

DXM Configuration Tool v3

Instruction Manual

Original Instructions
201127 Rev. C
7 September 2018
© Banner Engineering Corp. All rights reserved



201127

Contents

1 DXM Configuration Tool	3
1.1 Banner Engineering Corp. Software Copyright Notice	3
1.2 Manually Install the DXM Driver	3
1.3 Update the Software	4
1.4 Save and Upload the Configuration File	4
2 Software Screens	6
2.1 Local Registers	6
2.1.1 Local Register Configuration	6
2.1.2 Modify Multiple Registers	9
2.1.3 Local Registers in Use	11
2.1.4 DXM Display	11
2.1.5 Protocol Conversion Overview	12
2.2 Register Mapping	12
2.2.1 Read Rules	12
2.2.2 Write Rules	14
2.2.3 Processing the Rules	15
2.2.4 J1939	15
2.2.5 Modbus TCP and Configuration	20
2.3 Action Rules	22
2.3.1 Threshold Rules	23
2.3.2 Register Copy Rules	24
2.3.3 Math and Logic Rules	25
2.3.4 Control Logic Rules	25
2.3.5 Trending and Filtering Rules	28
2.3.6 Tracker Rules	30
2.4 Scheduler	30
2.4.1 Create a Weekly Event	30
2.4.2 Create a One-Time Event	31
2.4.3 Create a Holiday Event	31
2.4.4 Create a Dynamic Schedule	32
2.5 Register View	32
2.6 Settings	33
2.6.1 General	33
2.6.2 Cloud Services	36
2.6.3 Logging	39
2.6.4 Scripting	41
2.6.5 Network	42
2.6.6 Mail and Messaging	44
2.6.7 Administration	45
2.6.8 Notes	46
2.6.9 I/O Board	46
2.6.10 Reprogram	52
3 Additional Information	53
3.1 Contact Us	53

1 DXM Configuration Tool

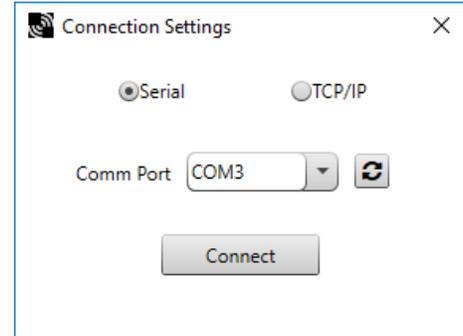
Configure the DXM using the DXM Configuration Tool. The DXM Configuration Tool can be used stand-alone or connected to the controller using USB or Ethernet. The tool creates an XML file defined for the DXM and can be used at the website level for configuration.

The DXM Configuration Tool restricts the naming of registers and rules to characters a-z, A-Z, 0-9, # \$ _ () space. A register name cannot end in a space. If an unacceptable name has been entered, a red ! displays.

Use the DXM Configuration Tool either while connected to a DXM or as a standalone configuration software.

There are two top-level menus similar to other Windows programs: **File** and **Device**.

- Use the **File** menu to manage the loading and saving of the XML configuration file created by the DXM Configuration Tool.
- Use the **Device** menu to handle the connection settings to the DXM as well as upload or download the XML configuration files. Also part of the **Device** menu is the communications tools used to view data traffic on the serial bus, useful in ScriptBasic programming.



Connect to the DXM by selecting the **Device > Connection Settings** menu. Connect via USB or Ethernet.

If connecting via Ethernet, set network parameters through the DXM LCD menu in the **System Cfg > Ethernet** menu. Network parameters can also be set within the DXM Configuration Tool. Setting parameters on the LCD menu overrides the parameters stored in the configuration file. To use the network parameters in the configuration file, reset the network parameters on the DXM LCD menu.

Banner recommends disconnecting the COMM port through the **Device** menu before turning off power or disconnecting the USB cable. Use **Device > Reboot** to restart the DXM if needed; the tool automatically disconnects the COMM port, then reconnect it again.



Tip: If connection attempts are failing (Application Status Icon in the footer of the tool is Red), close the DXM Configuration Tool and disconnect the USB cable from the computer. Reconnect the cable, launch the software, and attempt connecting again.

For a simple tutorial using the DXM Configuration Tool see the *DXM Controller Configuration Quick Start Guide*.

Right-click on most fields to access a context-sensitive menu. The most common options in the menus are **Cut**, **Copy**, **Paste**, **Delete**, and **Insert** to make data entry easier.

1.1 Banner Engineering Corp. Software Copyright Notice

This software is protected by copyright, trade secret, and other intellectual property laws. You are only granted the right to use the software and only for the purposes described by Banner. Banner reserves all other rights in this software. For so long as you have obtained an authorized copy of this software directly from Banner, Banner grants you a limited, nonexclusive, nontransferable right and license to use this software.

You agree not to use, nor permit any third party to use, this software or content in a manner that violates any applicable law, regulation or terms of use under this Agreement. You agree that you will not reproduce, modify, copy, deconstruct, sell, trade or resell this software or make it available to any file-sharing or application hosting service.

Disclaimer of Warranties. Your use of this software is entirely at your own risk, except as described in this agreement. This software is provided "AS-IS." To the maximum extent permitted by applicable law, Banner, its affiliates, and its channel partners disclaim all warranties, expressed or implied, including any warranty that the software is fit for a particular purpose, title, merchantability, data loss, non-interference with or non-infringement of any intellectual property rights, or the accuracy, reliability, quality or content in or linked to the services. Banner and its affiliates and channel partners do not warrant that the services are secure, free from bugs, viruses, interruption, errors, theft or destruction. If the exclusions for implied warranties do not apply to you, any implied warranties are limited to 60 days from the date of first use of this software.

Limitation of Liability and Indemnity. Banner, its affiliates and channel partners are not liable for indirect, special, incidental, punitive or consequential damages, damages relating to corruption, security, loss or theft of data, viruses, spyware, loss of business, revenue, profits, or investment, or use of software or hardware that does not meet Banner minimum systems requirements. The above limitations apply even if Banner and its affiliates and channel partners have been advised of the possibility of such damages. This Agreement sets forth the entire liability of Banner, its affiliates and your exclusive remedy with respect to the software use. You agree to indemnify and hold Banner and its affiliates and channel partners harmless from any and all claims, liability and expenses, including reasonable attorney's fees and costs, arising out of your use of the Services or breach of this Agreement (collectively referred to as "Claims"). Banner reserves the right at its sole discretion and at its own expense, to assume the exclusive defense and control of any Claims. You agree to reasonably cooperate as requested by Banner in defense of any Claims.

1.2 Manually Install the DXM Driver

If the DXM fails to connect over USB, first power down the DXM and unplug the USB cable from the PC, then follow these steps.

1. Verify the Windows update is configured to download device drivers.
 - a) On your Windows-based PC, go to the **Start** menu and select **Devices and Printers**.
 - b) Right-click the name of your computer, then select **Device installation settings**.
 - c) Select **Yes, do this automatically (recommended)**, then click **Save Changes**. If you're prompted for an administrator password or confirmation, type the password or provide confirmation. If Yes is already selected, click **Cancel** to close the dialog box.
2. Apply power to the DXM and plug the USB cable back into the computer. The computer should recognize the DXM. If not, go to step 3.
3. If the DXM is still unrecognized, follow these steps.
 - a) Open Windows File Explorer and browse to either: C:\Program Files (x86)\Banner Wireless\DXM Configuration Platform\Drivers or C:\Program Files\Banner Wireless\DXM Configuration Platform\Drivers.
 - b) Copy the files banner2001_cdc.cat and banner2001_cdc.inf into c:\Windows\inf.
 - c) From the **Start** menu, go to **Control Panel > Hardware and Sound > Device and Printers > Device Manager**. The DXM Controller should appear as an unknown device.
 - d) Right-click on the DXM icon and select **Properties**.
 - e) Click the **Driver** tab.
 - f) Click **Update Driver**.
 - g) Click **Browse my Computer** for the driver software.
 - h) Click **Let me pick from a list of device drivers on my computer**.
 - i) Select AT91 CDC class driver and click **Next**. The process is complete. If AT91 is not present, go to the next step.
 - j) Select **Show All Devices** and click **Next**.
 - k) The system drivers display. Under Manufacturer, search for Banner Engineering Corp.
 - l) Select the DXM Wireless Controller and click **Next**.
 - m) A warning message appears stating that installing this device driver is not recommended. Click **Yes**.
 - n) If the Device Manager lists a com port with the DXM Wireless Controller title, the driver successfully installed.

