChange	40117	ACI_Inputs[58]	Values from 0.5 to 10 (%) allowed.
SetMaxPercChange	40119	ACI_Inputs[59]	Values from 2 to 80 (%) allowed.
ForceCondMonitoringCalcs	40121	ACI_Inputs[60]	Reset back to 0 after desired calculations are complete and read.
ForceLivePressures	40123	ACI_Inputs[61]	Reset back to 0 after desired calculations are complete and read.
EnableSafeOfflinePressRangesCalcs	40125	ACI_Inputs[62]	Disabled by default for now. Enable if not causing any CPU time issues.
SS12	40141	ACI_Inputs[70]	These items are only used when Side Streams are present.
SS23	40143	ACI_Inputs[71]	
SS34	40145	ACI_Inputs[72]	For Special 2 stage mode feature (dynamic side stream OUT),
SS45	40147	ACI_Inputs[73]	Use Register 40141.
SS56	40149	ACI_Inputs[74]	
SSDegFin12	40151	ACI_Inputs[75]	Use a negative number for side streams OUT, and a positive
SSDegFin23	40153	ACI_Inputs[76]	number for side streams IN, between stages.
SSDegFin34	40155	ACI_Inputs[77]	
SSDegFin45	40157	ACI_Inputs[78]	Set the temperature of the side stream IN between
SSDegFin56	40159	ACI_Inputs[79]	the indicated stages.
ForceERCMExpressCalculations	40273	ACI_Inputs[136]	Set this to a "1" to force recalculations of all perf. Items.

In Special 2-Stage mode, use Stage1 Pd for setting interstage pressure.

List of All Active Reads:

COMMON CONTROL ITEMS:

NAME	ADDRESS	Long Name	AB Tag Name	Description
KrnlBsy	40275	IsKernelBusy	ACI_Control[0]	Returns a Boolean that indicates if the eRCM Express computational engine is currently busy doing calculations. A True value (non-zero) means that polling the eRCM Kernel control for results may return old and incorrect results. A False value (zero) indicates that it is safe to poll for information as calculations are
WDPulse	40277	eRCMExpressWatchdogPulse	ACI_Control[1]	Returns a unique number (from 0 to 6 million) that changes every time new operating conditions are processed by eRCM Express. Helps PLC determine if the full loop of communication/calculations is being implemented.
NxtLSUP	40279	NextLoadStepUp	ACI_Control[2]	Relative to Current Operating Conditions and Current Load Step, this register contains the Next Safe Up (increased load). If there is no safe load step with higher load, then a -1 is returned.
NxtLSDN	40281	NextLoadStepDown	ACI_Control[3]	Relative to Current Operating Conditions and Current Load Step, this register contains the Next Safe Down (decreased load). If there is no safe load step with lower load, then a -1 is returned.
NSU_Perc	40283	NextLSUpPercentChange	ACI_Control[4]	Returns the percent of load change from Current Load Step to Next Step Up, relative to maximum allowed load at the current speed. If there is no safe load step up, then this function returns 999999 as the percent change.
NSD_Perc	40285	NextLSDownPercentChange	ACI_Control[5]	Returns the percent of load change from Current Load Step to Next Step Down, relative to maximum allowed load at the current speed. If there is no safe load step down, then this function returns 999999 as the percent change.
MinRPM	40287	MinSpeedCurrentLS	ACI_Control[6]	For the Current Load Step, returns the lowest speed (rpm) for which unit can be adjusted without causing safety issues.
MaxRPM	40289	MaxSpeedCurrentLS	ACI_Control[7]	For the Current Load Step, returns the highest speed (rpm) for which unit can be adjusted without causing safety issues.
MinPs1	40291	MinSuctPressureCurrentLS	ACI_Control[8]	For the Current Load Step, returns the lowest suction pressure (psiG) for which unit can be adjusted without causing safety issues.
MaxPs1	40293	MaxSuctPressureCurrentLS	ACI_Control[9]	For the Current Load Step, returns the highest suction pressure (psiG) for which unit can be adjusted without causing safety issues.
OptmLS	40295	FindOptimalLoadStep	ACI_Control[10]	Returns the ideal Load Step to which to load, to fully load the unit. This is for reference only as the unit's/PLC's goals may not always be to maximize unit's load. **Additionally, this registers returns a value of -1 if all load step are invalid.**
CurrTorq	40297	CurrentTorque	ACI_Control[11]	For the Current Load Step, returns the Current Torque (%).
IsenEff	40299	IsentropicEfficiency	ACI_Control[12]	Returns unit's efficiency (measure of how efficient the compressor is at current operating point).
FuelRate	40301	FuelRate	ACI_Control[13]	Returns fuel rate (scf/hr) based on current load and current speed.
GetTransVVCP	40303	GetTransVVCP	ACI_Control[14]	Returns the transitional load step to go to when in special single VVCP mode.
GetTrans	40305	GetTrans	ACI_Control[15]	Placeholder for potential helper item for GetTransVVCP
MaxHPLS	40307	MaxLoadLS	ACI_Control[16]	Returns LS# of load step with Maximum Load (HP). If none, then -1.
MinHPLS	40309	MinLoadLS	ACI_Control[17]	Returns LS# of load step with Minimum Load (HP). If none, then -1.
MaxCAPLS	40311	MaxFlowLS	ACI_Control[18]	Returns LS# of load step with Maximum Flow (MMscfd). If none, then -1.
MinCAPLS	40313	MinFlowLS	ACI_Control[19]	Returns LS# of load step with Minimum Flow (MMscfd). If none, then -1.
PsT_LS	40315		ACI_Control[20]	Load Step with max flow when Ps unthrottled as much as possible.
PsT_Ps	40317		ACI_Control[21]	Suction Pressure with max flow when Ps unthrottled as much as possible.
PsT_Flow	40319		ACI_Control[22]	Flow when Ps unthrottled as much as possible.
PsT_Power	40321		ACI_Control[23]	Power when Ps unthrottled as much as possible.

GENERAL ITEMS: Useful items related to compressor's configuration, Start Up, and such.

NAME	ADDRESS	AB Tag Name	Description
GetMinPercChange	40325	ACI_Misc[0]	Returns Minimum Percent of Load (or flow if LSSonFlow is one of the flow methods) to consider when looking for Next Step Up or Down.
GetMaxPercChange	40327	ACI_Misc[1]	Returns the Maximum Percent of **load** to consider when looking for Next Step Up or Down.
SafeLoadStepStartup	40329	ACI_Misc[2]	Use CheckSafeStartup to force a calculation of this item. If value= -1, then there are no safe load steps for which to start the unit.
AuxLoad	40331	ACI_Misc[3]	Returns the Auxiliary Load plus any Frame Friction Load (HP).
BHPMax	40333	ACI_Misc[4]	Returns the driver's maximum load (HP).
BHPMax at TorqSP	40335	ACI_Misc[5]	Returns current Max Allowed Load (HP) based on current torque setting, rated load, current speed and T-Amb (if emissions engaged).
Elevation	40337	ACI_Misc[6]	Returns the elevation (ft) used for compressor modeling.
AtmPress	40339	ACI_Misc[7]	Returns Atmospheric Pressure (psiA).
NumCyls	40341	ACI_Misc[8]	Returns number of cylinders modeled in the current model.
NumLSs	40343	ACI_Misc[9]	Returns number of load step defined in the mode.
NumStgs	40345	ACI_Misc[10]	Returns number of compression stages used in the current model.
NumThrws	40347	ACI_Misc[11]	Returns number of throws modeled in the current model.
OEM ID	40349	ACI_Misc[12]	Returns ID Number indicating the OEM of the frame/unit being modeled. (See eRCM Software for list of OEM IDs.)
MechEff	40351	ACI_Misc[13]	Returns mechanical efficiency used to convert indicated horsepower to brake
MaxDischF	40353	ACI_Misc[14]	Returns the average Maximum Discharge Temperature (degF) allowed for all modeled cylinders on the unit.
RelHumid	40355	ACI_Misc[15]	Returns Relative Percent Humidity of Initial Inlet Gas being modeled, from 0 to 100 (%).
eRCMExpressVER	40357	ACI_Misc[16]	0=Initial Release, otherwise a number such that its digits are YYYYMMDDR. 201811243 means 2018-Nov-24 Rev3.

OFFLINE VALIDATION LIMITS: These are useful if communications with eRCM Express are lost.

OfflineMinSafePs1	40359	ACI_Misc[17]	If PLC losses communications with eRCM Express, then if PLC keeps compressor at current Load Step and at Current Speed, and if the Ts's do not
OfflineMaxSafePs1	40361	ACI_Misc[18]	vary by more than +/-10 degF, then unit will be safe as long as current 1st stage suction
OfflineMinSafePd1	40363	ACI_Misc[19]	pressure (Ps) and last stage discharge pressure (Pd) meet the following criteria:
OfflineMaxSafePd1	40365	ACI_Misc[20]	Unit is Safe If and Only If: **((OfflineMinSafePs1 <= Ps <= OfflineMaxSafePs1) AND (OfflineMinSafePd1 <= Pd <= OfflineMaxSafePd1)) OR**
OfflineMinSafePs2	40367	ACI_Misc[21]	**((OfflineMinSafePs2 <= Ps <= OfflineMaxSafePs2) AND (OfflineMinSafePd2 <= Pd <= OfflineMaxSafePd2))**
OfflineMaxSafePs2	40369	ACI_Misc[22]	*AND PLC does not change Load Step, and PLC does not change speed, and Ts's do not vary more than +/-10 degF.*
OfflineMinSafePd2	40371	ACI_Misc[23]	
OfflineMaxSafePd2	40373	ACI_Misc[24]	

USER MODEL ID NUMBER: Allows PLC to better verify eRCM model loaded is the model of interest.

UserModelID	40375	ACI_Misc[25]	A number from 1 to 2,000,000 (inclusive) set by user in eRCM to uniquely identify this file to the PLC. If 0, then ID # never set.

CONDITION MONITORING ITEMS: These are used to help diagnose issues with valves and rings.

NAME	ADDRESS	AB Tag Name	Description
UnitSlip	40385	ACI_Cond[0]	Effect of all valves (all stages and all cylinders) on the unit's flow as a whole.
CylSlip.C01	40387	ACI_Cond[1]	Effect of valve and ring leakage for each particular cylinder on that cylinder's effective flow rate.
CylSlip.C02	40389	ACI_Cond[2]	
CylSlip.C03	40391	ACI_Cond[3]	
CylSlip.C04	40393	ACI_Cond[4]	
CylSlip.C05	40395	ACI_Cond[5]	Customer can develop their own limits for when results trigger need to review cylinder's valves and rings, but:
CylSlip.C06	40397	ACI_Cond[6]	Results < -0.08 are very bad — Results > 0.08 are very bad
CylSlip.C07	40399	ACI_Cond[7]	Results between -0.08 and -0.04 are bad — Results between 0.04 and 0.08 are bad
CylSlip.C08	40401	ACI_Cond[8]	Results between -0.04 and -0.02 are of some concern — Results between 0.02 and -0.04 are of some concern
CylSlip.C09	40403	ACI_Cond[9]	**Results between -0.02 and +0.02 are normal**
CylSlip.C10	40405	ACI_Cond[10]	
CylTdF.C1	40407	ACI_Cond[11]	Based on supplied/measured pressures, theoretically, these are what the discharge temperatures per cylinder should be.
CylTdF.C2	40409	ACI_Cond[12]	
CylTdF.C3	40411	ACI_Cond[13]	
CylTdF.C4	40413	ACI_Cond[14]	
CylTdF.C5	40415	ACI_Cond[15]	
CylTdF.C6	40417	ACI_Cond[16]	
CylTdF.C7	40419	ACI_Cond[17]	
CylTdF.C8	40421	ACI_Cond[18]	
CylTdF.C9	40423	ACI_Cond[19]	
CylTdF.C10	40425	ACI_Cond[20]	

ERROR ARRAY:

AB Tag Name

NAME	ADDRESS		NAME	ADDRESS		NAME	ADDRESS		NAME	ADDRESS	ACI_ErrorArray[]
Err.LS.01	40445	248	Err.LS.26	40495	273	Err.LS.51	40545	298	Err.LS.76	40595	ACI_ErrorArray[0, 25, 50, 75]
Err.LS.02	40447	249	Err.LS.27	40497	274	Err.LS.52	40547	299	Err.LS.77	40597	ACI_ErrorArray[1, 26, 51, 76]
Err.LS.03	40449	250	Err.LS.28	40499	275	Err.LS.53	40549	300	Err.LS.78	40599	ACI_ErrorArray[2, 27, 52, 77]
Err.LS.04	40451	251	Err.LS.29	40501	276	Err.LS.54	40551	301	Err.LS.79	40601	ACI_ErrorArray[3, 28, 53, 78]
Err.LS.05	40453	252	Err.LS.30	40503	277	Err.LS.55	40553	302	Err.LS.80	40603	ACI_ErrorArray[4, 29, 54, 79]
Err.LS.06	40455	253	Err.LS.31	40505	278	Err.LS.56	40555	303	Err.LS.81	40605	ACI_ErrorArray[5, 30, 55, 80]
Err.LS.07	40457	254	Err.LS.32	40507	279	Err.LS.57	40557	304	Err.LS.82	40607	ACI_ErrorArray[6, 31, 56, 81]
Err.LS.08	40459	255	Err.LS.33	40509	280	Err.LS.58	40559	305	Err.LS.83	40609	ACI_ErrorArray[7, 32, 57, 82]
Err.LS.09	40461	256	Err.LS.34	40511	281	Err.LS.59	40561	306	Err.LS.84	40611	ACI_ErrorArray[8, 33, 58, 83]
Err.LS.10	40463	257	Err.LS.35	40513	282	Err.LS.60	40563	307	Err.LS.85	40613	ACI_ErrorArray[9, 34, 59, 84]
Err.LS.11	40465	258	Err.LS.36	40515	283	Err.LS.61	40565	308	Err.LS.86	40615	ACI_ErrorArray[10, 35, 60, 85]
Err.LS.12	40467	259	Err.LS.37	40517	284	Err.LS.62	40567	309	Err.LS.87	40617	ACI_ErrorArray[11, 36, 61, 86]
Err.LS.13	40469	260	Err.LS.38	40519	285	Err.LS.63	40569	310	Err.LS.88	40619	ACI_ErrorArray[12, 37, 62, 87]
Err.LS.14	40471	261	Err.LS.39	40521	286	Err.LS.64	40571	311	Err.LS.89	40621	ACI_ErrorArray[13, 38, 63, 88]
Err.LS.15	40473	262	Err.LS.40	40523	287	Err.LS.65	40573	312	Err.LS.90	40623	ACI_ErrorArray[14, 39, 64, 89]
Err.LS.16	40475	263	Err.LS.41	40525	288	Err.LS.66	40575	313	Err.LS.91	40625	ACI_ErrorArray[15, 40, 65, 90]
Err.LS.17	40477	264	Err.LS.42	40527	289	Err.LS.67	40577	314	Err.LS.92	40627	ACI_ErrorArray[16, 41, 66, 91]
Err.LS.18	40479	265	Err.LS.43	40529	290	Err.LS.68	40579	315	Err.LS.93	40629	ACI_ErrorArray[17, 42, 67, 92]
Err.LS.19	40481	266	Err.LS.44	40531	291	Err.LS.69	40581	316	Err.LS.94	40631	ACI_ErrorArray[18, 43, 68, 93]
Err.LS.20	40483	267	Err.LS.45	40533	292	Err.LS.70	40583	317	Err.LS.95	40633	ACI_ErrorArray[19, 44, 69, 94]
Err.LS.21	40485	268	Err.LS.46	40535	293	Err.LS.71	40585	318	Err.LS.96	40635	ACI_ErrorArray[20, 45, 70, 95]
Err.LS.22	40487	269	Err.LS.47	40537	294	Err.LS.72	40587	319	Err.LS.97	40637	ACI_ErrorArray[21, 46, 71, 96]
Err.LS.23	40489	270	Err.LS.48	40539	295	Err.LS.73	40589	320	Err.LS.98	40639	ACI_ErrorArray[22, 47, 72, 97]
Err.LS.24	40491	271	Err.LS.49	40541	296	Err.LS.74	40591	321	Err.LS.99	40641	ACI_ErrorArray[23, 48, 73, 98]
Err.LS.25	40493	272	Err.LS.50	40543	297	Err.LS.75	40593	322	Err.LS.100	40643	ACI_ErrorArray[24, 49, 74, 99]

Condition Monitoring Only License: All items retrieved will be 0's except for Current Load Step item.