1.3 Update the Software

The software version displays in the lower right corner. If there is a red exclamation point in front of the version number, you are not running the most recent version of the software.

Follow these steps to update your software.

1. Click on the red exclamation point. An **Update Application** pop-up box appears, listing the newest version available.
2. Click on the version number link. The newest version of the software downloads to the download directory listed in your browser.
3. Close the software program. If you leave the software running while updating, you will need to reboot the computer.
4. If you have downloaded a ZIP file, unzip the file and double-click the EXE to install the software. If you have downloaded an EXE file, double-click the file to install the software.

1.4 Save and Upload the Configuration File

After making any changes to the DXM configuration, you must save the configuration files to your computer, then upload it to the DXM.

Changes to the XML file are not automatically saved. Any adjustments require that the file be manually saved before exiting the tool and/or before sending the XML file to the DXM.

1. Save the XML configuration file to your hard drive by going to the **File > Save As** menu.
2. Go to the **Device > Send XML Configuration to DXM** menu.
 - If the Application Status indicator is red, close and restart the DXM Configuration Tool, unplug and re-plug in the USB cable and reconnect the DXM to the software.
 - If the Application Status indicator is green, the file upload is complete.
 - If the Application Status indicator is yellow, the file transfer is in progress.

The DXM reboots and begins running the new configuration.



2 Software Screens

The following sections explain the function of each screen.

2.1 Local Registers

The main storage elements for the DXM are its Local Registers. The Local Registers store 4-byte values that result from register mapping, action rules, or ScriptBasic.

Since a Modbus register is only 16-bits, all transactions with Modbus devices use the lower 2-bytes (16-bits) of the Local Registers.

The Local Register characteristics are defined on the **Local Registers** screen of the DXM Configuration Tool.

- Local Registers 1–845 are 32-bit unsigned integer based registers.
- Local Registers 846–850 are special function registers that can be reset registers.
- Local Registers 851–900 are non-volatile registers with limited write capability for permanent data storage. Refer to the DXM Instruction Manual for more information.
- Local Registers 1001–1900 are 32-bit IEEE 754 floating point registers. Floating point values require two Local Registers to store a value. Floating point Local Registers are referenced on the odd numbered register addresses, 1001, 1003, 1005. When using Action Rules/ Read Rules always reference the odd numbered register addresses.

2.1.1 Local Register Configuration

Click on the arrow of the local register to expand the window to show all characteristics that can be configured with the tool.

The screenshot displays the 'Local Registers' configuration interface. At the top, there are navigation arrows and a range selector 'Registers from 1 to 20'. Below this is a table of registers. Register 1 is selected and expanded to show its configuration options:

- Display Information:** Units (None), Sign Type (unsigned), Scaling (None), LCD permissions (None).
- Constant Value and Timer:** Constant (0).
- Cloud Settings:** Cloud permissions (None).
- Logging and Protocol Conversion:** SD card logging (None), Protocol conversion (None).
- Register Group:** Group name (empty field).

Registers 2, 3, 4, and 5 are listed below with expandable arrows, all currently showing 'None' for Units and 'Group' for the Register Group.

The configurable characteristics are:

Display Information

Display parameters affect how the data is shown on the DXM controller LCD and on the website. The display parameters do not change the Local Register data and do not affect the way the Action Rules view the data. To modify the Local Register data use the scale and offset capabilities of the Read Rules. Remember that Local Registers 1–845 are integer based.

Units—Select the units information for the Local Register data. This parameter field is only used for display on the DXM LCD and for the website. Select the Custom field to create user defined units.

Sign Type—Select unsigned data format to show Local Register data as a positive integer. Select the Signed format to treat the data as a two's complement value, displaying value as positive and negative.

Scaling—The scaling parameter changes the viewing of Local Register data. Scaling is performed in two steps: applying a scale (one of five functions: add, subtract, multiply, divide, or modulo (remainder in division)), then adding an offset value. The step order can be changed to add offset first and then apply the scale by selecting **Apply offset before scale value**.

Scale Value is a positive or negative number used to perform a user select function with the Local Register value.

Scale Offset is a positive or negative number added to the Local Register value.

Apply offset before scale value—Changes the order of the scaling process. Typically, the scale value is applied first and then the scale offset. Selecting this checkbox reverses this process, scale offset applied first followed by the Scale value.

LCD permissions—This parameter defines whether the Local Register contents will display on the DXM's **Registers** menu.

None—The Local Register will not be shown on the DXM LCD

Read—The Local Register will be shown on the DXM LCD

Read/Write—The DXM can read and write the Local Register value from the LCD.

Constant Value and Timer

Constant—Forces the local register to be set to a user-specified value. This can be useful to compare values when using action rules.

Timer—A Local Register defined as a Timer increments every 100 ms, or 1 second. Write to the Local Register to start the timer at a specific value.

The values cannot be overwritten by the host system, Action Rules, Read Rules, or a ScriptBasic program.

Cloud Settings

Enables the cyclical reporting of this register value to the web site. The rate of the cyclical push of data is defined under the **Settings > Cloud Services** tab.

Set the **Cloud Permissions** flag to define how the local register is displayed and controlled at the web server. Settings are none, read, write, or read/write.

Logging and Protocol Conversion

Defines the onboard micro SD card storage of this local register based on a cyclical rate or an event change. Up to three different log files can be stored on the micro SD card, each with different logging definitions. The micro SD card logs are set up under the **Settings > Logging** tab. The **Protocol Conversion** section defines this local register to be an input or output when used with the different protocols enabled on this device. Ethernet/IP™ uses this register definition.

The DXM has the ability to store Local Register data on the internal micro SD card at a user defined cyclical rate. Log data can be retrieved manually or by E-mail. The internal micro SD card log parameters are defined under the Settings > Logging tab.

SD card logging—Defines the onboard micro SD card storage of this local register. Up to three different log files can be stored on the micro SD card, each with different logging definitions.

None—no logging selected

Log 1—Save this Local Register data to the Data Log 1

Log 2—Save this Local Register data to the Data Log 2

Log 3—Save this Local Register data to the Data Log 3

The Protocol Conversion - section defines this local register to be an input or output for Ethernet/IP™. ModBus TCP has access to all Local Registers and does not need to be defined.

EIP Originator -> DXM—For EtherNet/IP, data from the host PLC is written to the DXM Local Registers.

EIP DXM -> Originator—For EtherNet/IP, data from the DXM Local Registers to sent to the host PLC

Register Group

Group Name—A user-defined name that can be used to associate multiple Local Registers to each other. This can be used at a website or server to group information for visualization.

Reset

Local Registers 846–850 are reset registers. These registers reset a processor if the register is not written to within a specified time. The Reset register is typically used with ScriptBasic to detect if the program is in a bad state.

Within the time is specified, the DXM must be programmed to write to this register before this time limit is reached or the processor is reset.

Modify Local Registers Individually

This example configuration sets the device to push register values to the web server.

The screenshot shows the 'Registers from 1 to 20' configuration screen. The first register is selected, showing the following configuration:

Register	Units	Group
1	RH	Humidity
2	None	None
3	None	None

The configuration panel for Register 1 includes the following sections:

- Display Information:** Units (RH), Sign Type (unsigned), Scaling (Divide), Scale value (100.00), Scale offset (0.0000), Apply offset before scale value, LCD permissions (None).
- Constant Value and Timer:** None.
- Cloud Settings:** Cloud permissions (Read).
- Logging and Protocol Conversion:** SD card logging (None), Protocol conversion (None).
- Register Group:** Group name (empty field).

1. Go to the **Local Registers > Local Register Configuration** screen.

The local registers are a pool of integer and floating point registers that can be used by the controller to save data that was created, store data from external devices, or store basic constants. The local registers are where the user defines the data sent to the web server.

This screen shows how each local register is defined.

2. Click on the down arrow next to the first line for Local Register 1.
The parameters for Local Register 1 display.

- To push the register values to the web server, set the **Cloud Permissions** to read.

If the **Cloud Permissions** are set to Read, the web server only views data from the device and cannot write data to the device. If the permissions are set to Write, the web server only writes to the device and cannot read the data. If the permissions are set to Read/Write, the web server can read the data from the device and write to the device from the web.

- Set the **LCD Permissions** to **Read**.

This allows this **Local Register** data to be displayed on the DXM LCD under the **Registers** menu.

2.1.2 Modify Multiple Registers

To change multiple registers at once, use the **Modify Multiple Registers** screen.

Select a range of Local Registers by using Starting register and Ending Register. Then select which parameter fields to modify. Each parameter has one of three selections.

- None—no changes
- Reset—change to default settings
- Change—modify parameter. Other selections will appear based on the parameter.

Starting register

First register in the range of registers to change.

Ending register

Ending register in the range of registers to change.

Modify Properties

List of register properties that can be user-defined. The register properties include:

- Name
- Register group
- Constant or timer
- Units
- Sign type
- Scaling
- LCD permissions
- Cloud reporting
- Cloud permissions
- SD card logging
- Protocol IO
- Host timeout

Reset Form

Click to set the parameters on the **Modify Properties** form to None.

Change Registers

Click to save your changes and apply them to the selected register(s).

Modify Multiple Registers

Modify a range of registers from the **Local Registers > Modify Multiple Registers** screen.

1. Enter the **Starting register** and **Ending register**.
2. Select the value to change using the drop-down list next to each value.
3. Enter the new value in the field provided.
4. To push register values to the web server, set **Cloud Permissions** to read.

If the **Cloud Permissions** are set to Read, the web server only views data from the device and cannot write data to the device. If the permissions are set to Write, the web server only writes to the device and cannot read the data. If the permissions are set to Read/Write, the web server can read the data from the device and write to the device from the web.

- Click **Change Registers**.

2.1.3 Local Registers in Use

Scroll through only the **Local Registers** defined, get quick access to all parameter data for a Local Register and show where it is being used in the defined configuration. Also use this tab to verify Action rules are using the correct Local Registers.

#	Register Name	Register Group	Units	Signed	Constant or Timer	Cloud Permissions	LCD Permissions	Protocol Conversion	Log Files	Read Rules	Write Rules	Threshold Rules	Copy Rules
1	Humidity 1		RH	<input type="checkbox"/>	None	Read	Read	None	None				
2	Humidity 2		RH	<input type="checkbox"/>	None	Read	None	None	None				

Modify Registers Using the Local Registers in Use Screen

Use this screen to modify the parameters of any local registers being used.

#	Register Name	Register Group	Units	Signed	Constant or Timer	Cloud Permissions	LCD Permissions	Protocol Conversion	Log Files	Read Rules	Write Rules	Threshold Rules	Copy Rules
1	Humidity 1		RH	<input type="checkbox"/>	None	Read	Read	None	None				
2	Humidity 2		RH	<input type="checkbox"/>	None	Read	None	None	None				

- Go to the **Local Registers > Local Registers in Use** screen.
A list of the local registers being used displays. To add more Local Registers to this tab, name more Local Registers using the **Local Register Configuration** screen.
- Using the drop-down lists, change the desired parameters.
- To push register values to the web server, set **Cloud Permissions** to read.

If **Cloud Permissions** are set to Read, the web server only views data from the device and cannot write data to the device. If the permissions are set to Write, the web server only writes to the device and cannot read the data. If the permissions are set to Read/Write, the web server can read the data from the device and write to the device from the web.

2.1.4 DXM Display

Use the **Local Registers > Local Register Configuration** screen to define which registers can be displayed under the Register menu of the DXM. Use the **DXM Display** screen to adjust the display order of the **Registers with LCD Permissions**. If the display order is not specified, the registers display in numerical order.

Registers with LCD permissions

1	Analog IN 1
2	Discrete OUT 1
3	Analog IN 2
4	Discrete OUT 2
5	Humidity
6	Temp F

→

Display order

1	2	Discrete OUT 1
2	4	Discrete OUT 2
3	1	Analog IN 1
4	3	Analog IN 2
5	6	Temp F
6	5	Humidity

↑
↓

Remove

The Local Registers defined to be displayed on the LCD menu are shown in the left column.

- Select the Local Register by clicking next to the **Registers with LCD Permissions** number.
- Click the arrow button to populate the **Registers with LCD Permissions** in the **Display order** column.
- After defining the Local Registers to display, adjust the order using the up and down arrows to the right of the **Display order** column.

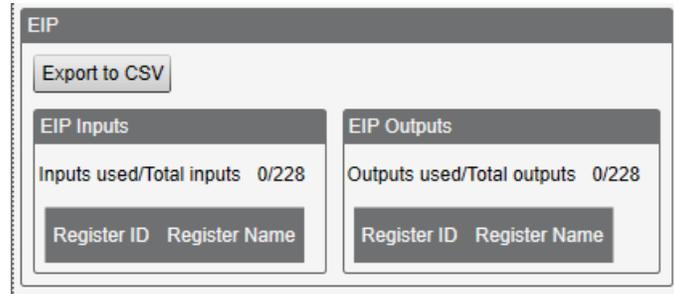
The display order can also be adjusted by selecting, dragging and dropping register within the Display order column.

If there are more Local Registers defined with LCD permissions than there are in the **Display order** column the remaining registers will be displayed in the order of the Local Register number. A Local Register can only be shown once on the LCD.

2.1.5 Protocol Conversion Overview

The **Protocol Conversion Overview** screen displays a list of the EitherNet/IP (EIP) Inputs and Outputs associated with DXM registers.

Define the Local Registers to be EtherNet/IP registers on the **Local Registers > Local Register Configuration** screen.



2.2 Register Mapping

Go to the **Register Mapping** screens to access and define read rules, write rules, J1939, and Modbus TCP.

Read and write rules allow the user to program the ability to read or write information from internal or external Modbus slaves to/from the local registers. On the DXM, use the read/write rules to access the Modbus registers of the LCD display, I/O base board, and the internal ISM radio.

The **Register Mapping** screens handle all direct register mapping. DXM Configuration Tool allows the user to create Write/Read rules that in turn create Modbus messages to external devices. How the user enters rules affects how a Modbus message is formed. If the user creates three individual read or write rules, those rules create three individual Modbus messages that will be sent out the RS-485 master port. If the user creates one read or write rule that spans multiple registers, the result is one Modbus message.

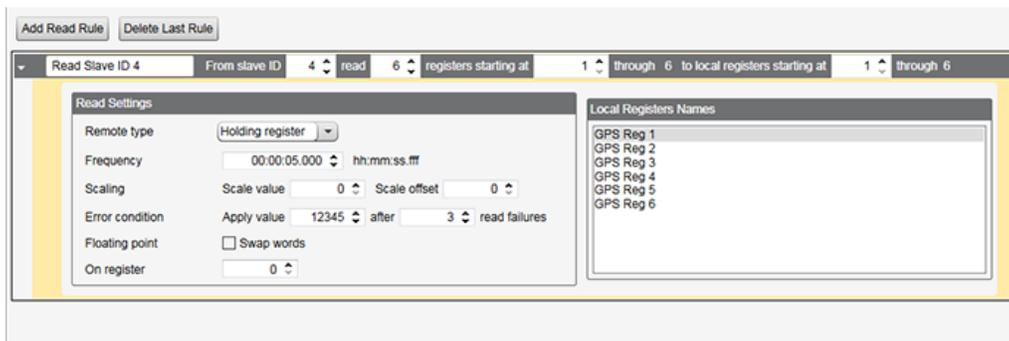
Click on the arrow of the read or write rules to show all the parameters for that rule.

2.2.1 Read Rules

Each read rule defines a Modbus slave ID and register range to read and then store in the Local Registers. The Local Register names shown are the registers that are being used by the read rule.



Important: Setting Read Rules or Write Rules to fast rates may cause other processes, ScriptBasic, LCD display, USB port accesses or API calls to be delayed or rejected.



The user defines parameters that can be applied to each read rule.

Remote Type

Select the register type from the drop-down list.

Read Holding register–16-bit register access, Modbus function 3; all Banner wireless devices use Holding Registers

Read Coil–Single bit access, Modbus function 1

Input Register–16-bit register access to read holding registers, Modbus function 4

Read Discrete Input–Single bit access, Modbus function 2

Frequency

Defines the cyclical rate at which the register is read. The maximum cyclic poll rate is controlled by the maximum polling rate set in the **Settings > General** screen, the **Modbus Master Communications** section. Changing that value causes the tool to adjust all polling rates. For example, setting the maximum rate to 1 second causes the tool to adjust all rules with a frequency of less than 1 second up to 1 second.