Bit Directory:

Bit 00 / Discharge Pressure Exceeded	Bit 16 / Max Pressure Differential Exceeded
Bit 01 / Discharge Temperature Exceeded	Bit 17 / Estimated Cylinder Discharge Temperatures are High
Bit 02 / Gas Rod Loads Exceeded	Bit 18 / Low Compression Ratio
Bit 03 / Net Rod Loads Exceeded	Bit 19 / High Compression Ratio
Bit 04 / Low Volumetric Efficiency	Bit 20 / Ariel Gas Rod Load Exceeded
Bit 05 / Throttling Occurred	Bit 21 / BHP per Throw Limit Exceeded
Bit 06 / Z-Factor Out Of Range	*Bit 22 / Reserved*
Bit 07 / Invalid Suction Pressure	*Bit 23 / Reserved*
Bit 08 / Invalid Discharge Pressure	*Bit 24 / Reserved*
Bit 09 / RESERVED (should always be 0)	*Bit 25 / Reserved*
Bit 10 / Failed Rod Reversal	*Bit 26 / Reserved*
Bit 11 / Pin Reversal Exceeded	*Bit 27 / Reserved*
Bit 12 / API618 3%-Rule Violated	*Bit 28 / Reserved*
Bit 13 / Non-obtainable Goal	*Bit 29 / Reserved*
Bit 14 / Blank Off Occurred	*Bit 30 / Reserved*
Bit 15 / Liquids May Be Forming	*Bit 31 / Reserved*

LOAD ARRAY:

AB Tag Name

NAME	ADDRESS		NAME	ADDRESS		NAME	ADDRESS		NAME	ADDRESS	ACI_LoadArray[]
Load.LS.01	40645	348	Load.LS.26	40695	373	Load.LS.51	40745	398	Load.LS.76	40795	ACI_LoadArray[0, 25, 50, 75]
Load.LS.02	40647	349	Load.LS.27	40697	374	Load.LS.52	40747	399	Load.LS.77	40797	ACI_LoadArray[1, 26, 51, 76]
Load.LS.03	40649	350	Load.LS.28	40699	375	Load.LS.53	40749	400	Load.LS.78	40799	ACI_LoadArray[2, 27, 52, 77]
Load.LS.04	40651	351	Load.LS.29	40701	376	Load.LS.54	40751	401	Load.LS.79	40801	ACI_LoadArray[3, 28, 53, 78]
Load.LS.05	40653	352	Load.LS.30	40703	377	Load.LS.55	40753	402	Load.LS.80	40803	ACI_LoadArray[4, 29, 54, 79]
Load.LS.06	40655	353	Load.LS.31	40705	378	Load.LS.56	40755	403	Load.LS.81	40805	ACI_LoadArray[5, 30, 55, 80]
Load.LS.07	40657	354	Load.LS.32	40707	379	Load.LS.57	40757	404	Load.LS.82	40807	ACI_LoadArray[6, 31, 56, 81]
Load.LS.08	40659	355	Load.LS.33	40709	380	Load.LS.58	40759	405	Load.LS.83	40809	ACI_LoadArray[7, 32, 57, 82]
Load.LS.09	40661	356	Load.LS.34	40711	381	Load.LS.59	40761	406	Load.LS.84	40811	ACI_LoadArray[8, 33, 58, 83]
Load.LS.10	40663	357	Load.LS.35	40713	382	Load.LS.60	40763	407	Load.LS.85	40813	ACI_LoadArray[9, 34, 59, 84]
Load.LS.11	40665	358	Load.LS.36	40715	383	Load.LS.61	40765	408	Load.LS.86	40815	ACI_LoadArray[10, 35, 60, 85]
Load.LS.12	40667	359	Load.LS.37	40717	384	Load.LS.62	40767	409	Load.LS.87	40817	ACI_LoadArray[11, 36, 61, 86]
Load.LS.13	40669	360	Load.LS.38	40719	385	Load.LS.63	40769	410	Load.LS.88	40819	ACI_LoadArray[12, 37, 62, 87]
Load.LS.14	40671	361	Load.LS.39	40721	386	Load.LS.64	40771	411	Load.LS.89	40821	ACI_LoadArray[13, 38, 63, 88]
Load.LS.15	40673	362	Load.LS.40	40723	387	Load.LS.65	40773	412	Load.LS.90	40823	ACI_LoadArray[14, 39, 64, 89]
Load.LS.16	40675	363	Load.LS.41	40725	388	Load.LS.66	40775	413	Load.LS.91	40825	ACI_LoadArray[15, 40, 65, 90]
Load.LS.17	40677	364	Load.LS.42	40727	389	Load.LS.67	40777	414	Load.LS.92	40827	ACI_LoadArray[16, 41, 66, 91]
Load.LS.18	40679	365	Load.LS.43	40729	390	Load.LS.68	40779	415	Load.LS.93	40829	ACI_LoadArray[17, 42, 67, 92]
Load.LS.19	40681	366	Load.LS.44	40731	391	Load.LS.69	40781	416	Load.LS.94	40831	ACI_LoadArray[18, 43, 68, 93]
Load.LS.20	40683	367	Load.LS.45	40733	392	Load.LS.70	40783	417	Load.LS.95	40833	ACI_LoadArray[19, 44, 69, 94]
Load.LS.21	40685	368	Load.LS.46	40735	393	Load.LS.71	40785	418	Load.LS.96	40835	ACI_LoadArray[20, 45, 70, 95]
Load.LS.22	40687	369	Load.LS.47	40737	394	Load.LS.72	40787	419	Load.LS.97	40837	ACI_LoadArray[21, 46, 71, 96]
Load.LS.23	40689	370	Load.LS.48	40739	395	Load.LS.73	40789	420	Load.LS.98	40839	ACI_LoadArray[22, 47, 72, 97]
Load.LS.24	40691	371	Load.LS.49	40741	396	Load.LS.74	40791	421	Load.LS.99	40841	ACI_LoadArray[23, 48, 73, 98]
Load.LS.25	40693	372	Load.LS.50	40743	397	Load.LS.75	40793	422	oad.LS.100	40843	ACI_LoadArray[24, 49, 74, 99]

Condition Monitoring Only License: All items retrieved will be 0's except for Current Load Step item.

FLOW ARRAY:

AB Tag Name

NAME	ADDRESS		NAME	ADDRESS		NAME	ADDRESS		NAME	ADDRESS	ACI_FlowArray[]
Flow.LS.01	40845	448	Flow.LS.26	40895	473	Flow.LS.51	40945	498	Flow.LS.76	40995	ACI_FlowArray[0, 25, 50, 75]
Flow.LS.02	40847	449	Flow.LS.27	40897	474	Flow.LS.52	40947	499	Flow.LS.77	40997	ACI_FlowArray[1, 26, 51, 76]
Flow.LS.03	40849	450	Flow.LS.28	40899	475	Flow.LS.53	40949	500	Flow.LS.78	40999	ACI_FlowArray[2, 27, 52, 77]
Flow.LS.04	40851	451	Flow.LS.29	40901	476	Flow.LS.54	40951	501	Flow.LS.79	41001	ACI_FlowArray[3, 28, 53, 78]
Flow.LS.05	40853	452	Flow.LS.30	40903	477	Flow.LS.55	40953	502	Flow.LS.80	41003	ACI_FlowArray[4, 29, 54, 79]
Flow.LS.06	40855	453	Flow.LS.31	40905	478	Flow.LS.56	40955	503	Flow.LS.81	41005	ACI_FlowArray[5, 30, 55, 80]
Flow.LS.07	40857	454	Flow.LS.32	40907	479	Flow.LS.57	40957	504	Flow.LS.82	41007	ACI_FlowArray[6, 31, 56, 81]
Flow.LS.08	40859	455	Flow.LS.33	40909	480	Flow.LS.58	40959	505	Flow.LS.83	41009	ACI_FlowArray[7, 32, 57, 82]
Flow.LS.09	40861	456	Flow.LS.34	40911	481	Flow.LS.59	40961	506	Flow.LS.84	41011	ACI_FlowArray[8, 33, 58, 83]
Flow.LS.10	40863	457	Flow.LS.35	40913	482	Flow.LS.60	40963	507	Flow.LS.85	41013	ACI_FlowArray[9, 34, 59, 84]
Flow.LS.11	40865	458	Flow.LS.36	40915	483	Flow.LS.61	40965	508	Flow.LS.86	41015	ACI_FlowArray[10, 35, 60, 85]
Flow.LS.12	40867	459	Flow.LS.37	40917	484	Flow.LS.62	40967	509	Flow.LS.87	41017	ACI_FlowArray[11, 36, 61, 86]
Flow.LS.13	40869	460	Flow.LS.38	40919	485	Flow.LS.63	40969	510	Flow.LS.88	41019	ACI_FlowArray[12, 37, 62, 87]
Flow.LS.14	40871	461	Flow.LS.39	40921	486	Flow.LS.64	40971	511	Flow.LS.89	41021	ACI_FlowArray[13, 38, 63, 88]
Flow.LS.15	40873	462	Flow.LS.40	40923	487	Flow.LS.65	40973	512	Flow.LS.90	41023	ACI_FlowArray[14, 39, 64, 89]
Flow.LS.16	40875	463	Flow.LS.41	40925	488	Flow.LS.66	40975	513	Flow.LS.91	41025	ACI_FlowArray[15, 40, 65, 90]
Flow.LS.17	40877	464	Flow.LS.42	40927	489	Flow.LS.67	40977	514	Flow.LS.92	41027	ACI_FlowArray[16, 41, 66, 91]
Flow.LS.18	40879	465	Flow.LS.43	40929	490	Flow.LS.68	40979	515	Flow.LS.93	41029	ACI_FlowArray[17, 42, 67, 92]
Flow.LS.19	40881	466	Flow.LS.44	40931	491	Flow.LS.69	40981	516	Flow.LS.94	41031	ACI_FlowArray[18, 43, 68, 93]
Flow.LS.20	40883	467	Flow.LS.45	40933	492	Flow.LS.70	40983	517	Flow.LS.95	41033	ACI_FlowArray[19, 44, 69, 94]
Flow.LS.21	40885	468	Flow.LS.46	40935	493	Flow.LS.71	40985	518	Flow.LS.96	41035	ACI_FlowArray[20, 45, 70, 95]
Flow.LS.22	40887	469	Flow.LS.47	40937	494	Flow.LS.72	40987	519	Flow.LS.97	41037	ACI_FlowArray[21, 46, 71, 96]
Flow.LS.23	40889	470	Flow.LS.48	40939	495	Flow.LS.73	40989	520	Flow.LS.98	41039	ACI_FlowArray[22, 47, 72, 97]
Flow.LS.24	40891	471	Flow.LS.49	40941	496	Flow.LS.74	40991	521	Flow.LS.99	41041	ACI_FlowArray[23, 48, 73, 98]
Flow.LS.25	40893	472	Flow.LS.50	40943	497	Flow.LS.75	40993	522	low.LS.100	41043	ACI_FlowArray[24, 49, 74, 99]

Condition Monitoring Only License: All items retrieved will be 0's except for Current Load Step item.

STAGE ARRAY:

AB Tag Name

NAME	STAGE-1	STAGE-2	STAGE-3	STAGE-4	STAGE-5	STAGE-6	Stg1	Stg2	Stg3	Stg4	Stg5	Stg6	ACI_StageArray[]
Stage-Item.01	41045	41105	41165	41225	41285	41345	523	553	583	613	643	673	ACI_StageArray[0, 30, 60 ,90, 120, 150]
Stage-Item.02	41047	41107	41167	41227	41287	41347	524	554	584	614	644	674	ACI_StageArray[1, 31, 61 ,91, 121, 151]
Stage-Item.03	41049	41109	41169	41229	41289	41349	525	555	585	615	645	675	ACI_StageArray[2, 32, 62 ,92, 122, 152]
Stage-Item.04	41051	41111	41171	41231	41291	41351	526	556	586	616	646	676	ACI_StageArray[3, 33, 63 ,93, 123, 153]
Stage-Item.05	41053	41113	41173	41233	41293	41353	527	557	587	617	647	677	ACI_StageArray[4, 34, 64 ,94, 124, 154]
Stage-Item.06	41055	41115	41175	41235	41295	41355	528	558	588	618	648	678	ACI_StageArray[5, 35, 65 ,95, 125, 155]
Stage-Item.07	41057	41117	41177	41237	41297	41357	529	559	589	619	649	679	ACI_StageArray[6, 36, 66 ,96, 126, 156]
Stage-Item.08	41059	41119	41179	41239	41299	41359	530	560	590	620	650	680	ACI_StageArray[7, 37, 67 ,97, 127, 157]
Stage-Item.09	41061	41121	41181	41241	41301	41361	531	561	591	621	651	681	ACI_StageArray[8, 38, 68 ,98, 128, 158]
Stage-Item.10	41063	41123	41183	41243	41303	41363	532	562	592	622	652	682	ACI_StageArray[9, 39, 69 ,99, 129, 159]
Stage-Item.11	41065	41125	41185	41245	41305	41365	533	563	593	623	653	683	ACI_StageArray[10, 40, 70 ,100, 130, 160]
Stage-Item.12	41067	41127	41187	41247	41307	41367	534	564	594	624	654	684	ACI_StageArray[11, 41, 71 ,101, 131, 161]
Stage-Item.13	41069	41129	41189	41249	41309	41369	535	565	595	625	655	685	ACI_StageArray[12, 42, 72 ,102, 132, 162]
Stage-Item.14	41071	41131	41191	41251	41311	41371	536	566	596	626	656	686	ACI_StageArray[13, 43, 73 ,103, 133, 163]
Stage-Item.15	41073	41133	41193	41253	41313	41373	537	567	597	627	657	687	ACI_StageArray[14, 44, 74 ,104, 134, 164]
Stage-Item.16	41075	41135	41195	41255	41315	41375	538	568	598	628	658	688	ACI_StageArray[15, 45, 75 ,105, 135, 165]
Stage-Item.17	41077	41137	41197	41257	41317	41377	539	569	599	629	659	689	ACI_StageArray[16, 46, 76 ,106, 136, 166]
Stage-Item.18	41079	41139	41199	41259	41319	41379	540	570	600	630	660	690	ACI_StageArray[17, 47, 77 ,107, 137, 167]
Stage-Item.19	41081	41141	41201	41261	41321	41381	541	571	601	631	661	691	ACI_StageArray[18, 48, 78 ,108, 138, 168]
Stage-Item.20	41083	41143	41203	41263	41323	41383	542	572	602	632	662	692	ACI_StageArray[19, 49, 79 ,109, 139, 169]
Stage-Item.21	41085	41145	41205	41265	41325	41385	543	573	603	633	663	693	ACI_StageArray[20, 50, 80 ,110, 140, 170]
Stage-Item.22	41087	41147	41207	41267	41327	41387	544	574	604	634	664	694	ACI_StageArray[21, 51, 81 ,111, 141, 171]
Stage-Item.23	41089	41149	41209	41269	41329	41389	545	575	605	635	665	695	ACI_StageArray[22, 52, 82 ,112, 142, 172]
Stage-Item.24	41091	41151	41211	41271	41331	41391	546	576	606	636	666	696	ACI_StageArray[23, 53, 83 ,113, 143, 173]
Stage-Item.25	41093	41153	41213	41273	41333	41393	547	577	607	637	667	697	ACI_StageArray[24, 54, 84 ,114, 144, 174]
Stage-Item.26	41095	41155	41215	41275	41335	41395	548	578	608	638	668	698	ACI_StageArray[25, 55, 85 ,115, 145, 175]
Stage-Item.27	41097	41157	41217	41277	41337	41397	549	579	609	639	669	699	ACI_StageArray[26, 56, 86 ,116, 146, 176]
Stage-Item.28	41099	41159	41219	41279	41339	41399	550	580	610	640	670	700	ACI_StageArray[27, 57, 87 ,117, 147, 177]
Stage-Item.29	41101	41161	41221	41281	41341	41401	551	581	611	641	671	701	ACI_StageArray[28, 58, 88 ,118, 148, 178]
Stage-Item.30	41103	41163	41223	41283	41343	41403	552	582	612	642	672	702	ACI_StageArray[29, 59, 89 ,119, 149, 179]