Offset and Scaling

Scale and Offset are applied to the Modbus data before writing to the Local Registers.

The Scale value is multiplied followed by adding of the offset value. Dividing can be accomplished by using scale values less than 1. Subtracting the offset value can be accomplished by using a negative offset value.

The result is held in the Local Register.

Apply Offset before Scale Value—The execution order for scaling values is changed to first add the offset value then multiply scale value. The result is held in the Local Register.

Error Conditions

Applies a default value to the Local Register after a user-defined number of Modbus register read fails.

Floating Point - Swap Words

A floating point value is a 32-bit value requiring two consecutive Modbus register reads or writes. The DXM expects the most significant part of the floating point value to be first (lowest address) followed by the least significant part. If a Modbus slave device sends the least significant part first, select **Swap Words** to align the words correctly.

On Register

The On Register is the address of a Local Register used to enable the Read rule. If the On Register is zero, it is not used. If the On Register is specified, the data value of the Local Register controls the executions of the Read rule.

When the value of the Local Register is zero, the Read Rule is idle. When the value of the Local Register is non-zero, it will continuously execute the Read Rule as fast as it can. This speed is governed the number of Read/Write rules defined and by the Modbus Master Communications parameter Maximum Polling Rate (default 50 ms). This parameter is located on the **Settings > General** screen.

The Frequency interval is not followed when using the **On Register** feature.

Slave IDs

DXM ISM Gateway radio: 1

DXM I/O Board: 200

DXM LCD: 201

MultiHop Remote Slave radios: 11 through 60

Create a Read Rule

Use the **Register Mapping > Read Rules** screen to create a new read rule.

1. Click **Add Read Rule**.
A new read rule is created.
2. Click the arrow next to the first rule to view the parameters.
The list of parameters opens.
3. Type in the rule's name in the 'none' field.
4. Select slave ID of the source device.
5. Select the number of registers to read from the source device.
6. Select the starting register of the source device.
7. Select the starting register of the local/target device.
8. Set the desired parameters.



Tip: Use fewer read rules for applications that require higher response times by taking advantage of the Alternative Modbus Register groupings for DX80 Performance Nodes as shown in the DXM Wireless Controller Instruction manuals.

2.2.2 Write Rules

The write rules write Local Register data to the defined Modbus slave ID and registers. The Local Register names shown are the registers used by the write rule.



Important: Setting Read Rules or Write Rules to fast rates may cause other processes, ScriptBasic, LCD display, USB port accesses or API calls to be delayed or rejected.

The screenshot shows the 'Add Write Rule' configuration window. At the top, there are buttons for 'Add Write Rule' and 'Delete Last Rule'. Below these are dropdown menus for 'None', 'Write', and '1' through '1', followed by 'to slave id' set to '0', and 'starting at register' '1' through '1'. The main area is divided into two panes. The left pane, titled 'Write Settings', contains:

- Remote type:** Holding register (dropdown)
- Frequency:** Cyclical (dropdown) with a time interval of 00:00:01.000 in hh:mm:ss.fff format.
- Scaling:** Scale value 0.000000 and Scale offset 0. There are checkboxes for 'Apply offset before scale value' and 'Swap words', both currently unchecked.
- Floating point:** A checkbox for 'Swap words' is also present.

 The right pane, titled 'Local Registers Names', is currently empty and contains the text 'None'.

The user-defined parameters that can be applied to each Write rule are:

Remote Type

Defines the type of Modbus functions to use

Write Holding register, Modbus function 16; all Banner wireless devices use Holding Registers

Write Multiple Coil, Modbus function 15

Write Single Holding register, Modbus function 6

Frequency

Defines how often to write the Local Register to the Modbus slave device in one of two ways:

Cyclical—Causes a Modbus write based on a timing interval, as fast as possible or a specified time interval.

On Change of Local Register Data—Allows the user to specify a certain criteria when to write to a Modbus slave device. For example, if the Local Register changes by a user-specified amount, the write occurs. If the user wants to write the Local Register to the Modbus slave at a minimum interval, use the **write AT LEAST** time setting. Use the **write AT MOST** time interval to minimize the write cycles for Local Registers that change frequently.

The maximum cyclic poll rate is controlled by the maximum polling rate set in the **Settings > General** screen, the **Modbus Master Communications** section. Changing that value causes the tool to adjust all polling rates. For example, setting the maximum rate to 1 second causes the tool to adjust all rules with a frequency of less than 1 second up to 1 second.

Offset and Scaling

Scale and Offset are applied to the Local Register data as it is being sent in the Modbus write command. Local Register data is not changed in the process.

The Scale value is multiplied followed by adding of the offset value. Dividing can be accomplished by using scale values less than 1. Subtracting the offset value can be accomplished by using a negative offset value.

The result is held in the Local Register.

Apply Offset before Scale Value—The execution order for scaling values is changed to first add the offset value then multiply scale value. The result is held in the Local Register.

Floating Point - Swap Words

A floating point value is a 32-bit value requiring two consecutive Modbus register reads or writes. The DXM expects the most significant part of the floating point value to be first (lowest address) followed by the least significant part. If a Modbus slave device sends the least significant part first, select **Swap Words** to align the words correctly.

Create a Write Rule

Use the **Register Mapping > Write Rules** screen to create a new write rule.

1. Click **Add Write Rule**.
A new write rule is created.
2. Click the arrow next to the first rule to view the parameters.
The user-defined parameters are displayed.
3. Type in the rule's name in the **Name** field.
4. Select the number of registers to write to the target device.
5. Select the starting register of the local/source device.
6. Select the slave ID of the target device.
7. Select the starting register of the target device.
The ending register automatically fills in based on the starting register and the number of registers selected in step 4.
8. Set the desired parameters.



Tip: Use fewer read rules for applications that require higher response times by taking advantage of the Alternative Modbus Register groupings for DX80 Performance Nodes as shown in the DXM Wireless Controller Instruction manuals.

2.2.3 Processing the Rules

Rules and functions are evaluated by the DXM in a specific order.

1. The read rules are executed first, beginning with the first rule defined and continuing in the order the rules were entered into the DXM Configuration Tool.
2. After the read rules are executed, the write rules are processed, in the same order.
3. After the write rules are processed, the DXM Configuration Tool starts over with processing the read rules.

The read/write rules take time to complete, not because they require processing power, but because each rule has a lengthy overall communication time relative to the processor execution cycle time. So, in parallel, the action rules are also solved.

Each action rule is processed in order, similar to the read/write rules. The groups of action rules are solved in this specific order:

1. Calculate (Math) rules are next, continually processed
2. Copy rules, processed only when a change of state is detected on the source register
3. Threshold rules, continually processed
4. Control Logic, continually processed
5. Trending, continually processed
6. Trackers, continually processed

2.2.4 J1939

J1939 is a wired communications protocol standard maintained by the Society of Automotive Engineers (SAE). Commonly used in off-road and heavy equipment, J1939 is a higher level protocol that is built upon the Controller Area Network (CAN) physical layer developed and originally published by Robert Bosch GmbH in 1991.

All J1939 messages are sent using the 29 bit message identifier, also known as the extended frame format, described in CAN 2.0B.

Typical J1939 data frames consist of the 29 bit identifier, 8 byte payload, a 15 bit CRC, and 20 additional bits used as flags, delimiters, or simply reserved for future use. Therefore a typical CAN packet is approximately $29 + 8 \times 8 + 15 + 20 = 128$ bits long. At the standard rate of 250 kbits/sec, a J1939 packet is a little over 500 μ s long. Actual duration may vary because the protocol defines that "bit stuffing" shall occur any time there are 5 bits in a row of identical polarity.

There is no centralized network coordination function in a J1939 network. Every node (known as an Electronic Control Unit, or EC) in a J1939 network "claims" a unique address for itself during network commissioning (after power-up). All ECs on the bus can be producers, consumers, or both producers and consumers of data. When an EC has data that it wants to write to the bus, it begins transmitting while simultaneously monitoring the channel for collisions. Collisions can be detected immediately because each bit is sent using a dominant/recessive scheme and any EC that sees a dominant bit on the channel when it transmitted recessive bit will know to immediately cease the transmission. This is known as non-destructive bus arbitration.

Note that a minimum of two nodes must be used to initialize communication on a CAN bus. Since a transmitted message must be acknowledged in the ACK bit by a receiver, the transmitting controller sends out an error flag if the message is not properly ACKed.

Messages in J1939 are organized by a field in the 29-bit header known as the Parameter Group Number or PGN. The PGN may be thought of as an index into a large data table. The J1939 standard defines what the data means in every such table entry, but also leaves many PGNs to be defined in a proprietary manner.

For example, the PGN 65262 is defined to contain Engine Temperature. Within the 8 bytes sent with this PGN, the J1939 specification also defines that byte 1 is for Engine Coolant Temperature, byte 2 contains Engine Fuel Temperature 1, bytes 3-4 encode Engine Oil Temperature 1, bytes 5-6 encode Engine Turbocharger Oil Temperature, byte 7 represents Engine Intercooler Temperature, and byte 8 is for Engine Intercooler Thermostat Opening.

Theory of Operation

The Banner DXM functions as a protocol converter between its internal local register space and J1939 PGNs.

Users must configure the DXM to select the J1939 PGNs to be monitored, which is known as creating a J1939 Read Rule. Similarly, the DXM can be configured to transmit messages if desired, which is known as creating a J1939 Write Rule. Read Rule and Write Rules map out the link between local registers and PGNs.

Data contained in DXM local registers may be accessed using the scripting engine in the DXM.

At startup, the DXM attempts to claim a unique J1939 address on the CAN bus it is connected to. It does so by sending its desired address to the PGN 0xEE00 and waiting for a reply. The DXM broadcasts the contents of its NAME field, which is specified in the DXM Configuration Tool using the J1939 Name fields. All sub-fields are currently user selectable except the Manufacturer ID.

The DXM allows two separate methods for specifying the address to claim on a J1939 network. Either one may be used; the choice of the method can be determined by whichever one is more convenient. In the first method, a list of acceptable addresses is specified. The DXM attempts to claim the first (leftmost) address, and if denied, will proceed to the next and the next. In the second method, a range of acceptable addresses is specified and the DXM attempts to claim the lowest, then increment that choice by 1 until it is able to claim one or reaches the end of the list.

No further J1939 functions (transmitting or receiving) begin until the DXM claims its address. In the DXM Configuration Tool, go to the **Register Mapping > J1939** screen and click **Read device J1939 address and name** to view the current DXM device address. If it returns -1, it was unsuccessful in claiming an address.

After the DXM has claimed an address, it passively listens to all J1939 traffic. Any messages that contain PGNs for which there is a matching read rule are decoded and the data is copied into the desired local registers. Any PGNs in the J1939 Write Rules area are transmitted onto the bus at the interval chosen. Each time a J1939 message is about to be transmitted, the DXM populates the J1939 data fields with the contents of the specified local registers, then writes the message onto the bus.

The acceptance criteria for J1939 Read Rules are:

1. The CAN protocol version must pass. This should always be the case in a J1939 network, but may have to be revisited if this device is used for other CAN bus applications.
2. The 16 bits that make up the arriving PGN are examined. Specifically the PGN is considered to be the concatenation of the 8 bit PDU Format and the 8 bit PDU specific fields.
 - a. If the PGN \geq 0xF000, then the message destination is global and the PGN is left unaltered as it is handed to the next layer of the internal stack. The implication of this is that read rules for PGNs in this range should have the PGN in the rule set to exactly what the device documentation says.
 - b. If the PGN $<$ 0xF000, then it is a peer-to-peer message and the PGN is considered to be the 8-bit PDU format bits followed by 8 bits of 0. Put another way, PGN = (PGN & 0xFF00), where & represents a bitwise logical AND operation.
 - a. Read rules for PGNs in this range should have the PGN in the rule set to mask off the lower 8 bits to zero.
 - b. In this case, the lower 8 bits (PDU specific bits) are the destination address. Destination address = PGN & 0x00FF.
 - c. If the address claimed by the DXM does not match the message's intended destination address, the message is discarded. To prevent messages from being discarded this way, select **Accept peer to peer messages sent to any address** in the DXM Configuration Tool.

Start Up Guide for the DXM

Verify your DXM is powered up and connected to a computer that has the DXM Configuration Tool installed.

The screenshot shows the 'J1939 Rules' configuration window. At the top, there are tabs for 'J1939 Configuration' and 'J1939 Rules'. The 'J1939 Name' section contains a table with columns: Arbitrary address capable (checked), Industry group (0), Vehicle system instance (0), Vehicle system (0), Function (0), Function instance (0), ECU instance (0), and Identity number (0). The 'J1939 Address and Comms' section has two radio buttons: 'Addresses to acquire' (unselected) and 'Find an address in the range' (selected). The 'Find an address in the range' section includes 'Start address' (0) and 'End address' (0) spinners. Below this is a 'BIP' spinner set to 25% and an unchecked checkbox for 'Accept peer to Peer messages sent to any address'. A 'Read device J1939 address and name' button is present, along with 'Current J1939 address -1' and 'DXM J1939 name 0000000000000000'. The 'J1939 Statistics' section has two columns of statistics, each with a 'Get' button. The left column shows: 'Number of transmits: 0', 'Read rules satisfied: 0', 'Messages passed through stack: 0', and 'Messages passed through CAN peripheral: 0'. The right column shows: 'Name claims: 0', 'Address claims: 0', 'Address fails: 0', and 'Address overrides: 0'. A 'Clear traffic and filtering totals' button is at the bottom left of the statistics section.

1. On the DXM: Make the wired connection to the bus. Using the labels on the housing decal, CH (pin 16) is for CANH and CL (pin 15) is for CANL.
2. On your computer: Launch the DXM Configuration Tool and navigate to the **Register Mapping > J1939** screen.
3. Configure the **J1939 Name** for the DXM.

All sub-fields of the NAME area are user-configurable except the Manufacturer Code, which is permanently set to 758 (decimal) as granted to Banner Engineering by SAE.

4. Specify the preferred addresses for the DXM to claim using one of two methods:
 - Create an explicit list of addresses in the **ID1** through **ID8** fields. The DXM attempts to claim the address under **ID1** first. If that address is already taken, it attempts to claim the address under **ID2** and continues until it claims an address.
 - List the **Start address** and the **End address**. The DXM attempts to claim the **Start address** first and increment sequentially until it successfully claims or reaches the **End address**.
5. **BIP** may be left at 25%. This field is the "babbling idiot protection" and establishes is the maximum fraction of the J1939 channel bandwidth that the internal stack allows the DXM to consume with data transmissions. It is a safeguard to protect the user from specifying too many or too frequent J1939 transmissions that could cause bus collisions and performance issues.
6. Select **Accept Peer to Peer messages sent to any Address** to allow the DXM to listen in on peer-to-peer messages that aren't directed to its own bus address.

Even though most messages on the J1939 bus are sent to a broadcast address, some peer-to-peer messaging is possible.

Create a J1939 Read Rule

Create a J1939 read rule to configure the J1939 to monitor for desired Parameter Group Numbers (PGNs) and capture their data.

These instruction assume the DXM is powered up and connected to a computer running the DXM Configuration Tool.

The screenshot shows the 'J1939 Read' configuration window. It has two buttons at the top: 'Add J1939 Read' and 'Delete Last J1939 Read'. Below the buttons is a table with the following columns: Name, Source Address, Local register, PGN, and Local register byte width. The current configuration is: Name: None, Source Address: 0, Local register: 1, PGN: None, Local register byte width: Two, 2.

1. From the **Register Mapping > J1939** screen, click **Add J1939 Read**.

2. Enter a name for the rule.

This name does not affect the actual behavior of the system.

3. Enter a **Source Address**.

- If the bus address of the EC sending the PGN is known, enter that address as the **Source Address**.
- To accept messages from any source, enter the wildcard value of 255 as the **Source Address**.

Every J1939 message includes a **Source Address**, which is the bus address of the device that sent the message. The DXM filters incoming messages based on this address.

4. Enter a **Local Register**.

The **Local Register** is the base address in local register memory space that the data should be sent to. When creating multiple read rules, choose to base addresses spaced so that the same local register does not get written to by multiple read rules.

5. Enter a **PGN**.

This is the parameter group number (PGN) of the J1939 message that this rule captures.