Condition Monitoring Only License: All items retrieved will be 0's except for items #1-#4, 22 and 26

Items

Item 1: Suction Pressure at Gauge (psig)
Item 2: Discharge Pressure at Gauge (psig)
Item 3: Suction Temperature (°F)
Item 4: Adiabatic Discharge Temperature (°F)
Item 5: Z-Suction Compressibility Factor
Item 6: Z-Discharge Compressibility Factor
Item 7: Gas K-Value
Item 8: Compression Ratio
Item 9: Suction Pressure at Flange (psig)
Item 10: Discharge Pressure at Flange (psig)
Item 11: Load (BHP) per stage
Item 12: Flow (MMscfd) per stage
Item 13 (reserved)
Item 14: (reserved)
Item 15: Compressibility Factor at current Base Conditions
Item 16: Specific Gravity of gas being compressed via this stage
Item 17: Mole Weight of gas being compressed via this stage
Item 18: (reserved)
Item 19: (reserved)
Item 20: (reserved)
Item 21: Base Temperature for Base Conditions (°F)
Item 22: Discharge pressure reflecting pressure drops up to the cylinder flange (psiA)
Item 23: Discharge Temperature at cylinder flange (°R)
Item 24: (reserved)
Item 25: (reserved)
Item 26: Suction pressure reflecting pressure drops up to the cylinder flange (psiA)
Item 27: (reserved)
Item 28: (reserved)
Item 29: (reserved)
Item 30: (reserved)

THROW ARRAY:

AB Tag Name

NAME	Throw-1	Throw-2	Throw-3	Throw-4	Throw-5	Throw-6	Throw-7	Throw-8	Throw-9	Throw-10	T1	T2	T3	T4	T5	T6	T7	T8	T9	T10	ACI_ThrowArray[]
Throw-Item.01	41405	41465	41525	41585	41645	41705	41765	41825	41885	41945	703	733	763	793	823	853	883	913	943	973	ACI_ThrowArray[0, 30, 60, 90, 120, 150, 180, 210, 240, 270]
Throw-Item.02	41407	41467	41527	41587	41647	41707	41767	41827	41887	41947	704	734	764	794	824	854	884	914	944	974	ACI_ThrowArray[1, 31, 61, 91, 121, 151, 181, 211, 241, 271]
Throw-Item.03	41409	41469	41529	41589	41649	41709	41769	41829	41889	41949	705	735	765	795	825	855	885	915	945	975	ACI_ThrowArray[2, 32, 62, 92, 122, 152, 182, 212, 242, 272]
Throw-Item.04	41411	41471	41531	41591	41651	41711	41771	41831	41891	41951	706	736	766	796	826	856	886	916	946	976	ACI_ThrowArray[3, 33, 63, 93, 123, 153, 183, 213, 243, 273]
Throw-Item.05	41413	41473	41533	41593	41653	41713	41773	41833	41893	41953	707	737	767	797	827	857	887	917	947	977	ACI_ThrowArray[4, 34, 64, 94, 124, 154, 184, 214, 244, 274]
Throw-Item.06	41415	41475	41535	41595	41655	41715	41775	41835	41895	41955	708	738	768	798	828	858	888	918	948	978	ACI_ThrowArray[5, 35, 65, 95, 125, 155, 185, 215, 245, 275]
Throw-Item.07	41417	41477	41537	41597	41657	41717	41777	41837	41897	41957	709	739	769	799	829	859	889	919	949	979	ACI_ThrowArray[6, 36, 66, 96, 126, 156, 186, 216, 246, 276]
Throw-Item.08	41419	41479	41539	41599	41659	41719	41779	41839	41899	41959	710	740	770	800	830	860	890	920	950	980	ACI_ThrowArray[7, 37, 67, 97, 127, 157, 187, 217, 247, 277]
Throw-Item.09	41421	41481	41541	41601	41661	41721	41781	41841	41901	41961	711	741	771	801	831	861	891	921	951	981	ACI_ThrowArray[8, 38, 68, 98, 128, 158, 188, 218, 248, 278]
Throw-Item.10	41423	41483	41543	41603	41663	41723	41783	41843	41903	41963	712	742	772	802	832	862	892	922	952	982	ACI_ThrowArray[9, 39, 69, 99, 129, 159, 189, 219, 249, 279]
Throw-Item.11	41425	41485	41545	41605	41665	41725	41785	41845	41905	41965	713	743	773	803	833	863	893	923	953	983	ACI_ThrowArray[10, 40, 70, 100, 130, 160, 190, 220, 250, 280]
Throw-Item.12	41427	41487	41547	41607	41667	41727	41787	41847	41907	41967	714	744	774	804	834	864	894	924	954	984	ACI_ThrowArray[11, 41, 71, 101, 131, 161, 191, 221, 251, 281]
Throw-Item.13	41429	41489	41549	41609	41669	41729	41789	41849	41909	41969	715	745	775	805	835	865	895	925	955	985	ACI_ThrowArray[12, 42, 72, 102, 132, 162, 192, 222, 252, 282]
Throw-Item.14	41431	41491	41551	41611	41671	41731	41791	41851	41911	41971	716	746	776	806	836	866	896	926	956	986	ACI_ThrowArray[13, 43, 73, 103, 133, 163, 193, 223, 253, 283]
Throw-Item.15	41433	41493	41553	41613	41673	41733	41793	41853	41913	41973	717	747	777	807	837	867	897	927	957	987	ACI_ThrowArray[14, 44, 74, 104, 134, 164, 194, 224, 254, 284]
Throw-Item.16	41435	41495	41555	41615	41675	41735	41795	41855	41915	41975	718	748	778	808	838	868	898	928	958	988	ACI_ThrowArray[15, 45, 75, 105, 135, 165, 195, 225, 255, 285]
Throw-Item.17	41437	41497	41557	41617	41677	41737	41797	41857	41917	41977	719	749	779	809	839	869	899	929	959	989	ACI_ThrowArray[16, 46, 76, 106, 136, 166, 196, 226, 256, 286]
Throw-Item.18	41439	41499	41559	41619	41679	41739	41799	41859	41919	41979	720	750	780	810	840	870	900	930	960	990	ACI_ThrowArray[17, 47, 77, 107, 137, 167, 197, 227, 257, 287]
Throw-Item.19	41441	41501	41561	41621	41681	41741	41801	41861	41921	41981	721	751	781	811	841	871	901	931	961	991	ACI_ThrowArray[18, 48, 78, 108, 138, 168, 198, 228, 258, 288]
Throw-Item.20	41443	41503	41563	41623	41683	41743	41803	41863	41923	41983	722	752	782	812	842	872	902	932	962	992	ACI_ThrowArray[19, 49, 79, 109, 139, 169, 199, 229, 259, 289]
Throw-Item.21	41445	41505	41565	41625	41685	41745	41805	41865	41925	41985	723	753	783	813	843	873	903	933	963	993	ACI_ThrowArray[20, 50, 80, 110, 140, 170, 200, 230, 260, 290]
Throw-Item.22	41447	41507	41567	41627	41687	41747	41807	41867	41927	41987	724	754	784	814	844	874	904	934	964	994	ACI_ThrowArray[21, 51, 81, 111, 141, 171, 201, 231, 261, 291]
Throw-Item.23	41449	41509	41569	41629	41689	41749	41809	41869	41929	41989	725	755	785	815	845	875	905	935	965	995	ACI_ThrowArray[22, 52, 82, 112, 142, 172, 202, 232, 262, 292]
Throw-Item.24	41451	41511	41571	41631	41691	41751	41811	41871	41931	41991	726	756	786	816	846	876	906	936	966	996	ACI_ThrowArray[23, 53, 83, 113, 143, 173, 203, 233, 263, 293]
Throw-Item.25	41453	41513	41573	41633	41693	41753	41813	41873	41933	41993	727	757	787	817	847	877	907	937	967	997	ACI_ThrowArray[24, 54, 84, 114, 144, 174, 204, 234, 264, 294]
Throw-Item.26	41455	41515	41575	41635	41695	41755	41815	41875	41935	41995	728	758	788	818	848	878	908	938	968	998	ACI_ThrowArray[25, 55, 85, 115, 145, 175, 205, 235, 265, 295]
Throw-Item.27	41457	41517	41577	41637	41697	41757	41817	41877	41937	41997	729	759	789	819	849	879	909	939	969	999	ACI_ThrowArray[26, 56, 86, 116, 146, 176, 206, 236, 266, 296]
Throw-Item.28	41459	41519	41579	41639	41699	41759	41819	41879	41939	41999	730	760	790	820	850	880	910	940	970	1000	ACI_ThrowArray[27, 57, 87, 117, 147, 177, 207, 237, 267, 297]
Throw-Item.29	41461	41521	41581	41641	41701	41761	41821	41881	41941	42001	731	761	791	821	851	881	911	941	971	1001	ACI_ThrowArray[28, 58, 88, 118, 148, 178, 208, 238, 268, 298]
Throw-Item.30	41463	41523	41583	41643	41703	41763	41823	41883	41943	42003	732	762	792	822	852	882	912	942	972	1002	ACI_ThrowArray[29, 59, 89, 119, 149, 179, 209, 239, 269, 299]

Condition Monitoring Only License: All items retrieved will be 0's.

Items Item 1: Compression Forces based on Gas Pressures at Flanges (% of Allowed Limit)
Item 2: Tension Forces based on Gas Pressures at Flanges (% of Allowed Limit)
Item 3: Compression Forces based on Internal Gas Pressures (% of Allowed Limit)
Item 4: Tension Forces based on Internal Gas Pressures (% of Allowed Limit)
Item 5: Compression Forces based on Internal Gas Pressures and Reciprocating Weights (% of Allowed Limit)
Item 6: Tension Forces based on Internal Gas Pressures and Reciprocating Weights (% of Allowed Limit)
Item 7: 1=Passed Pin Reversals, 0=Failed Pin Reversals (1 or 0)
Item 8: (reserved)
Item 9: (reserved)
Item 10: (reserved)
Item 11: Compression Forces based on Gas Pressures at Flanges (Allowed Limit)
Item 12: Tension Forces based on Gas Pressures at Flanges (Allowed Limit)
Item 13: Compression Forces based on Internal Gas Pressures (Allowed Limit)
Item 14: Tension Forces based on Internal Gas Pressures (Allowed Limit)
Item 15: Compression Forces based on Internal Gas Pressures and Reciprocating Weights (Allowed Limit)
Item 16: Tension Forces based on Internal Gas Pressures and Reciprocating Weights (Allowed Limit)
Item 17: True (-1) if a Tandem cylinder is present on this throw, otherwise False (0) is returned.
Item 18: Cylinder Number of cylinder on this throw, or the cylinder number of the Outboard Cylinder if a tandem is present. If cylinder number not found, then a 0 is returned.
Item 19: Cylinder Number of cylinder on this throw, or the cylinder number of the Inboard
Item 20: Degrees of Crank Angle in Compression. (From 0° to 360°)
Item 21: Degrees of Crank Angle in Tension.
Item 22: Reciprocating Weights used on this throw for Rod Load Inertia forces. (lbs)
Item 23: Reversal Weight used on this throw for Pin Load Inertia forces and Degrees of Pin Reversal. (lbs)
Item 24:: Gas-at-Flange Rod Load Compression forces (lbf)
Item 25: Net Rod Load Compression forces (lbf)
Item 26: Net Rod Load Tension forces (lbf)
Item 27: Gas-at-Flange Rod Load Tension forces (lbf)
Item 28: (reserved)
Item 29: Internal Gas-Only Rod Load Compression forces (lbf)
Item 30: Internal Gas-Only Rod Load Tension forces (lbf)

CYL_HE ARRAY:

AB Tag Name

CylHE-4	CylHE-5	CylHE-6	CylHE-7	CylHE-8	CylHE-9	CylHE-10	HE1	HE2	HE3	HE4	HE5	HE6	HE7	HE8	HE9	HE10	ACI_CylHEArray[]
42245	42325	42405	42485	42565	42645	42725	1003	1043	1083	1123	1163	1203	1243	1283	1323	1363	ACI_CylHEArray[0, 40 ,80, 120, 160, 200, 240, 280, 320, 360]
42247	42327	42407	42487	42567	42647	42727	1004	1044	1084	1124	1164	1204	1244	1284	1324	1364	ACI_CylHEArray[1, 41 ,81, 121, 161, 201, 241, 281, 321, 361]
42249	42329	42409	42489	42569	42649	42729	1005	1045	1085	1125	1165	1205	1245	1285	1325	1365	ACI_CylHEArray[2, 42 ,82, 122, 162, 202, 242, 282, 322, 362]
42251	42331	42411	42491	42571	42651	42731	1006	1046	1086	1126	1166	1206	1246	1286	1326	1366	ACI_CylHEArray[3, 43 ,83, 123, 163, 203, 243, 283, 323, 363]
42253	42333	42413	42493	42573	42653	42733	1007	1047	1087	1127	1167	1207	1247	1287	1327	1367	ACI_CylHEArray[4, 44 ,84, 124, 164, 204, 244, 284, 324, 364]
42255	42335	42415	42495	42575	42655	42735	1008	1048	1088	1128	1168	1208	1248	1288	1328	1368	ACI_CylHEArray[5, 45 ,85, 125, 165, 205, 245, 285, 325, 365]
42257	42337	42417	42497	42577	42657	42737	1009	1049	1089	1129	1169	1209	1249	1289	1329	1369	ACI_CylHEArray[6, 46 ,86, 126, 166, 206, 246, 286, 326, 366]
42259	42339	42419	42499	42579	42659	42739	1010	1050	1090	1130	1170	1210	1250	1290	1330	1370	ACI_CylHEArray[7, 47 ,87, 127, 167, 207, 247, 287, 327, 367]
42261	42341	42421	42501	42581	42661	42741	1011	1051	1091	1131	1171	1211	1251	1291	1331	1371	ACI_CylHEArray[8, 48 ,88, 128, 168, 208, 248, 288, 328, 368]
42263	42343	42423	42503	42583	42663	42743	1012	1052	1092	1132	1172	1212	1252	1292	1332	1372	ACI_CylHEArray[9, 49 ,89, 129, 169, 209, 249, 289, 329, 369]
42265	42345	42425	42505	42585	42665	42745	1013	1053	1093	1133	1173	1213	1253	1293	1333	1373	ACI_CylHEArray[10, 50 ,90, 130, 170, 210, 250, 290, 330, 370]
42267	42347	42427	42507	42587	42667	42747	1014	1054	1094	1134	1174	1214	1254	1294	1334	1374	ACI_CylHEArray[11, 51 ,91, 131, 171, 211, 251, 291, 331, 371]
42269	42349	42429	42509	42589	42669	42749	1015	1055	1095	1135	1175	1215	1255	1295	1335	1375	ACI_CylHEArray[12, 52 ,92, 132, 172, 212, 252, 292, 332, 372]
42271	42351	42431	42511	42591	42671	42751	1016	1056	1096	1136	1176	1216	1256	1296	1336	1376	ACI_CylHEArray[13, 53 ,93, 133, 173, 213, 253, 293, 333, 373]
42273	42353	42433	42513	42593	42673	42753	1017	1057	1097	1137	1177	1217	1257	1297	1337	1377	ACI_CylHEArray[14, 54 ,94, 134, 174, 214, 254, 294, 334, 374]
42275	42355	42435	42515	42595	42675	42755	1018	1058	1098	1138	1178	1218	1258	1298	1338	1378	ACI_CylHEArray[15, 55 ,95, 135, 175, 215, 255, 295, 335, 375]
42277	42357	42437	42517	42597	42677	42757	1019	1059	1099	1139	1179	1219	1259	1299	1339	1379	ACI_CylHEArray[16, 56 ,96, 136, 176, 216, 256, 296, 336, 376]
42279	42359	42439	42519	42599	42679	42759	1020	1060	1100	1140	1180	1220	1260	1300	1340	1380	ACI_CylHEArray[17, 57 ,97, 137, 177, 217, 257, 297, 337, 377]
42281	42361	42441	42521	42601	42681	42761	1021	1061	1101	1141	1181	1221	1261	1301	1341	1381	ACI_CylHEArray[18, 58 ,98, 138, 178, 218, 258, 298, 338, 378]
42283	42363	42443	42523	42603	42683	42763	1022	1062	1102	1142	1182	1222	1262	1302	1342	1382	ACI_CylHEArray[19, 59 ,99, 139, 179, 219, 259, 299, 339, 379]
42285	42365	42445	42525	42605	42685	42765	1023	1063	1103	1143	1183	1223	1263	1303	1343	1383	ACI_CylHEArray[20, 60 ,100, 140, 180, 220, 260, 300, 340, 380]
42287	42367	42447	42527	42607	42687	42767	1024	1064	1104	1144	1184	1224	1264	1304	1344	1384	ACI_CylHEArray[21, 61 ,101, 141, 181, 221, 261, 301, 341, 381]
42289	42369	42449	42529	42609	42689	42769	1025	1065	1105	1145	1185	1225	1265	1305	1345	1385	ACI_CylHEArray[22, 62 ,102, 142, 182, 222, 262, 302, 342, 382]
42291	42371	42451	42531	42611	42691	42771	1026	1066	1106	1146	1186	1226	1266	1306	1346	1386	ACI_CylHEArray[23, 63 ,103, 143, 183, 223, 263, 303, 343, 383]
42293	42373	42453	42533	42613	42693	42773	1027	1067	1107	1147	1187	1227	1267	1307	1347	1387	ACI_CylHEArray[24, 64 ,104, 144, 184, 224, 264, 304, 344, 384]
42295	42375	42455	42535	42615	42695	42775	1028	1068	1108	1148	1188	1228	1268	1308	1348	1388	ACI_CylHEArray[25, 65 ,105, 145, 185, 225, 265, 305, 345, 385]
42297	42377	42457	42537	42617	42697	42777	1029	1069	1109	1149	1189	1229	1269	1309	1349	1389	ACI_CylHEArray[26, 66 ,106, 146, 186, 226, 266, 306, 346, 386]
42299	42379	42459	42539	42619	42699	42779	1030	1070	1110	1150	1190	1230	1270	1310	1350	1390	ACI_CylHEArray[27, 67 ,107, 147, 187, 227, 267, 307, 347, 387]
42301	42381	42461	42541	42621	42701	42781	1031	1071	1111	1151	1191	1231	1271	1311	1351	1391	ACI_CylHEArray[28, 68 ,108, 148, 188, 228, 268, 308, 348, 388]
42303	42383	42463	42543	42623	42703	42783	1032	1072	1112	1152	1192	1232	1272	1312	1352	1392	ACI_CylHEArray[29, 69 ,109, 149, 189, 229, 269, 309, 349, 389]
42305	42385	42465	42545	42625	42705	42785	1033	1073	1113	1153	1193	1233	1273	1313	1353	1393	ACI_CylHEArray[30, 70 ,110, 150, 190, 230, 270, 310, 350, 390]
42307	42387	42467	42547	42627	42707	42787	1034	1074	1114	1154	1194	1234	1274	1314	1354	1394	ACI_CylHEArray[31, 71 ,111, 151, 191, 231, 271, 311, 351, 391]
42309	42389	42469	42549	42629	42709	42789	1035	1075	1115	1155	1195	1235	1275	1315	1355	1395	ACI_CylHEArray[32, 72 ,112, 152, 192, 232, 272, 312, 352, 392]
42311	42391	42471	42551	42631	42711	42791	1036	1076	1116	1156	1196	1236	1276	1316	1356	1396	ACI_CylHEArray[33, 73 ,113, 153, 193, 233, 273, 313, 353, 393]
42313	42393	42473	42553	42633	42713	42793	1037	1077	1117	1157	1197	1237	1277	1317	1357	1397	ACI_CylHEArray[34, 74 ,114, 154, 194, 234, 274, 314, 354, 394]
42315	42395	42475	42555	42635	42715	42795	1038	1078	1118	1158	1198	1238	1278	1318	1358	1398	ACI_CylHEArray[35, 75 ,115, 155, 195, 235, 275, 315, 355, 395]
42317	42397	42477	42557	42637	42717	42797	1039	1079	1119	1159	1199	1239	1279	1319	1359	1399	ACI_CylHEArray[36, 76 ,116, 156, 196, 236, 276, 316, 356, 396]
42319	42399	42479	42559	42639	42719	42799	1040	1080	1120	1160	1200	1240	1280	1320	1360	1400	ACI_CylHEArray[37, 77 ,117, 157, 197, 237, 277, 317, 357, 397]
42321	42401	42481	42561	42641	42721	42801	1041	1081	1121	1161	1201	1241	1281	1321	1361	1401	ACI_CylHEArray[38, 78 ,118, 158, 198, 238, 278, 318, 358, 398]
42323	42403	42483	42563	42643	42723	42803	1042	1082	1122	1162	1202	1242	1282	1322	1362	1402	ACI_CylHEArray[39, 79 ,119, 159, 199, 239, 279, 319, 359, 399]

All items retrieved will be 0's, except items #1-4, 11.

Items

Item 1: Suction Pressure into cylinder (psig)
Item 2: Discharge Pressure out of cylinder (psig)
Item 3: Suction temperature of gas into cylinder (°F)
Item 4: Estimated discharge temperature of cylinder (°F)
Item 5: Z-Suction Compressibility Factor
Item 6: Z-Discharge Compressibility Factor
Item 7: Suction Volumetric Efficiency (%)
Item 8: Discharge Volumetric Efficiency (%)
Item 9: Fixed clearance (%) of cylinder end
Item 10: Effective clearance (%) of cylinder end
Item 11: Isentropic Efficiency (%) per end
Item 12: Adiabatic horsepower (BHP) used to compress gas per end
Item 13: Valve loss (BHP) used to move gas through valves
Item 14: Parasitic loss (BHP) used when end is deactivated
Item 15: Total load (BHP) used per end
Item 16: Flow (MMscfd) per end
Item 17: Which Stage of compression this cylinder serves
Item 18: Bore Diameter (in)
Item 19: Rod Diameter (in)
Item 20: End Active (1=Active, 0=Deactivated)
Item 21: Cylinder Friction (fraction)
Item 22: Cylinder Slippage (fraction)
Item 23: Deactivation Pressure source used (0=Suction, 1=Discharge)
Item 24: Cylinder MAWP (or max allowed pressure set in eRCM model) (psiG)
Item 25: Suction Valves being Used in Active Compression Mode
Item 26: Discharge Valves being Used in Active Compression Mode
Item 27: Suction Valves being Used in Deactivated End Mode
Item 28: Discharge Valves being Used in Deactivated End Mode
Item 29: Throw Number on which this cylinder is located.
Item 30: (reserved)
Item 31: (reserved)
Item 32: (reserved)
Item 33: Maximum allowed Discharge Temperature for this cylinder (°F)
Item 34: (reserved)
Item 35: (reserved)
Item 36: (reserved)
Item 37: (reserved)
Item 38: Length of the Stroke for the VVCP on this cylinder end. (in)
Item 39: (reserved)
Item 40: (reserved)

CYL_CE ARRAY:

AB Tag Name

CylCE-3	CylCE-4	CylCE-5	CylCE-6	CylCE-7	CylCE-8	CylCE-9	CylCE-10	HE1	HE2	HE3	HE4	HE5	HE6	HE7	HE8	HE9	HE10	ACI_CylCEArray[]
42965	43045	43125	43205	43285	43365	43445	43525	1403	1443	1483	1523	1563	1603	1643	1683	1723	1763	ACI_CylCEArray[0, 40, 80, 120, 160, 200, 240, 280, 320, 360]
42967	43047	43127	43207	43287	43367	43447	43527	1404	1444	1484	1524	1564	1604	1644	1684	1724	1764	ACI_CylCEArray[1, 41, 81, 121, 161, 201, 241, 281, 321, 361]
42969	43049	43129	43209	43289	43369	43449	43529	1405	1445	1485	1525	1565	1605	1645	1685	1725	1765	ACI_CylCEArray[2, 42, 82, 122, 162, 202, 242, 282, 322, 362]
42971	43051	43131	43211	43291	43371	43451	43531	1406	1446	1486	1526	1566	1606	1646	1686	1726	1766	ACI_CylCEArray[3, 43, 83, 123, 163, 203, 243, 283, 323, 363]
42973	43053	43133	43213	43293	43373	43453	43533	1407	1447	1487	1527	1567	1607	1647	1687	1727	1767	ACI_CylCEArray[4, 44, 84, 124, 164, 204, 244, 284, 324, 364]
42975	43055	43135	43215	43295	43375	43455	43535	1408	1448	1488	1528	1568	1608	1648	1688	1728	1768	ACI_CylCEArray[5, 45, 85, 125, 165, 205, 245, 285, 325, 365]
42977	43057	43137	43217	43297	43377	43457	43537	1409	1449	1489	1529	1569	1609	1649	1689	1729	1769	ACI_CylCEArray[6, 46, 86, 126, 166, 206, 246, 286, 326, 366]
42979	43059	43139	43219	43299	43379	43459	43539	1410	1450	1490	1530	1570	1610	1650	1690	1730	1770	ACI_CylCEArray[7, 47, 87, 127, 167, 207, 247, 287, 327, 367]
42981	43061	43141	43221	43301	43381	43461	43541	1411	1451	1491	1531	1571	1611	1651	1691	1731	1771	ACI_CylCEArray[8, 48, 88, 128, 168, 208, 248, 288, 328, 368]
42983	43063	43143	43223	43303	43383	43463	43543	1412	1452	1492	1532	1572	1612	1652	1692	1732	1772	ACI_CylCEArray[9, 49, 89, 129, 169, 209, 249, 289, 329, 369]
42985	43065	43145	43225	43305	43385	43465	43545	1413	1453	1493	1533	1573	1613	1653	1693	1733	1773	ACI_CylCEArray[10, 50, 90, 130, 170, 210, 250, 290, 330, 370]
42987	43067	43147	43227	43307	43387	43467	43547	1414	1454	1494	1534	1574	1614	1654	1694	1734	1774	ACI_CylCEArray[11, 51, 91, 131, 171, 211, 251, 291, 331, 371]
42989	43069	43149	43229	43309	43389	43469	43549	1415	1455	1495	1535	1575	1615	1655	1695	1735	1775	ACI_CylCEArray[12, 52, 92, 132, 172, 212, 252, 292, 332, 372]
42991	43071	43151	43231	43311	43391	43471	43551	1416	1456	1496	1536	1576	1616	1656	1696	1736	1776	ACI_CylCEArray[13, 53, 93, 133, 173, 213, 253, 293, 333, 373]
42993	43073	43153	43233	43313	43393	43473	43553	1417	1457	1497	1537	1577	1617	1657	1697	1737	1777	ACI_CylCEArray[14, 54, 94, 134, 174, 214, 254, 294, 334, 374]
42995	43075	43155	43235	43315	43395	43475	43555	1418	1458	1498	1538	1578	1618	1658	1698	1738	1778	ACI_CylCEArray[15, 55, 95, 135, 175, 215, 255, 295, 335, 375]
42997	43077	43157	43237	43317	43397	43477	43557	1419	1459	1499	1539	1579	1619	1659	1699	1739	1779	ACI_CylCEArray[16, 56, 96, 136, 176, 216, 256, 296, 336, 376]
42999	43079	43159	43239	43319	43399	43479	43559	1420	1460	1500	1540	1580	1620	1660	1700	1740	1780	ACI_CylCEArray[17, 57, 97, 137, 177, 217, 257, 297, 337, 377]
43001	43081	43161	43241	43321	43401	43481	43561	1421	1461	1501	1541	1581	1621	1661	1701	1741	1781	ACI_CylCEArray[18, 58, 98, 138, 178, 218, 258, 298, 338, 378]
43003	43083	43163	43243	43323	43403	43483	43563	1422	1462	1502	1542	1582	1622	1662	1702	1742	1782	ACI_CylCEArray[19, 59, 99, 139, 179, 219, 259, 299, 339, 379]
43005	43085	43165	43245	43325	43405	43485	43565	1423	1463	1503	1543	1583	1623	1663	1703	1743	1783	ACI_CylCEArray[20, 60, 100, 140, 180, 220, 260, 300, 340, 380]
43007	43087	43167	43247	43327	43407	43487	43567	1424	1464	1504	1544	1584	1624	1664	1704	1744	1784	ACI_CylCEArray[21, 61, 101, 141, 181, 221, 261, 301, 341, 381]
43009	43089	43169	43249	43329	43409	43489	43569	1425	1465	1505	1545	1585	1625	1665	1705	1745	1785	ACI_CylCEArray[22, 62, 102, 142, 182, 222, 262, 302, 342, 382]
43011	43091	43171	43251	43331	43411	43491	43571	1426	1466	1506	1546	1586	1626	1666	1706	1746	1786	ACI_CylCEArray[23, 63, 103, 143, 183, 223, 263, 303, 343, 383]
43013	43093	43173	43253	43333	43413	43493	43573	1427	1467	1507	1547	1587	1627	1667	1707	1747	1787	ACI_CylCEArray[24, 64, 104, 144, 184, 224, 264, 304, 344, 384]
43015	43095	43175	43255	43335	43415	43495	43575	1428	1468	1508	1548	1588	1628	1668	1708	1748	1788	ACI_CylCEArray[25, 65, 105, 145, 185, 225, 265, 305, 345, 385]
43017	43097	43177	43257	43337	43417	43497	43577	1429	1469	1509	1549	1589	1629	1669	1709	1749	1789	ACI_CylCEArray[26, 66, 106, 146, 186, 226, 266, 306, 346, 386]
43019	43099	43179	43259	43339	43419	43499	43579	1430	1470	1510	1550	1590	1630	1670	1710	1750	1790	ACI_CylCEArray[27, 67, 107, 147, 187, 227, 267, 307, 347, 387]
43021	43101	43181	43261	43341	43421	43501	43581	1431	1471	1511	1551	1591	1631	1671	1711	1751	1791	ACI_CylCEArray[28, 68, 108, 148, 188, 228, 268, 308, 348, 388]
43023	43103	43183	43263	43343	43423	43503	43583	1432	1472	1512	1552	1592	1632	1672	1712	1752	1792	ACI_CylCEArray[29, 69, 109, 149, 189, 229, 269, 309, 349, 389]
43025	43105	43185	43265	43345	43425	43505	43585	1433	1473	1513	1553	1593	1633	1673	1713	1753	1793	ACI_CylCEArray[30, 70, 110, 150, 190, 230, 270, 310, 350, 390]
43027	43107	43187	43267	43347	43427	43507	43587	1434	1474	1514	1554	1594	1634	1674	1714	1754	1794	ACI_CylCEArray[31, 71, 111, 151, 191, 231, 271, 311, 351, 391]
43029	43109	43189	43269	43349	43429	43509	43589	1435	1475	1515	1555	1595	1635	1675	1715	1755	1795	ACI_CylCEArray[32, 72, 112, 152, 192, 232, 272, 312, 352, 392]
43031	43111	43191	43271	43351	43431	43511	43591	1436	1476	1516	1556	1596	1636	1676	1716	1756	1796	ACI_CylCEArray[33, 73, 113, 153, 193, 233, 273, 313, 353, 393]
43033	43113	43193	43273	43353	43433	43513	43593	1437	1477	1517	1557	1597	1637	1677	1717	1757	1797	ACI_CylCEArray[34, 74, 114, 154, 194, 234, 274, 314, 354, 394]
43035	43115	43195	43275	43355	43435	43515	43595	1438	1478	1518	1558	1598	1638	1678	1718	1758	1798	ACI_CylCEArray[35, 75, 115, 155, 195, 235, 275, 315, 355, 395]
43037	43117	43197	43277	43357	43437	43517	43597	1439	1479	1519	1559	1599	1639	1679	1719	1759	1799	ACI_CylCEArray[36, 76, 116, 156, 196, 236, 276, 316, 356, 396]
43039	43119	43199	43279	43359	43439	43519	43599	1440	1480	1520	1560	1600	1640	1680	1720	1760	1800	ACI_CylCEArray[37, 77, 117, 157, 197, 237, 277, 317, 357, 397]
43041	43121	43201	43281	43361	43441	43521	43601	1441	1481	1521	1561	1601	1641	1681	1721	1761	1801	ACI_CylCEArray[38, 78, 118, 158, 198, 238, 278, 318, 358, 398]
43043	43123	43203	43283	43363	43443	43523	43603	1442	1482	1522	1562	1602	1642	1682	1722	1762	1802	ACI_CylCEArray[39, 79, 119, 159, 199, 239, 279, 319, 359, 399]