- If $PGN \geq 0xF000$ (61440), the PGN is a broadcast rule. The message arrives and passes through the stack as long as the PGN matches the read rule.
 - If $PGN < 0xF000$ (61440), it is a peer-to-peer message. To be received, the lower 8 bits of the PGN must match the bus address claimed by the DXM. (This check can be overridden by selecting **Accept peer to peer messages sent to any address**.) Because the lower 8 bits of the PGN (also known as the PDU Specific) are used as an address, the PGN in the read rule should have its lower 8 bits zeroed out to make the matching work.
6. Enter the **Local register byte width**.

Local registers are 32 bits, so it is possible to pack 1, 2, or 4 bytes of J1939 data into each local register. For J1939 messages with 8 bytes of data (the majority), packing means the data consumes 8, 4, or 2 local registers respectively depending on the choice.

Create a J1939 Write Rule

Create a J1939 write rule to transmit local register data in a J1939 message at a periodic interval.

These instructions assume the DXM is powered up and connected to a computer running the DXM Configuration Tool.

Name	Address	Local register	PGN	Local register byte width	Time between sends
None	0	1	None	Two, 2	55 ms

1. From the **Register Mapping > J1939** screen, click **Add J1939 Write**.

2. Enter a name for the rule.

This name does not affect the actual behavior of the system.

3. Enter a destination **Address** for the message.

The broadcast address is 255.

4. Enter the **Local register**.

The **Local register** is the data source's base address in local register memory space.

5. Enter the **PGN**.

Only PGNs in the range 0xFF00 (65280) to 0xFFFF (65535) are able to be specified by the manufacturer. Use caution when assigning data to PGNs outside of this range, because the data must adhere to the formatting specified by the Society of Automotive Engineers or risk being misinterpreted by other ECs on the bus.

6. Enter the **Local register byte width**.

Local registers are 32 bits wide, so it is possible to retrieve 1, 2, or 4 bytes of J1939 data from each local register. Because most J1939 messages have 8 bytes of data, the data is sourced from 8, 4, or 2 local registers, respectively, depending on the choice.

Example: Receive a PGN Using Broadcast Communication

A manufacturer's datasheet for an EC states it transmits its information on PGN 0xFF01 (65281).

Because this PGN is in the range 0xFF00–0xFFFF (65280–65535), the data is manufacturer-specific and not constrained by a definition controlled by the SAE.

1. On the **Register Mapping > J1939** screen, click **Add J1939 Read** to create a new read rule.
2. Enter a **Name**, for example, `Proprietary_msg1`.
3. Enter the **Source address** (the sender's bus address) or enter 255 to accept data from any address on the bus.
4. Enter the **Local register** (the base address for the data).
5. Enter a **PGN** of 65281.
6. Enter the **Local register byte width** (1, 2, or 4), which stores the bytes per local 32-bit register.

The following example J1939 message shows the relation between byte order on the bus and in the local register space:

	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8
J1939 Data	0x01	0x23	0x45	0x67	0x89	0xAB	0xCD	0xEF
Reg. Address	Base	Base +1	Base +2	Base +3	Base +4	Base +5	Base +6	Base +7
1 byte/reg	0xEF	0xCD	0xAB	0x89	0x67	0x45	0x23	0x01
2 bytes/reg	0xEFCD	0xAB89	0x6745	0x2301	(n/a)	(n/a)	(n/a)	(n/a)
4 bytes/reg	0xEFCDAB89	0x67452301	(n/a)	(n/a)	(n/a)	(n/a)	(n/a)	(n/a)

Example: Receive a PGN Sent Using Peer-to-Peer Communication

A manufacturer's datasheet for a sensor says it transmits its information on PGN 0xCBF4 (52212).

Because this PGN is in the range 0x0000–0xEF00, (0–61184) this is meant to be a peer-to-peer message. The lower 8 bits of the PGN contain the destination address.

1. On the **Register Mapping > J1939** screen, click **Add J1939 Read** to create a new read rule.
2. Enter a **Name**, for example, `p2p_msg2`.
3. Enter the **Source address** (the sender's bus address) or enter 255 to accept data from any address on the bus.
4. Enter the **Local register** (the base address for the data).
5. Enter a **PGN** of 51968 or 0xCB00. Note that the lower 8 bits have been masked.
6. Enter the **Local register byte width** (1, 2, or 4) bytes per local register.
7. Enter the Address claim. As a peer-to-peer message, this PGN is intended for the device at address 0xF4 (244) only. There are two ways to allow the DXM to read this message:
 - Have the DXM claim address 0xF4 (244) on the bus; or
 - Select **Accept peer to peer messages sent to any address**

Example: Monitor Diagnostic Troubleshooting Codes

The J1939 standard specifies a method of transmitting diagnostic trouble codes. Active trouble codes are sent using PGN 65226 or 0xFECA (consult the J1939 documentation for other related PGNs, such as Previously Active Trouble Codes).

1. On the **Register Mapping > J1939** screen, click **Add J1939 Read** to create a new read rule.
2. Enter a **Name**, for example, `active_DTCs`.
3. Enter the **Source address** or 255 to accept data from any address on the bus.
4. Enter the **Local register** (the base address for the data).
5. Enter a **PGN** of 65226 or 0xFECA.
6. Enter the **Local register byte width** (1, 2, or 4) bytes per local register.

Each diagnostic trouble code consumes 8 bytes, but there may be more than one active code from a given EC. Such longer messages are sent using the Transport Protocol feature of J1939, which the DXM supports. It is important that the user reserve enough local register space for the messages in case there are more than the standard 8 bytes.

Example: Transmit a PGN Using Broadcast Communication

To transmit manufacturer-specific data on PGN 65296 or 0xFF10, follow these steps.

1. On the **Register Mapping > J1939** screen, click **Add J1939 Write** to create a new write rule.
2. Enter a **Name**, for example, `tx_msg2`.
3. Enter the **Address** (the destination address for the message, for example 49d=0x31).
4. Enter the **Local Register** (the base address to retrieve the data from).
5. Enter a **PGN** of 61184.

6. Enter the **Local register byte width** (1, 2, or 4) per local register.
7. Enter a **Time between sends** of 1500 ms.

If this message is observed as a standard CAN bus message, the 29-bit arbitration ID appears as follows: 0x1CEF3116.

- The leading 1C is the priority, which has been set to the lowest, or 8.
- EF is the higher 8 bits of the PGN.
- 31 is the destination address.
- 16 is the address that this DXM has claimed on the bus.

Example: Transmit a PGN Using Peer-to-Peer Communication

To transmit manufacturer-specific data on PGN 61184 or 0xEF00, follow these steps.

1. On the **Register Mapping > J1939** screen, click **Add J1939 Write** to create a new write rule.
2. Enter a **Name**, for example, `tx_msg2`.
3. Enter the **Address**.
4. Enter the **Local register** (the base address from which to retrieve the data).
5. Enter a **PGN** of 65296.
6. Enter the **Local register byte width** (1, 2, or 4) per local register.
7. Enter a **Time between sends** of 1500 ms.

Send Diagnostic Commands

The DXM firmware supports some diagnostic commands that can be entered through the DXM Configuration Tool.

Use the buttons on the **J1939 Configuration** screen to send the following commands:

- Read device J1939 address and name
- Get traffic and filtering totals
- Clear traffic and filtering totals
- Get name and address claim totals

To manually send these commands, follow these instructions.

1. Go to the **Device > View Serial Activity** menu.
A **Traffic on port COMxx** window opens.
2. In the single-line text box at the bottom of the window, type a command. Click **SEND**.
The resulting serial traffic, including response, will be shown. The successful execution of CMD00xx always results in a response of RSP00xx, followed immediately by one or more parameters (xx indicates the Command Number).

Command (case sensitive, no spaces)	Command Name and Meaning	Sample Response	Fields in Response (after RSP00xx, where xx = confirmation of command number)
CMD0080	Address claimed	RSP008022	Decimal value of Address claimed by the DXM. For this example, 22.
CMD0081	NAME field	RSP00815A5ADA5EFFFFB4DA	ASCII string of the DXM's NAME field in hex. The NAME is eight bytes long, so the field will contain 16 hexadecimal digits.
CMD0082	Traffic and filtering	RSP00820,2,3,4	Totalizing counter. Number of Transmits (0), number of Read Rules satisfied (2), number of messages passed through stack (3), and number of messages passed through CAN peripheral of micro (4)
CMD0083	Statistics	RSP00830,1,2,3	Statistics on name and address claiming: Name, Address, Address fails, overrides
CMD0084	Clear traffic totalizer	RSP00840	Clears the totalizer values (traffic counting) that are accessed by CMD0082

2.2.5 Modbus TCP and Configuration

Use the **Register Mapping > Modbus TCP** screen to define Ethernet communication rules for the DXM to access remote register data from Modbus TCP servers. First define devices under the **TCP Configuration** tab, then define the Read/Write rules for the devices.

At the top of the window is an **Enable Modbus TCP** check box. Select this box to globally enable Modbus over TCP. The Start up delay defines the amount of time between the DXM power on and the execution of the Modbus TCP Read/Write rules.

The **Modbus TCP** screen defines

The screenshot shows the 'TCP Configuration' window with two tabs: 'Read rules' and 'Write rules'. The 'Devices' section contains two identical configuration blocks for 'Device 1' and 'Device 2'. Each block includes an 'Enable' checkbox, a 'Name' field, an 'IP Address' field (0.0.0.0), a 'Port' dropdown (502), a 'Poll rate' field (00:00:00.000) with '(HH:mm:ss.fff)' below it, and a 'Time out' field (00:00:00.000) with '(HH:mm:ss.fff)' below it.

To use Modbus TCP, follow these steps:

1. Select **Enable Modbus TCP**.
2. Enable a **Device** connection.
3. Configure the settings (IP Address, Port, Poll rate, and Time out).
4. Create and enable a rule.
5. Link the rule to an enabled Socket.

The Modbus TCP interface parameters are:

Enable Modbus TCP

Globally enables or disables Modbus TCP

Start up Delay

Defines the time delay between boot up and the execution of the rules

The individual Ethernet socket parameters are:

Enable

Must be selected to open the socket connection to the Modbus TCP server

IP Address

Defines the IP address of the Modbus TCP server

The format is xxx.xxx.xxx.xxx

Port

Defines the port to be used for the Modbus TPC protocol

Default: 502

Poll Rate

Defined how often to execute the write and read rules defined for this socket connection

Time out

How long the processor waits for an acknowledgement from the Modbus server before going on to the next rule

If Modbus TCP is disabled, this section is not copied into the final XML configuration file. If the Modbus TCP rules don't point to an enabled device, the rules are not written into the final XML configuration file.

Create a Modbus TCP Write or Read Rule

Use the **Register Mapping > Modbus TCP > Write Rules** screen to define each Modbus TCP register write transaction to a selected Modbus TCP server device. Use the **Register Mapping > Modbus TCP > Read Rules** screen to define each Modbus TCP register read operation to a selected Modbus TCP server device.

1. Enable Modbus TCP.

- Click **Add Write Rule** to create a new Write rule or click **Add Read Rule** to create a new Read rule.

- Name your rule. The name has no affect on the rule's operation.
- From the drop-down list, select the **Using device**. Device connections are defined on the **TCP Configuration** screen. A new write or read rule is created. Write and read rule parameters are written in a form of a sentence to make it easier to understand the Modbus operation.
- Select the number of Local Registers, the starting register, Modbus ID, and the starting target register.
 - Number of Registers**—For Write Rules, the number of internal Local Registers to write to the Modbus server. For Read Rules, the number of Modbus registers to read from the remote server device.
 - Local Register Address**—For Write rules, the Local Register Address is the starting address of the Local Register data. For Read rules, the Local Register Address is the starting address of the Local Registers to put the read data
 - Modbus ID**—An identifier also known as the unit number. If there are multiple devices at the server IP address, this field uniquely identifies the server. For other DXM devices this field will be 1. DXM devices to not respond to a unit number of 0.
 - Remote Modbus Register Address**—For Write Rules, defines where to write the data in the server. For Read Rules, defines the starting address of the Modbus registers to read from the remote server device.

The **Local Registers in Use** drop-down list shows all Local Register names associated to this write or read rule.

Click **Delete Last Rule** to remove the last created rule.

2.3 Action Rules

Action rules allow for simple logic functions and simple manipulation of local register data. The processing of an action rule is autonomous from other local register functions. Threshold rules provide the mechanism to create event-driven conditions, such as events to the cloud, local logs, or an email address. There are three groups that make up action rules.

Read rules are executed first, beginning with the first rule defined. The read rules execute in the order they were entered into the DXM Configuration Tool. After all read rules execute, the write rules are processed, in order. After the write rules are processed, the system begins again with the read rules.

Processing the read/write rules takes a long time to complete, not because they take a lot of processing power, but because each rule has a lengthy overall communication time relative to the processor execution cycle time. So, in parallel, action rules are solved. Each action rule is processed in order, similar to the read/write rules. The groups of action rules are solved in this specific order:

- Calculate (Math) rules are next, continually processed
- Copy rules, processed only when a change of state is detected on the source register
- Threshold rules, continually processed
- Control Logic, continually processed
- Trending, continually processed
- Trackers, continually processed



Important: Define all action rules to use integer-based Local Registers or floating point Local Registers. Do not mix the two Local Register types within an Action Rule. To move integer-based Local Registers 1-850 to floating point Local Registers, use the Register Copy rules.

2.3.1 Threshold Rules

A **Threshold Rule** triggers event messages sent to the cloud (event-driven push), triggers events to be stored to the local event log and creates a standard push message to the cloud with all defined registers being sent.

A Threshold rule creates a simple IF-THEN-ELSE comparison for a register to determine its value and set another register to indicate if the rule is true or false (you must select a true option to also set a false option). The definition section of the threshold rule sets the comparison and values. The definitions of the threshold rules can further be defined by the optional parameters, Hysteresis, On Time, E-mail/SMS and Logging options.

If selected, the false option (else condition) is applied. This creates an IF-THEN-ELSE structure. To remove the ELSE condition un-check the false option or set the result register to itself.



Important: Read Rules can scale incoming Modbus register values to make Threshold rules easier to understand. If Read Rule scaling is not used, the values stored in Local Registers are the raw Modbus values from the slave device. Display scaling and offset are not applied for these comparisons.

When local register select register select operation select Value/Register select value or register.

When TRUE, set local register select register to Value/Register select value or register.

Value/Register—When Value is selected, provide a fixed number in the following entry box. When Register is selected, provide a Local Register address in the following entry box. The contents of the Local Register will be used in the Threshold rule. When Same as Source is selected (else statement), the contents of the Local Register defined in the If comparison is used as the result.

Hysteresis

Optional parameter that is enabled only when values are non-zero. How hysteresis is applied depends on the comparison. For a test that becomes true 'if greater than,' the test will not return to false until the Local Register is less than the test value by a margin of at least this hysteresis value. If a test becomes true 'if less than,' it will not return to false until the Local Register is greater than the test value by a margin of at least this hysteresis value.

Minimum On Time and Minimum Off Time are time-based parameters that govern how long a statement must be true or false to activate the output register.

On Time

Stores in a Local Register how long (in minutes) the Threshold rule has been true.

The On Time value is shown in the Local Register in minutes but maintains the internal counter in seconds. This results in the accuracy of seconds, not rounding of the value stored in the Local Register.

After Trigger

After Trigger, set source to 0—After a Push is sent as a result of the Threshold rule. The source Local Register can be set to zero. This will indicate the Push data packet was sent.

Logging Options

Threshold rules create events. These events can cause certain actions to occur, such as:

- Sending threshold events to the cloud. An immediate push with the threshold register states is sent to the webserver when this rule becomes active.
- Saving the state to an internal event log. Go to **Settings > Logging** to set up the event log.
- Pushing data to the cloud when the threshold rule becomes active. All local registers with the Cloud Reporting parameter ON will push to the cloud. (**Local Registers > Local Register Configuration**)

E-Mail/SMS

When a Threshold rule becomes active or inactive the controller can send an E-Mail or SMS message.

Select which recipient should be notified when this rule changes state. SMS messages can only be sent when cellular is selected as the network interface. (**Settings > Cloud Services**).

Create a Threshold Rule

1. From the **Action Rules > Thresholds** screen, click **Add Threshold Rule**.
2. Enter the **local register** you are comparing.
3. Select the mathematical function and value.
4. Select what the system should do if the register value is true or false.
5. Select any other values, such as **Hysteresis**, **On Time**, or **Logging Options**.
6. Select any email or text messages you'd like sent upon a state transition.