License: All items retrieved will be 0's, except items #1-4, 11.

Item	Item 1: Suction Pressure into cylinder (psig)	Item 21: Cylinder Friction (fraction)
	Item 2: Discharge Pressure out of cylinder (psig)	Item 22: Cylinder Slippage (fraction)
	Item 3: Suction temperature of gas into cylinder (°F)	Item 23: Deactivation Pressure source used (0=Suction, 1=Discharge)
	Item 4: Estimated discharge temperature of cylinder (°F)	Item 24: Cylinder MAWP (or max allowed pressure set in eRCM model) (psiG)
	Item 5: Z-Suction Compressibility Factor	Item 25: Suction Valves being Used in Active Compression Mode
	Item 6: Z-Discharge Compressibility Factor	Item 26: Discharge Valves being Used in Active Compression Mode
	Item 7: Suction Volumetric Efficiency (%)	Item 27: Suction Valves being Used in Deactivated End Mode
	Item 8: Discharge Volumetric Efficiency (%)	Item 28: Discharge Valves being Used in Deactivated End Mode
	Item 9: Fixed clearance (%) of cylinder end	Item 29: Throw Number on which this cylinder is located.
	Item 10: Effective clearance (%) of cylinder end	*Item 30: (reserved)*
	Item 11: Isentropic Efficiency (%) per end	*Item 31: (reserved)*
	Item 12: Adiabatic horsepower (BHP) used to compress gas per end	*Item 32: (reserved)*
	Item 13: Valve loss (BHP) used to move gas through valves	Item 33: Maximum allowed Discharge Temperature for this cylinder (°F)
	Item 14: Parasitic loss (BHP) used when end is deactivated	*Item 34: (reserved)*
	Item 15: Total load (BHP) used per end	*Item 35: (reserved)*
	Item 16: Flow (MMscfd) per end	*Item 36: (reserved)*
	Item 17: Which Stage of compression this cylinder serves	*Item 37: (reserved)*
	Item 18: Bore Diameter (in)	Item 38: Length of the Stroke for the VVCP on this cylinder end. (in)
	Item 19: Rod Diameter (in)	*Item 39: (reserved)*
	Item 20: End Active (1=Active, 0=Deactivated)	*Item 40: (reserved)*

RANGES:

ITEM	ADDRESS	AB Tag Name	Item Description
Range-01	43605	ACI_Ranges[0]	01. Suction Pressure (psiG) Min
Range-02	43607	ACI_Ranges[1]	02. Suction Pressure (psiG) Max
Range-03	43609	ACI_Ranges[2]	03. Discharge Pressure (psiG) Min
Range-04	43611	ACI_Ranges[3]	04. Discharge Pressure (psiG) Max
Range-05	43613	ACI_Ranges[4]	05. Speed (RPM) Min
Range-06	43615	ACI_Ranges[5]	06. Speed (RPM) Max
Range-07	43617	ACI_Ranges[6]	07. Torque (%) Min
Range-08	43619	ACI_Ranges[7]	08. Torque (%) Max
Range-09	43621	ACI_Ranges[8]	09. Ambient Temperature (°F) Min
Range-10	43623	ACI_Ranges[9]	10. Ambient Temperature (°F) Max
Range-11	43625	ACI_Ranges[10]	11. Stage 1 Suction (°F) Min (where WhichStage is Specified)
Range-12	43627	ACI_Ranges[11]	12. Stage 1 Suction (°F) Max (where WhichStage is Specified)
Range-13	43629	ACI_Ranges[12]	13. Stage 2 Suction (°F) Min (where WhichStage is Specified)
Range-14	43631	ACI_Ranges[13]	14. Stage 2 Suction (°F) Max (where WhichStage is Specified)
Range-15	43633	ACI_Ranges[14]	15. Stage 3 Suction (°F) Min (where WhichStage is Specified)
Range-16	43635	ACI_Ranges[15]	16. Stage 3 Suction (°F) Max (where WhichStage is Specified)
Range-17	43637	ACI_Ranges[16]	17. Stage 4 Suction (°F) Min (where WhichStage is Specified)
Range-18	43639	ACI_Ranges[17]	18. Stage 4 Suction (°F) Max (where WhichStage is Specified)
Range-19	43641	ACI_Ranges[18]	19. Stage 5 Suction (°F) Min (where WhichStage is Specified)
Range-20	43643	ACI_Ranges[19]	20. Stage 5 Suction (°F) Max (where WhichStage is Specified)
Range-21	43645	ACI_Ranges[20]	21. Stage 6 Suction (°F) Min (where WhichStage is Specified)
Range-22	43647	ACI_Ranges[21]	22. Stage 6 Suction (°F) Max (where WhichStage is Specified)

eRCM Express™

Sample Pseudo

Start Up Code

This is just pseudo code for understanding concepts and order. Actual code, syntax, optimization, calls, etc. are handled by PLC programmers.

```
// ======================================
// Modbus Block Section References
// Used by PLC routines interacting with eRCM Express
// ======================================
eRCM_CommonWritesStart    = 40001
eRCM_CommonWritesLen      = 10
eRCM_SetViewerFileStart   = 40101
eRCM_SetViewerFileLen     = 1
eRCM_ForceCalcsStart      = 40273
eRCM_ForceCalcsLen        = 1
eRCM_CheckSafeStartUp     = 40107
eRCM_CheckSafeStartUpLen  = 1
eRCM_LStoSetforSU         = 40329
eRCM_LStoSetforSULen      = 1

eRCM_IsKernelBusyStart    = 40275
eRCM_IsKernelBusyLen      = 1
eRCM_CommonReadsStart     = 40277
eRCM_CommonReadsLen       = 13
eRCM_CommonFixedDataStart = 40325
eRCM_CommonFixedDataLen   = 26

eRCM_ErrorArrayStart      = 40445
eRCM_LoadArrayStart       = 40645
eRCM_FlowArrayStart       = 40845
// Make the ArrayLen equal to the max number of Load Steps from all model files!
eRCM_ArrayLen             = 30

eRCM_StageInfoStart       = 41045
eRCM_StageInfoPerStageLen = 30
eRCM_ThrowInfoStart       = 41405
eRCM_ThrowInfoPerThrowLen = 30
eRCM_HECylInfoStart       = 42005
eRCM_HECylInfoPerHECylLen = 40
eRCM_CECylInfoStart       = 42805
eRCM_CECylInfoPerCECylLen = 40
eRCM_RangeInfoStart       = 43605
eRCM_RangeInfoLen         = 22

eRCMExpressCount as Integer
// ======================================
```

```
// ======================================
// Compressor is warmed up, and we are ready to bring compressor online.
// First, load in the correct eRCM Viewer model, and then verify it.
// ======================================

  // Load in the correct file
  Modbus.NumberOfDataValues = eRCM_SetViewerFileLen
  Modbus.Address = eRCM_SetViewerFileStart
  Modbus.Data(1) = 2     // Load in the "2Two-stg Injection.rvf" file
  Modbus.Write()
  Sleep 500 ms

  // Verify that this file is correct. If should be a 2-stage unit with 6 throws,
  // 6 cylinders, and 23 load steps.
  Modbus.NumberOfDataValues = eRCM_CommonFixedDataLen
  Modbus.Address = eRCM_CommonFixedDataStart
  Modbus.Read()
  // All must be true, else wrong file!
  // NumStgs=2 AND NumCyls=6 AND NumThrows=6 AND NumLSs=23
  IF Modbus.Data(11)=2 AND Modbus.Data(9)=6 AND _
     Modbus.Data(12)=6 AND Modbus.Data(10)=23 AND
     Modbus.Data(26)=123321 THEN    ' 123321 is User ID value for this model.
     MESSAGE "Correct eRCM Viewer model loaded."
  ELSE
     ALARM "Wrong eRCM Viewer model in eRCM Express. Stop unit!"
     SHUTDOWN
     END
  END IF

  // Check to see if we can force unit to start in the safest load step.
  // This is only practical if access to the unit's final discharge pressure
  // is known. Unit's "actual" discharge pressure is likely about the same as
  // suction pressure since we are in bypass mode.
  IF (Field Discharge Pressure is Known) THEN

    Modbus.NumberOfDataValues = eRCM_CommonWritesLen
    Modbus.Address = eRCM_CommonWritesStart
    Modbus.Data(1) = UnitABC.CurrentLoadStep
    Modbus.Data(2) = UnitABC.Stage1.Ps_PSIG
    Modbus.Data(3) = UnitABC.Station.Pd_PSIG      // This item is SPECIAL!
    Modbus.Data(4) = UnitABC.Driver.CurrentRPM
    Modbus.Data(5) = UnitABC.Stage1.Ts_F
    Modbus.Data(6) = UnitABC.Stage2.Ts_F
    Modbus.Data(7) = 0
    Modbus.Data(8) = 0
    Modbus.Data(9) = 0
    Modbus.Data(10) = 0
    Modbus.Write()

    Modbus.NumberOfDataValues = eRCM_CheckSafeStartUpLen
    Modbus.Address = eRCM_CheckSafeStartUp
    Modbus.Data(1) = 1     // The "1" tells eRCM Express to return Safe Start Up LS
    Modbus.Write()

    Modbus.NumberOfDataValues = eRCM_ForceCalcsLen
    Modbus.Address = eRCM_ForceCalcsStart
    Modbus.Data(1) = 1  // Force a calculation of the compressor model
```

```
    Modbus.Write()
```

TWO OPTIONS - DO ONE OR THE OTHER.

```
    // Wait for eRCM Express to finish calculation (make a subroutine!)
    // This fully guarantees that the data being read back is appropriate for
    // the operating data sent to the eRCM Express.
    Modbus.NumberOfDataValues = eRCM_IsKernelBusyLen
    Modbus.Address = eRCM_IsKernelBusyStart
    eRCMExpressCount = 0
    DO
      Sleep 20 ms
      Modbus.ReadData()       // Read data in the "IsKernelBusy" register.
      If Modbus.Data(1) = 0 Then Exit DO     // If 0, calculations now complete.
      eRCMExpressCount = eRCMExpressCount + 1
    LOOP UNTIL eRCMExpressCount > 10         // Prevent Infinite Loop
    IF eRCMExpressCount > 10 THEN
      MSG "CRITICAL ISSUE: eRCM Express Unit does not appear to be returning
data."
      // NOTE: If gateway device used, you may need to wait longer for updates.
      EXIT
    END IF
```

```
    // This method avoids needing to poll the IsKernelBusy register until it goes
    // back to zero (0). Most calculations will complete in 150 ms. Thus, simply
    // wait a long enough period to guarantee calculations are complete.
    // However, for some models, you may need to wait 250-350 ms for completion.
    Sleep 150 ms
```

```
    Modbus.NumberOfDataValues = eRCM_ LStoSetforSULen
    Modbus.Address = eRCM_ LStoSetforSU
    Modbus.ReadData()
    IF Modbus.Data(1) <> -1 THEN
      CurrentLoadStep = Modbus.Data(1)
    ELSE
      // No safe load step found! Not good.
      // Company needs to decide if unit should NOT be allowed to go ONLINE, or
      // if they want to force a default load step to use instead.
      CurrentLoadStep = -1
    ENDIF

    Modbus.NumberOfDataValues = eRCM_CheckSafeStartUpLen
    Modbus.Address = eRCM_CheckSafeStartUp
    Modbus.Data(1) = 0    // The "0" tells eRCM Express to stop determining Safe
Start Up LS
    Modbus.Write()

  ELSE

    // If we do not have access to final discharge pressure, then it is not clear
    // which (if any) load step to implement to bring unit online safely. Thus,
    // based on OEM, Operators, Engineering, the load step to use during closing
    // of the Bypass Valve has been determined to be "???". Thus, implement this
    // step.
    CurrentLoadStep = 23    // For Two-stage injection, use last load step, LS#23.

  ENDIF
```

```
  IF CurrentLoadStep > 0 THEN
    IMPLEMENT_LoadStep // Call routine to physically implement current load step
  ELSE
    MSG "CRITICAL ISSUE: No safe load steps found for Start Up. Shut Down!"
    EXIT
  END IF

// ======================================
```

The code below is relevant to AFTER the unit is online.

```
// ======================================
// Compressor is now online - Bypass Fully Closed.
// ======================================

  // Now, pass data to eRCM, get results back and take actions accordingly

    // Pass the current operating data
    Modbus.NumberOfDataValues = eRCM_CommonWritesLen
    Modbus.Address = eRCM_CommonWritesStart
    Modbus.Data(1) = UnitABC.CurrentLoadStep
    Modbus.Data(2) = UnitABC.Stage1.Ps_PSIG
    Modbus.Data(3) = UnitABC.Stage2.Pd_PSIG
    Modbus.Data(4) = UnitABC.Driver.CurrentRPM
    Modbus.Data(5) = UnitABC.Stage1.Ts_F
    Modbus.Data(6) = UnitABC.Stage2.Ts_F
    Modbus.Data(7) = 0
    Modbus.Data(8) = 0
    Modbus.Data(9) = 0
    Modbus.Data(10) = 0
    Modbus.Write()

    Modbus.NumberOfDataValues = eRCM_ForceCalcsLen
    Modbus.Address = eRCM_ForceCalcsStart
    Modbus.Data(1) = 1  // Force a calculation of the compressor model
    Modbus.Write()

    // Wait for eRCM Express to finish calculation (Choose one of two options)
    Sleep 150 ms // May need to be set higher (i.e. 250-350 ms for some models)

    // Get results back
    Modbus.NumberOfDataValues = eRCM_CommonReadsLen
    Modbus.Address = eRCM_CommonReadsStart
    Modbus.Read()
    UnitABC.NSU       = Modbus.ReadData(2)
    UnitABC.NSD       = Modbus.ReadData(3)
    UnitABC.NSU_Perc  = Modbus.ReadData(4)
    UnitABC.NSD_Perc  = Modbus.ReadData(5)
    UnitABC.MinSpeedCurrLS = Modbus.ReadData(6)
    UnitABC.MaxSpeedCurrLS = Modbus.ReadData(7)
    UnitABC.MinPsCurrLS    = Modbus.ReadData(8)
    UnitABC.MaxPsCurrLS    = Modbus.ReadData(9)
    UnitABC.OptimalLS     = Modbus.ReadData(10)
    UnitABC.CurrTorq_Perc = Modbus.ReadData(11)
    UnitABC.UnitEff_Perf  = Modbus.ReadData(12)
```

```
UnitABC.FuelRate       = Modbus.ReadData(13)

Modbus.NumberOfDataValues = eRCM_ArrayLen
Modbus.Address = eRCM_ErrorArrayStart
Modbus.Read()
FOR I = 1 TO UnitABC.NumLoadSteps
  UnitABC.ErrorArray(I) = Modbus.ReadData(I)
NEXT I
Modbus.Address = eRCM_LoadArrayStart
Modbus.Read()
FOR I = 1 TO UnitABC.NumLoadSteps
  UnitABC.LoadArray(I) = Modbus.ReadData(I)
NEXT I
Modbus.Address = eRCM_FlowArrayStart
Modbus.Read()
FOR I = 1 TO UnitABC.NumLoadSteps
  UnitABC.FlowArray(I) = Modbus.ReadData(I)
NEXT I

// This remaining data is useful for display, but does not normally need to be
// retrieved each cycle. Rather, only retrieve it when needed for display.
// ---------------------------------

Modbus.NumberOfDataValues = eRCM_StageInfoPerStageLen
Modbus.Address = eRCM_StageInfoStart
FOR I = 1 TO UnitABC.NumStages
  Modbus.Read()
  FOR N = 1 to eRCM_StageInfoPerStageLen
    UnitABC.StageArray(I, N) = Modbus.ReadData(I)
  NEXT N
  Modbus.Address = Modbus.Address + 2*eRCM_StageInfoPerStageLen
NEXT I

Modbus.NumberOfDataValues = eRCM_ThrowInfoPerThrowLen
Modbus.Address = eRCM_ThrowInfoStart
FOR I = 1 TO UnitABC.NumThrows
  Modbus.Read()
  FOR N = 1 to eRCM_ThrowInfoPerThrowLen
    UnitABC.ThrowArray(I, N) = Modbus.ReadData(I)
  NEXT N
  Modbus.Address = Modbus.Address + 2* eRCM_ThrowInfoPerThrowLen
NEXT I

Modbus.NumberOfDataValues = eRCM_HECylInfoPerHECylLen
Modbus.Address = eRCM_HECylInfoStart
FOR I = 1 TO UnitABC.NumCylinders
  Modbus.Read()
  FOR N = 1 to eRCM_HECylInfoPerHECylLen
    UnitABC.HECylArray(I, N) = Modbus.ReadData(I)
  NEXT N
  Modbus.Address = Modbus.Address + 2*eRCM_HECylInfoPerHECylLen
NEXT I

Modbus.NumberOfDataValues = eRCM_CECylInfoPerCECylLen
Modbus.Address = eRCM_CECylInfoStart
FOR I = 1 TO UnitABC.NumCylinders
  Modbus.Read()
```

```
      FOR N = 1 to eRCM_CECylInfoPerCECylLen
        UnitABC.CECylArray(I, N) = Modbus.ReadData(I)
      NEXT N
      Modbus.Address = Modbus.Address + 2*eRCM_CECylInfoPerCECylLen
    NEXT I
    // ---------------------------------


    // Act on those results
    … Expertise of the automation/control team …

    // Compressor Errors for Unit (Compressor and Driver) Errors
    // The ErrorArray only returns errors associated with the compressor
    //   being unsafe. Overloading and/or underloading a driver (engine or motor)
    //   can lead to issues. Thus, if you want to invalidate load steps based on
    //   driver issues, the PLC needs to do that.
    // Next Step Up, Next Step Down, and OptimalLS are based on compressor safety
    //   and on over/under loading of driver, and on making sure that load step
    //   changes are not so larger that the change in load can lead to the engine
    //   surging and/or shutting down.

    // IF need for more load and NSU ≠ -1 THEN Load up to NSU…

    // IF need for less load and NSD ≠ -1 THEN Load down to NSD…

    // CRITICAL CHECK
    // ==============================
    // IF NSU=-1 AND NSD=-1 AND ErrorArray(CurrentLS) ≠ 0 THEN SHUT DOWN! No safe
load steps!
    // ==============================

    // Display desired items of interest on various screens…

// ======================================
```

eRCM Express™

Modbus Registers - Reference Changes when Replacing Older Units

When replacing an older eRCM Express unit (one with an industrial PC as its platform) with an eRCM Express with an mCore SDR as its platform, some Modbus register numbers will need to be changed, and a few other changes in the PLC and/or gateway devices will also need to be implemented.

- When a recalculation of compressor performance and safety is desired, then after sending the operating conditions, a value of one (1) needs to be set in the **ForceERCMExpressCalculations** register and sent to eRCM Express.
  - Note: There are two (2) registers named ForceERCMExpressCalculations. The Reg#40075 is more ideal for users only using the Quick Write List so that the entire group of registers can be written in one Modbus block. Nevertheless, setting either register to a one (1) will direct eRCM Express to recalculate the compressor performance based on latest inputs.

- Interpretation of the bits in the ErrorArray() is still the same. However, retrieving the data for that array is a bit different now:
  - Previous: The registers were retrieved as FLOATS, which had to be converted to INTEGERs within the PLC before inspecting the bits.
  - New: The registers for (only) this array must now be retrieved directly as INTEGER, and thus no conversion is required before inspecting the bits.

To help with the updating of Modbus registers, the following tables show the previous Modbus register used, and its new corresponding register value in the mCore SDR unit:

eRCM Express - PC Platform (Old)	
NAME	ADDRESS
CurrLS	40005
PsG	40009
PdG	40011
RPM	40013
Ts1F	40015
Ts2F	40017
Ts3F	40019
Ts4F	40021
Ts5F	40023
Ts6F	40025
Next Viewer File Request	42799
DriverMaxBHP	42801
LSSelectionOnFlow	42723
CheckSafeStartup	42725
TorqSP	40001
LSMode	40003
TambF	40007
MinLoadFlowChangeAllowed	42719
MaxLoadChangeAllowed	42721

eRCM Express - mCore Platform (New)		
NAME	ADDRESS	
CurrLS	40001	
PsG	40003	
PdG	40005	
CurrSpeed	40007	
Ts1F	40009	
Ts2F	40011	
Ts3F	40013	
Ts4F	40015	
Ts5F	40017	
Ts6F	40019	
SetViewerFile	40101	
SetDriverMaxHP	40103	
LSSOnFlow	40105	
CheckSafeStartUp	40107	
TorqSP	40109	
LSMode	40111	
TambF	40113	
SetMinPercChange	40117	
SetMaxPercChange	40119	
ForceERCMExpressCalculations	40273	Must be set to "1" to force recalculations.

KrnlBsy	40027
WDPulse	40075
NxtLSUP	40031
NxtLSDN	40033
NSU_Perc	40071
NSD_Perc	40073
MinRPM	40035
MaxRPM	40037
MinPs1	40039
MaxPs1	40041
OptmLS	40029
CurrTorq	40047
IsenEff	40053
SafeLoadStepStartup	42727
AuxLoad	40045
BHPMax	40049
BHPMax at TorqSP	40055
Elevation	40051
Patm	40043
NumCyls	40061
NumLSs	40063
NumStgs	40065
NumThrws	40067
OEM ID	40069
MechEff	40059
MaxDischF	40057
RelHumid	40077

KrnlBsy	40275
WDPulse	40277
NxtLSUP	40279
NxtLSDN	40281
NSU_Perc	40283
NSD_Perc	40285
MinRPM	40287
MaxRPM	40289
MinPs1	40291
MaxPs1	40293
OptmLS	40295
CurrTorq	40297
IsenEff	40299
LStoSetforSU	40329
AuxLoad	40331
BHPMax	40333
BHPMax at TorqSP	40335
Elevation	40337
AtmPress	40339
NumCyls	40341
NumLSs	40343
NumStgs	40345
NumThrws	40347
OEM ID	40349
MechEff	40351
MaxDischF	40353
RelHumid	40355

ERROR ARRAY:			
40101	to	40199	
LOAD ARRAY:			
40201	to	40299	
FLOW ARRAY:			
40301	to	40399	
STAGE ARRAY:			
40401	to	40651	
THROW ARRAY:			
40653	to	41171	
HE_CYL ARRAY:			
41173	to	41791	
CE_CYL ARRAY:			
41793	to	42411	

RANGES:	
PsMin	42413
PsMax	42415
PdMin	42417
PdMax	42419
RPMmin	42421
RPMmax	42423
TorqMin	42425
TorqMax	42427
AmbTempMin	42429
AmbTempMax	42431
TsStg1Min	42433
TsStg1Max	42435
TsStg2Min	42437
TsStg2Max	42439
TsStg3Min	42441
TsStg3Max	42443
TsStg4Min	42445
TsStg4Max	42447
TsStg5Min	42449
TsStg5Max	42451
TsStg6Min	42453
TsStg6Max	42455

ERROR ARRAY:				New unit provides 100 Load steps
Err.LS.01	40445	to	40643	instead of only 50 for older units
LOAD ARRAY:				New unit provides 100 Load steps
Load.LS.01	40645	to	40843	instead of only 50 for older units
FLOW ARRAY:				New unit provides 100 Load steps
Flow.LS.01	40845	to	41043	instead of only 50 for older units
STAGE ARRAY:				
Stage-Item.01	41045	to	41403	Sub Items use same indexes
THROW ARRAY:				
Throw-Item.01	41405	to	42003	Sub Items use same indexes
HE_CYL ARRAY:				
CylHE-Item.01	42005	to	42803	Sub Items use same indexes
CE_CYL ARRAY:				
CylCE-Item.01	42805	to	43603	Sub Items use same indexes

RANGES:	
Range-01	43605
Range-02	43607
Range-03	43609
Range-04	43611
Range-05	43613
Range-06	43615
Range-07	43617
Range-08	43619
Range-09	43621
Range-10	43623
Range-11	43625
Range-12	43627
Range-13	43629
Range-14	43631
Range-15	43633
Range-16	43635
Range-17	43637
Range-18	43639
Range-19	43641
Range-20	43643
Range-21	43645
Range-22	43647

eRCM Express™

Changes to Manual

by Version

From Version 2019-Jul-1 to Version 2019-Oct-29:

- Added support for Allen-Bradley protocol.
- Added descriptions for the eRCM Express Version feature, and for the Offline Validation Limits.
    - **New registers**: Common Comparison Items (40307..40313), eRCM Express Version Number (40357), Offline Validation (40359..40373).
- Updated references used in manual to show both Allen-Bradley tag name and associated Modbus register.
    - Example for Suction Pressure Input (PsG): `(ACI_Inputs[1]; Reg #40003).`
- Updated tables to show Allen-Bradley and Modbus references for all data items.

From Version 2018-Aug-20 to Version 2019-Feb-1:

- A few minor corrections and typos.
- Added notes/text for features new to Version 0.5.1 of the eRCM Express Firmware
- Checklist now added to shipping contents.

From Version 2018-Mar-5 to Version 2018-Aug-20:

- Multiple minor corrections.
- Adding of Addendums, I, II, III, and IV.
- Some additional notes/images for installation of device, for local testing and field use.
- Some small changes to pseudo code examples.
- Description of using timers in PLC rather than polling for **IsKernelBusy** register.

From Version 2018-Feb-16 to Version 2018-Mar-5:

- Update of general images and rewording of some text. No technical information changed.

eRCM Express™

ADDENDUM I

Load Step

Selection Modes

Reciprocating compressors allow for significant loading and unloading of the driver (engine or motor) based on actuating/de-actuating devices (such as clearance pockets and end deactivators).

At times, multiple hardware configurations may require the same power requirements but generate different flow rates. As such, at times configuration X will require less load than configuration Y, but then at a different compression ratio configuration X can require more load than configuration Y. When this happens, it is commonly known as “curve-crossing”.

Ideally, when a PLC requests NSU (Next Step Up) it will be a load step that will increase load (or increase flow) on the unit, not decrease load (or flow). Thus, when curve-crossing is present, the complexity of selecting the NSU and NSD increases somewhat.

Furthermore, end users do not generally desire lots of unloading devices to engage at once. If the control lines have insufficient actuation pressure, then the result can be multiple devices actuating late (2 seconds later after others). This can cause the dynamic load on the driver to jump up and down. Often, some engines cannot handle too much jumping of load and will shut down. Thus, part of picking the NSU and NSD may involve consideration for load step efficiency, and number of devices engaging at the same time.

Different types of unloading sequences relative to the unit’s operating map are better handled by different load step selection methodologies. One method does not handle all unit arrangements. Thus, eRCM Express provides multiple methods for load step selection. The Applications Engineer that created the eRCM Viewer model for use in the eRCM Express should have reviewed the load steps, the operating map, and any concerns, and subsequently set the model to use the best load step selection method for it.

Currently, eRCM provides nine (9) load step selection modes. Each one has its own pros and cons.

Option #31 and Option #41 cannot be changed via the PLC, as these modes are for specific hardware implementations.

When in doubt, set unit to **Load Step Selection Mode** to **3**, and set the **MinLoadFlowPercChange** to **1** (1%). These settings are good, albeit not necessarily most efficient, for most (but not all) models.

eRCM Provides the following Load Step Selection Modes:

Option 1: Flow with Optimization

Description: Selects most efficient load steps based on a minimum flow rate change between load steps.
Pro(s): This is usually the best way to control a reciprocating compressor.
Con(s): May skip load steps (at times potenitally too many load steps) and cause too many unloading devices to actuate at once.

Option 2: Flow without Optimization

Description: Selects load steps based on a minimum flow rate change between load steps.
Pro(s): Skips unnecessary steps as needed based on a minimum flow rate change.
Con(s): May increase (decrease) flow rate while simultaneously decreasing (increasing) load.

Option 3: Flow based on following the Load Step Sequence

Description: Selects load steps based on following load step sequence until a positive and a negative flow rate are identified.
Pro(s): Faithfully follows load step sequence, and only skips steps as needed based on safety and increasing/decreasing flow rates.
Con(s): May select inefficient load steps, and/or cause unit load to jump up and down.

Option 11: Load with Optimization

Description: Selects most efficient load steps based on a minimum load change between load steps.
Pro(s): This is a very common way to control a reciprocating compressor.
Con(s): May skip load steps (at times potenitally too many load steps) and cause many unloading devices to actuate at once.

Option 12: Load without Optimization

Description: Selects load steps based on a minimum load change between load steps.
Pro(s): Skips unnecessary steps as needed based on a minimum load change.
Con(s): May increase (decrease) load while simultaneously decreasing (increasing) flow rate.

Option 13: Load based on following Load Step Sequence

Description: Selects load steps based on following load step sequence until a positive and a negative load change are identified.
Pro(s): Faithfully follows load step sequence, and only skips steps as needed based on safety and increasing/decreasing loads.
Con(s): May select inefficient load steps, and/or cause unit flow rates to jump up and down.

Option 21: Follow Load Step Sequence

Description: Simply selects NSU and NSD based strictly on the load step sequence, skipping steps only due to safety.
Pro(s): Ideal for units with well-defined, well-spaced, no curve-crossing of load steps, and very few safety issues limiting load steps.
Con(s): Can cause unit to jump up and down in load and/or flow if load steps and load step sequence are not well-defined (see above).

Option 31: Use when ALL unloading is via parallel Automated Variable Volume Clearance Pockets (VVCP)

Description: When safe, NSU and NSU will always differ by one (1) from the current load step.
Pro(s): NSU and NSD are always one load step away (when safe). Allows for easy control of units using aVVCP on all cylinders with unloading.
Con(s): Do NOT use this option unless unit unloading meets the specifications reqiuired for Option 31. *** Mode CANNOT be changed via PLC. ***

Option 41: Use when ONLY one (1) Automated VVCP is used with Fixed Clearance and End Deactivation Unloading Devices

Description: Tries to adjust aVVCP first. If either NSU or NSD not successful, then for that item leaves aVVCP as-is and selects complementing step.
Pro(s): Ideal option for handling multiple fast actuation devices (pockets and unloaders) with a slow actuating aVVCP.
Con(s): Do NOT use this option unless unit unloading meets the specifications reqiuired for Option 41. An aVVCP can lead to cases where unit's full load cannot be achieved (without unloading the aVVCP first). *** Mode CANNOT be changed via PLC. ***

eRCM Express™

ADDENDUM II

Notes for Manual

Mode Operations

Most PLCs will provide a Manual Control Mode for controls. This mode is often critical when the unit is unhealthy, and needs locked in to a certain load step, when an analyst is collecting data and unit needs to hold load step and speed during tests, and when an operator needs to set the unit to a specific load step for review.

Nevertheless, even in Manual Control Mode, the eRCM Express should be consulted for unit safety. Operators should not be allowed to load/unload to any unsafe load steps. Also, if the current load step becomes unsafe, then the control panel must change to a safe load step (even in Manual Control Mode) or Shut Down.

If the Load Step Selection Mode is not the simple **Load via Load Step Sequence** method, then some safe load steps may potentially be skipped over. This is ideal for normal operations as inefficient or inappropriate load steps should normally be skipped over. However, this can be problematic if the operator's/analyst's goal is to set the unit on one of the load steps that is being skipped.

Thus, when the unit is put in Manual Control Mode, it may be prudent to change the Load Step Selection from the current method to **Load via Load Step Sequence**. In this mode, eRCM Express will select NSU and NSD (**N**ext **S**tep **U**p, and **N**ext **S**tep **D**own) strictly by following the defined load step sequence, only skipping load steps if they are unsafe. Keep in mind, that in this mode, the NSU can actually lead to "lower load". However, this method will generally allow the operator/analyst to achieve the load step they desire.

Upon leaving Manual Control Mode, reset the Load Step Selection Mode back to the original method of the unit – simply write a zero (0) to the Load Step Selection register. Otherwise, future data will not be appropriate for controls.

Overview:

- User opts to enter Manual Control Mode:
  - Write a "21" to LoadStepSelectionMode `(ACI_Inputs[52]; Reg#40105)` to tell eRCM Express that subsequent NSU and NSD values to be chosen via that method.
    - Method 21 is the generic **Follow Load Step Sequence** method, and as such, NSU/NSD will tend to follow the defined load step sequence in order, only skipping load steps due to safety issues.
    - This method is the one most likely to allow an operator/analyst to load to a particular load step and then stop it there for review/testing.
- User exits Manual Control Mode:
  - Write a "0" to LoadStepSelectionMode to tell eRCM Express to return to its original method for determining NSU and NSD values.

eRCM Express™

ADDENDUM III

Allowing Load Steps

With Non-critical

Errors

If a reciprocating compressor has a lot of clearance pockets, then at medium to higher compression ratios, it is not uncommon for many of the lower-load load steps to be clipped due to Low Volumetric Efficiency issues.

Thus, trying to bring the unit online, or even performing a normal stop, can be troublesome as the power change on the engine or motor between safe load steps may be too large. eRCM Express will not select safe Next Steps Up/Down if they result in too large of a load change on the driver.

However, certain load steps, such as those determined to have Low Volumetric Efficiency (LowVE) issues and/or Predicted High Discharge Temperature (HiTd) issues, can be used as transitional load steps. That is, it is generally ok to transverse through these types of load steps, but the unit cannot run (cumulatively) in them for more than a few minutes.

- Some units have experienced serious cylinder damage due to running in LowVE after only about ten (10) minutes. Other units run in LowVE areas for hours and days without issues (other than cylinder getting hot, but not too hot). Thus, the amount of safe time needs to be determined by the end-user.
    - Generally, running in LowVE less than 2-3 minutes is not an issue.

Thus, for some end users, they may need to set the **IgnoreNonCriticalErrors** register `(ACI_Inputs[57]; Reg#40115)` to a 1 during Normal Start Up and Normal Stopping. This allows the unit to load smoother, albeit it may select certain load steps that put the unit in LowVE or HighTd. However, if the total time for these events is short (0-3 minutes), then no real concerns are raised.

- After the end-user's specified time limit, if loading/unloading goals are not achieved, then **IgnoreNonCriticalErrors** register must be set back to 0. If no safe load steps are then determined, then Shut Down is the prudent action to take.

Alternative use of **IgnoreNonCriticalErrors**:

- If during normal operations, OptimalLS ≠ -1 and OptimalLS ≠ CurrLS, but NextStepUp = -1, then this means that there is an opportunity to load the unit up more, but the potential load change from CurrLS to OptimalLS is too large to take.
    - By electing to ignore some non-critical safety issues for a few minutes, the unit may be able to step though some transitional load steps to reach OptimalLS.

eRCM Express™

ADDENDUM IV

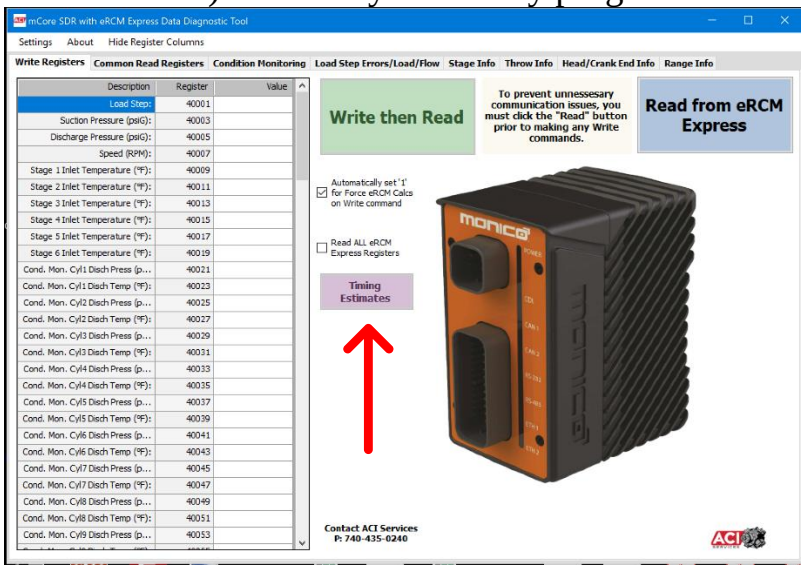
Determining

Calculation

Times

To make sure the PLC is waiting long enough to allow the eRCM Express to calculate all performance data, do the following: ***(Only applies when using Modbus protocol.)***

1. Connect a PC to the eRCM Express.
2. Run the eRCM Express Diagnostic Software on the PC.
3. Select the correct type of eRCM Express.
4. Specify which modeling file to use.
5. Select the option to run performance timings.
    a. Specify option for data block sizes: Minimum, Normal, Everything
        i. Minimum: Typical if only using the Quick List for Writes and Reads
        ii. Normal: Typical if limiting data based on number of load steps, number of stages, number of throws, and number of cylinders.
        iii. Everything: Reads back **all** data registers. If using a gateway (ProSoft Card or Red Lion) then many users may program those to read back all data.



6. Wait until performance timings are complete.
    a. Average Time to Send Write Data to eRCM Express
    b. Average Time to Calculate Results within eRCM Express
    c. Average Time to Retrieve Read Data back from eRCM Express
7. In the PLC, after writing the data to eRCM Express, wait "*Average Time to Calculate Results*" plus 50 ms before reading data back from the eRCM Express.
    a. For most units, this will likely be around 100-150 ms.
    b. For more complex units, this may be in the 150-350 ms range.
8. If there are multiple models in the eRCM Express, then each one should be checked separately. The PLC code might then use a common wait time for all models (if they all have similar times), or set the wait time uniquely per model (if their times tend to differ a lot).

eRCM Express™

ADDENDUM V

mCore Operations

Manual

Tel 281•350•8751 Fax 888•418•5955
www.monicoinc.com

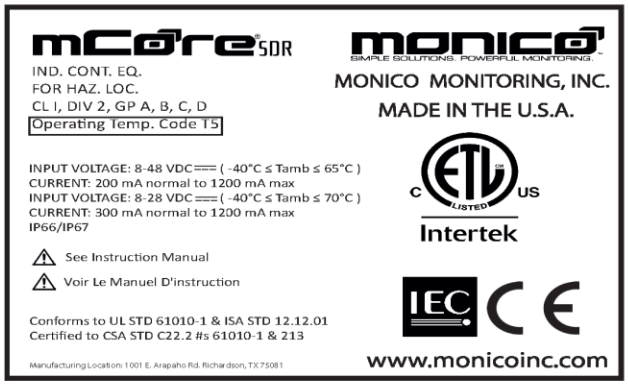
mCore® SECURE DATA ROUTER - ADVANCED PROTOCOL CONVERTER

Thank you for purchasing mCore® SDR (Secure Data Router). Your device comes preconfigured and ready to install.

Prior to use, thoroughly read the instructions in this manual to connect and use this product correctly. Please retain this manual for future reference and make sure that this manual is available to all users. To ensure the safety and proper operation of the device and any connected equipment, installation or relocation should be performed by qualified personnel only.

IND. CONT. EQ.
FOR HAZ. LOC.
CL I, DIV 2, GP A, B, C, D
Operating Temp. Code T5

Conforms to UL STD 61010-1 & ISA STD 12.12.01
Certified to CSA STD C22.2 #s 61010-1 & 213



SAFETY PRECAUTIONS

Thoroughly read and follow the safety precautions and operating instructions listed in this manual before using the product. After reading, retain this manual for future reference.

- Do not use or mount the product in any manner or location not intended.
- This manual should be available for anyone operating, repairing or relocating the product.
- The product should be installed, repaired or removed by qualified personnel only.
- Do not disassemble or modify the product. Improper installation or repair may cause injury, damage, electric shock or fire.
- To ensure proper operation and avoid damage to the product, use appropriate tools along with recommended connectors and recommended wire gauge.
- Do not open or break the seal between the housing and the connector header.
- Do not operate a unit that has been damaged.
- Extended environmental conditions: wet location, outdoor use, ambient temperature - 40°C and 70°C, and altitude up to 2000m.

WARNINGS

WARNING – EXPLOSION HAZARD. DO NOT CONNECT OR DISCONNECT WHEN ENERGIZED.
AVERTISSEMENT – RISQUE D'EXPLOSION. NE PAS BRANCER OU DÉBRANCHER LORSQUE LE CIRCUIT EST SOUS TENSION.
WARNING – EXPLOSION HAZARD. DO NOT DISCONNECT WHILE THE CIRCUIT IS LIVE OR UNLESS THE AREA IS FREE OF INGNITIBLE CONCENTRATIONS.
AVERTISSEMENT – RISQUE D'EXPLOSION. NE PAS DÉBRANCHER SI LE CIRCUIT EST SOUS TENSION, À MOINS QUE LE MILIEU SOIT LIBRE DE SUBSTANCES INFLAMMABLES CONCENTRÉES.
WARNING – EXPLOSION HAZARD. DO NOT REMOVE OR REPLACE BATTERIES OR PLUG-IN MODULES (AS APPLICABLE) UNLESS POWER HAS BEEN DISCONNECTED OR THE AREA IS FREE OF IGNITIBLE CONCENTRATIONS.
AVERTISSEMENT – RISQUE D'EXPLOSION. NE PAS RETIRER OU REMPLACER DES BATTERIES OU DES MODULES DE PLUG-IN (COMME APPLICABLES) À MOINS QUE LA PUISSANCE A ETE DÉCONNECTEE OU LA ZONE EST LIBRE DE CONCENTRATIONS IGNITIVE.
WARNING – DO NOT REMOVE THE AMPSEAL CONNECTORS UNLESS THE EQUIPMENT NEEDS TO UNDERGO REPAIR OR MAINTENANCE. AFTER REPAIR OR MAINTENANCE HAS BEEN DONE, PLEASE CHECK THE BARE PINS ON THE EQUIPMENT TO ENSURE THERE IS NO DUST OR WATER PRESENT. IP RATING OF THE EQUIPMENT IS ONLY MAINTAINED WITH THE AMPSEAL CONNECTORS ATTACHED TO THE EQUIPMENT.
AVERTISSEMENT - NE RETIREZ PAS LES CONNECTEURS AMPSEAL À MOINS QUE L'ÉQUIPEMENT NE SOIT RÉPARÉ OU ENTRETENU. UNE FOIS LA RÉPARATION OU L'ENTRETIEN EFFECTUÉE, VÉRIFIEZ LES BROCHES NUES SUR L'ÉQUIPEMENT POUR VOUS ASSURER QU'IL N'Y A PAS DE POUSSIÈRE OU D'EAU PRÉSENTE. LA CLASSIFICATION IP DE L'ÉQUIPEMENT EST UNIQUEMENT ENTRETENUE AVEC LES CONNECTEURS AMPSEAL ATTACHÉS À L'ÉQUIPEMENT.
WARNING – TO MAINTAIN CLASS 1, DIVISION 2 RATING, THE UNIT MUST BE INSTALLED IN A TOOL-SECURED ENCLOSURE USING ONE OF THE NEC WIRING METHODS THAT IS OUTLINED IN THIS MANUAL.
AVERTISSEMENT - POUR MAINTENIR L'INDICE DE CLASSE 1, DIVISION 2, L'UNITÉ DOIT ÊTRE INSTALLÉE DANS UN BOÎTIER SÉCURISÉ EN UTILISANT L'UNE DES MÉTHODES DE CÂBLAGE NEC DÉCRITES DANS CE MANUEL.

FOR CONFIGURATION AND OR INSTALLATION ASSISTANCE PLEASE CONTACT MONICO AT 281-350-8751 – OPTION 1

GETTING STARTED

mCore® SDR is designed for industrial applications requiring protocol translation between CDL (CAT® Data Link), S.A.E. J1939, Modbus RTU, and Modbus TCP. Each unit comes to you preconfigured and ready to install.

The mCore® SDR is easy to mount, with two mounting options, and environmentally sealed to protect against dust ingress and temporary immersion in up to 1 meter of water (IP66 & IP67). It was designed specifically for industrial mobile, mining, marine, and off-road applications. The device displays LED indicating lights configured to provide positive confirmation of power, connectivity, and data transmit/data receive.

Figure 1
Weight: 2.0 lbs. (.907 Kg)

CONNECTING MCORE

The unit should be connected to the desired ports according to the pinout below using the recommended connectors shown in Table 1 on the last page of this manual.

The pin numbers are labeled on each AMPSEAL key connector to ensure proper connection.

Pin 1 Blank	Pin 2 Blank	Pin 3 Blank
Pin 4 Reset (+)	Pin 5 Reset (-)	
Pin 6 Key Switch	Pin 7 Battery (-)	Pin 8 Battery (+)

Pinout for Key Connector A

** Please carefully review the Input Power Requirements on the next page. **

Pin 1 RS485 (-)	Pin 2 RS485 (+)	Pin 3 CDL (+)	Pin 4 J1939 #2 Term	Pin 5 J1939 #2 Term	Pin 6 J1939 #1 Term	Pin 7 J1939 #1 Term	Pin 8 J1939 #1 Comm	Pin 9 Ethernet 1 Rx (+)	Pin 10 Ethernet 1 Rx (-)	Pin 11 Ethernet 2 Rx (-)	Pin 12 Ethernet 2 Rx (+)
Pin 13 RS232 Comm	Pin 14 RS485 Comm	Pin 15 CDL (-)	Pin 16 J1939 #2 High	Pin 17 J1939 #2 Low	Pin 18 J1939 #1 High	Pin 19 J1939 #1 Low	Pin 20 Ethernet 1 Tx (+)	Pin 21 Ethernet 1 Tx (-)	Pin 22 Ethernet 2 Tx (-)	Pin 23 Ethernet 2 Tx (+)	
Pin 24 RS232 Tx	Pin 25 RS232 Rx	Pin 26 CDL Comm	Pin 27 J1939 #2 Comm	Pin 28 Blank	Pin 29 Blank	Pin 30 Blank	Pin 31 Blank	Pin 32 Blank	Pin 33 Blank	Pin 34 Blank	Pin 35 Blank

Pinout for Key Connector B

Pinout is depicting the female header connectors and respective key orientations on your mCore® SDR that will mate to the male connectors shown in Table 1. Be sure to follow the AMPSEAL "How-to Instructions" located at:

laddinc.com > resources > how-to-instructions > ampseal-connectors

laddinc.com > resources > how-to-instructions > ampseal-16-contact-crimping

These instructions include steps for proper inserting, crimping, and removing of wires into the pin connectors. For proper hand crimping, an AMPSEAL 2119118-1 hand crimping tool (not included) must be used.

mCore® SDR is approved for Class I Div 2 Group A, B, C and D. In order to maintain this rating the unit must be installed in a separate tool secured enclosure and comply with the one of the following NEC wiring methods:

1. Extra-hard usage cord – Section 501.140 of the NEC and Rule J18-160 of the CE Code Part I;

2. Instrumentation tray cable (Type ITC or CIC) – Section 501.10(B) and 501.105(B)(6) of the NEC and Rule J18-152 of the CE Code Part I;

3. Power-limited tray cable (Type PLTC) – Article 725 of the NEC; and

4. Tray cable (Type TC) installed per Article 336 of the NEC and Rule 12-2202 of the CE Code Part I.

FOR CONFIGURATION AND OR INSTALLATION ASSISTANCE PLEASE CONTACT MONICO AT 281-350-8751 – OPTION 1

INPUT POWER REQUIREMENTS

mCore® SDR power is supplied through Pins 7 and 8 on Key Connector A (see Table 1). Pins 7 and 8 can be located on the Pinout Guide under "Connecting mCore". The input voltage requirements are 8–48 VDC for operating temperatures between -40°C and +65°C. The input voltage requirements are limited to 8–28 VDC for operating temperatures between -40°C and +70°C. mCore® SDR has internal reverse polarity protection, but will not operate under reverse polarity conditions.

In addition to the battery positive and battery negative leads, there is a key switch wire that must be wired into the positive terminal of the power source or wired to a switch connected to the positive terminal of the power source.
mCore® SDR will not boot if key switch is not powered.

MOUNTING mCORE®

Surface-Mount: The unit should be mounted securely against a flat surface, using two ¼" fasteners (not provided), to a suitable location as close to the engine as possible. DO NOT mount directly to the engine block. The unit should be mounted, as shown in Figure 1 on page 2, either horizontally or vertically. Horizontal is the optimal mounting orientation considering LED visibility and heat dissipation. However, other mounting orientations are acceptable.

DIN Rail Mount: The mCore® SDR unit should be mounted vertically on a horizontal DIN rail. Tilt the unit to a 45-degree angle and insert the top lip of the DIN rail bracket onto the DIN rail. Then attach the bottom lip to the DIN rail to snap the unit into place (illustrated in Figure 2). A minimum of 1" of space should be maintained on all sides of the unit to ensure proper heat regulation.

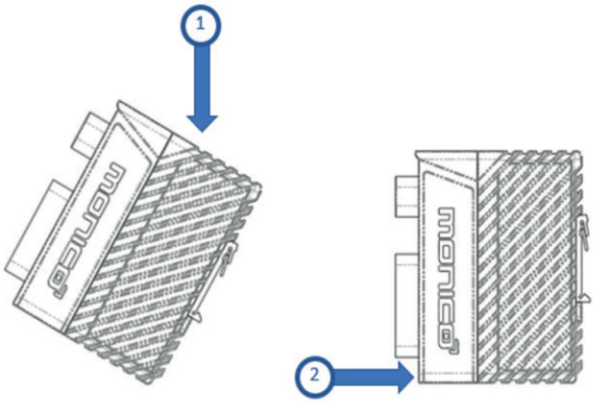


Figure 2

To remove the unit from the DIN rail, push down on the unit until the bottom lip is clear from the rail. Then pull out from the bottom. The unit should remove with ease.

GROUNDING mCORE®

After mounting and connecting mCore® SDR, the unit must be grounded in compliance with local and national electrical codes. It must be externally grounded using a customer-supplied ground wire before any power is applied. Contact the appropriate electrical inspection authority if you are uncertain that suitable grounding is available.

TOOLS REQUIRED:

QTY: 1	Grounding Lug (included)
QTY: 1	6-Gauge Ground Wire (not included)
QTY: 1	Nut Driver (not included)
QTY: 1	Pliers or Crimping Tool (not included)

PROCEDURE:

Step 1: Use the Pliers or Crimping Tool to crimp the 6-Gauge Ground Wire to the Grounding Lug.
Step 2: Connect the Grounding Wire to the mCore™ SDR Grounding Lug connection point, shown below in Figure 3.
Step 3: Place the Grounding Wire over the Grounding Lug and tighten these components using a nut driver. Tighten to 9.6 in-lbs.
Step 4: Connect the other end of the wire to a reliable earth ground if possible. For most effective grounding, use the grounding standards listed below.

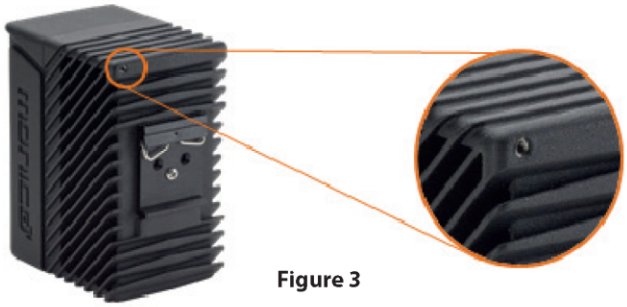


Figure 3

* Safety guidelines for proper grounding are outlined in OSHA Standard 1926.962.
* National standards for proper grounding are outlined in IEEE Standard 142.

POWER DISCONNECT

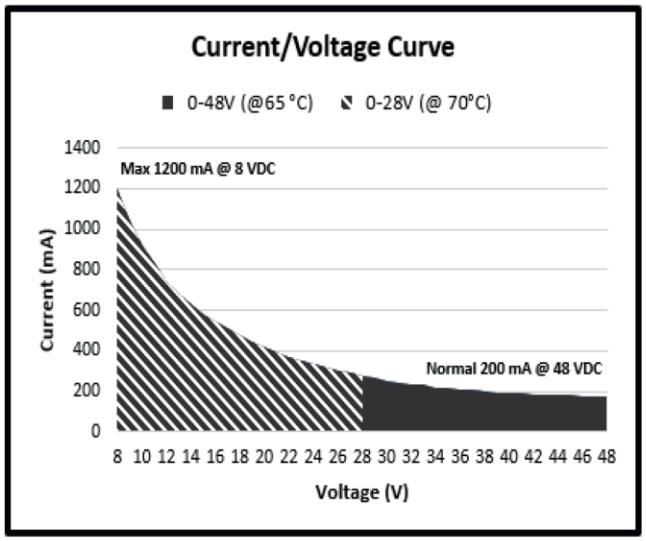
To disconnect power to mCore® SDR lift the locking ear on the side of Key Connector A using a flat blade screwdriver and pull gently. The locking ear is shown below in Figure 4.

Failure to protect mCore® SDR or use in any manner not specified may result in damage.

Figure 4

FOR CONFIGURATION AND OR INSTALLATION ASSISTANCE PLEASE CONTACT MONICO AT 281-350-8751 – OPTION 1

CURRENT/VOLTAGE REQUIREMENTS



** User is required to install a 2 Amp fast blow fuse rated for at least 50VDC (or the maximum voltage of the supplied power) to protect against short circuit. **

Table 1

mCore Mating Connector Options (*AMPSEAL*)

Item	Description	AMPSEAL Part Number
	Plug, Key Connector A	776286-1
	Plug, Key Connector B	776164-1
	Sealing Plug	770678-1
	Backshell	776463-1 and 776464-1
	Connector	770854-1

* Connectors are AMPSEAL available through LADD Industries

mCORE® LED INDICATOR GUIDE

	Solid Green	Blinking Green	Blinking Amber	Solid Red	Blinking Red	No LED
Ethernet 1	Connected	Passing Traffic, 100MB	Passing Traffic, 10MB	Not Connected		Not Configured
Ethernet 2	Connected	Passing Traffic, 100MB	Passing Traffic, 10MB	Not Connected		Not Configured
CDL		Receiving Packets as Expected	100% Passive Data	Bad Data/Reverse Polarity	No Traffic	
CAN 1	Good Connection	Valid Data		Bad Connection or No Data		Not Configured
CAN 2	Good Connection	Valid Data		Bad Connection or No Data		Not Configured
RS485	Good Connection	Valid Data		Bad Connection or No Data		
RS232	Good Connection	Valid Data		Bad Connection or No Data		
Power	Power Good; Setup Files Present	Power Good; No Setup Files			Controlled Shutdown State	

FOR CONFIGURATION AND OR INSTALLATION ASSISTANCE PLEASE CONTACT MONICO AT 281-350-8751 – OPTION 1

eRCM Express™

ADDENDUM VI

mCore's

Allen-Bradley

Handshaking

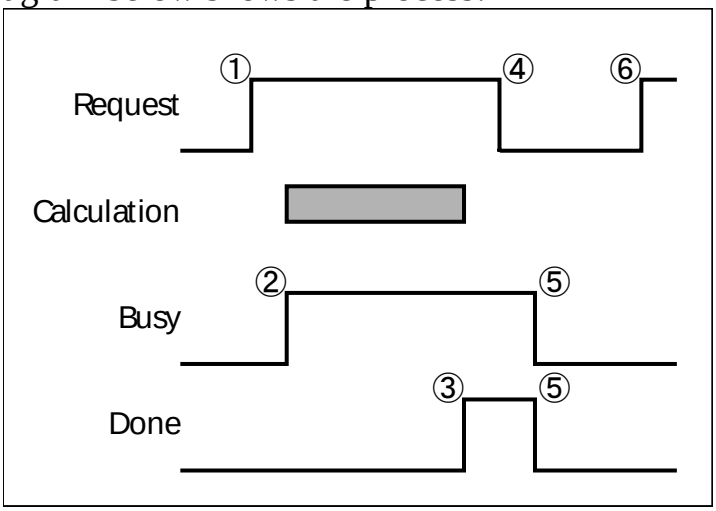
Allen-Bradley Handshaking

eRCM Express Software in the mCore device takes a set of inputs and calculates many outputs that provide optimization and performance information about a compressor. The module typically operates with a remote device that provides the inputs and consumes the outputs, and the interaction between the data transfers from the remote device and the mCore must be carefully managed to ensure that a coherent set of input data is used for the calculation and that a coherent set of output data is consumed by the remote device.

This management is performed by using three handshake values:

Name	Description	Controlled By	AB Tag Element
Request	Indicates that a new calculation is requested.	Remote Device	ACI_PLC_REQ[0]
Busy	Indicates that the calculation is in progress.	mCore	ACI_PLC_ACK[0]
Done	Indicates that the calculation has completed.	mCore	ACI_PLC_ACK[1]

The diagram below shows the process:



The numbered steps are as follows:

1. The remote device sets Request to start to the process.
2. The mCore notices Request is high and sets Busy, while at the same time starting the calculation.
3. The mCore completes the calculation and sets Done.
4. The remote device notices Done is high and clears the Request.
5. The mCore notices Request is low and clears both Done and Busy.
6. The remote device notices Done is low and can start the process again.

When mCore is acting as a master, such as in the AB Tag protocol, the mCore will read the ACI inputs from the remote device after Step 2 but before the calculation starts and write the ACI outputs before Step 3 but after the calculation completes.

eRCM Express™

ADDENDUM VII

Changing Protocols

Modbus / Allen-Bradley

Changing Protocols in an mCore

While mCore units provide a multitude of protocols, for eRCM Express functions, mCore units provide Modbus and Allen-Bradley protocols for sending and retrieving data. However, one of these must be set to be used. End users can always change from Modbus to Allen-Bradley, and vice versa, but the mCore unit will need to be reconfigured for that protocol.

For more details on how to set and use either the Modbus for Allen-Bradley protocol, reference **Section 9** of the Monico mCore Manual.