2.3.2 Register Copy Rules

Use the **Register Copy** screen to copy a local register into another local register or range of registers.



Important: To move integer-based Local Registers 1–850 to floating point Local Registers, use the **Register Copy** rules. **Register Copy** automatically converts the integer value to a floating point value.



Important: Creating multiple copy rules using the same starting register will not be consistent. Instead create a daisy-chaining of registers. For example, copy register 1 to register 2, then copy register 2 to register 5, then copy register 5 to register 10.

Name	Copy Register	To Register	Through Register	Affecting the following registers
Analog In	1 <i>Humidity 1</i>	4	5	Humidity 1 Copy 1 Humidity 1 Copy 2

Add New Rule

Creates a new copy rule

Copy Register

Starting register of the data source range of registers

To Register

Starting register of the data target range of registers

Through Register

Ending register of the data target range of registers

Create a Register Copy Rule

1. From the **Action Rules > Register Copy** screen, click **Add New Rule**.
2. Enter the **Copy Register**. This is the first register for the source of the data.

3. Enter the **To Register** and **Through Register**. These are the registers for the beginning and end of the target range of registers.
 - If you are copying the contents of one register, enter the same register number into the **To Register** and **Through Register** fields.
 - If you are copying a range of registers, the range is defined by the **To Register** and **Through Register**. The **Copy Register** is the starting register of the source range.

2.3.3 Math and Logic Rules

Math/Logic rules deal with 32-bit register logic with results from 0 to 4,294,967,295. In contrast, **Control Logic** rules are binary rules, with the results being either 0 or 1.

Name	Operation	Local Register	And/Through	Local Register	Affecting the following registers	Store the result to	Local Register
New math rule	Average	8	Through	11	Temp 1 Temp 2 Temp 3 Temp 4	12	Average Temp

Some operations are valid for ranges of registers. Select **And** or **Thru** to select two registers or multiple registers in a range. **Average**, **Sum**, and **Logic Operations** are valid for ranges of multiple registers. The Local Registers are unsigned 32-bit integers. All math and logic functions can operate on all 32-bits. The Local Register operations are:

- **Add**—Adds two local registers and stores the result in a Local Register.
- **Average**—Averages the values of multiple registers and stores the result.
- **Divide**—Divides local register 1 by local register 2 and stores the result. Dividing by zero results in a zero.
- **Logic NOT**—Performs a bit-wise one's complement on a Local Register and stores the result.
- **Logic OR, AND, NOR, NAND, XOR**—Performs bit-wise logic function on multiple registers and stores the result. Bit-wise logic functions operate on all 32-bits of the Local Registers. To perform logic function on a simple 0 or 1 in registers, use the Control Logic rules.
- **Multiply**—Multiplies one local register by another and stores the result.
- **Sum**—Adds multiple contiguous local registers and stores the result in a Local Register.
- **Subtract**—Subtracts Local Register 2 from Local Register 1 and stores the result. For negative numbers, the results are in two's complement form.

Create a Math Rule

Use math rules to perform mathematical calculations on registers and store the results.

1. From the **Action Rules > Math/Logic** screen, click **Add Math Rule**.
2. Enter a **Name** for your new math rule.
3. Select the desired **Operation** from the drop-down list.
4. Select the starting **Local Register**, **And** or **Through**, and the second (when using **And**) or ending (when using **Through**) **Local Register** to perform your mathematical calculations on.
5. Select the **Local Register** you'd like to store your results in.

2.3.4 Control Logic Rules

Control Logic rules are binary rules, with the results being either 0 or 1. In contrast, the Math Logic rules deal with 32-bit register logic with results from 0 to 4,294,967,295.

Some Control Logic rules create memory elements using the DXM Local Registers as inputs and outputs. A value of zero in a Local Register is defined as 0 in the Control Logic rule, a non-zero value in a Local Register is defined as a 1 value. Control Logic rules are evaluated when an input value changes state. Memory elements include S/R Flip/Flop, D Flip/Flop, T Flip/Flop and J-K Flip/Flop

Control logic also works for floating point registers and virtual registers. In the case of floating point values, anything over 0.5 is considered logical 1.

Control Logic			
Function	In 1	In 2	Q out
AND	0	0	0
	1	0	0
	0	1	0
	1	1	1
OR	0	0	0
	1	0	1
	0	1	1
	1	1	1
XOR	0	0	0
	1	0	1
	0	1	1
	1	1	0

Control Logic			
Function	In 1	In 2	Q out
NOT	0	-	1
	1	-	0
NAND	0	0	1
	1	0	1
	0	1	1
NOR	1	1	0
	0	0	1
	1	0	0
NOR	0	1	0
	1	1	0
	0	0	1



Note: * Output Qn is an inverted output from the Q output and represents optional inputs or outputs. Leave the optional inputs or outputs connected to 0 when not used.

J-K Flip-Flop

If J and K are different, the output Q takes the value of J. The **Enable**, **Clock**, and **Qn** connections are not required for operation.

J-K Flip-Flop					
*Enable	J	K	* Clock	Q	Qn *
0	X	X	X	No change	No change
1	X	X	0 or 1 or ↓	No change	No change
1	0	0	↑	No change	No change
1	1	0	↑	1	0
1	0	1	↑	0	1
1	1	1	↑	No change	No change

If an input is zero it is false (0), and if a value is non-zero (anything but zero) it is considered true (1). All but one requires two inputs; the NOT rule only has one input. Some of the Control Logic rules provide basic non-memory logic functions: AND, OR, XOR, NOT, NAND and NOR. These basic logic functions are based on zero and non-zero values and result in a 0 or 1.

Optionally, the rules can have a **Clock** input and an **Enable** input. The **Clock** input determines the time (0-1 rising edge) the rule is evaluated. The **Enable** input enables or disables the logic function.

- AND—Output is high (1) only if both inputs are non-zero
- OR—Output is high (1) if either input is non-zero
- XOR—Output is high (1) if either, but not both, inputs are non-zero
- NOT—Input is inverted to the output
- NAND—Output is low (0) if both inputs are high (> 0)
- NOR—Output is high (1) if both inputs are zero (0)

Set-Reset (S/R) Flip-Flop

The Set/Reset Flip-Flop, or S/R Latch, is the most basic memory device. The S input sets the output to 1, and the R input sets the output to 0. Selecting **Insert input** turns the inputs to low active, so a 0 on the S input set the Q output to 1. The truth table below is defined without the invert flag.

Set/Reset (S/R) Flip-Flop				
S	R	Q	Qn *	
1	0	1	0	
0	0	No change	No change	
0	1	0	1	
1	1	Invalid ¹		

Delay (D) Flip-Flop

The Delay (D) Flip-Flop takes the D input to the outputs at the next rising edge of the clock. All other states of clock have no effect on the outputs. The **Enable** input is optional. A **Clock** input and a **D** input changing at the same time results in an undetermined state.

Invert Inputs—Typically a input value of "1" will be evaluated as true (active), selecting "Invert inputs" will result in a "0" value to be considered true(active).

FlipFlop is R dominated—xxx

Table references:

- X = doesn't matter what the value is.
- <up arrow>, <down arrow> = signify a rising or falling edge of the clock input.
- * = optional input or output

Delay (D) Flip-Flop				
*Enable	D	Clock	Q	Qn *
0	X	X	No change	No change
1	X	0 or 1 or ↓	No change	No change
1	0	↑	0	1
1	1	↑	1	0

Toggle (T) Flip-Flop

The Toggle (T) Flip-Flop toggles the state of the output if the input is 1. The **Enable** input is optional. A **Clock** input and a **T** input changing at the same time results in an undetermined state.

Toggle (T) Flip-Flop				
*Enable	T	Clock	Q	Qn *
0	X	X	No change	No change
1	X	0 or 1 or ↓	No change	No change
1	0	↑	0	1
1	1	↑	(toggle)	(toggle)

Create a Control Logic Rule

1. From the **Action Rules > Control Logic** screen, click **Add Control Rule**.
2. **Name** your new rule.
3. Select the logical operator from the **Gate** drop-down list.

¹ The Set input and Reset input active at the same time is an illegal condition and should not be used.

4. From the drop-down lists, select the Input 1, Input 2, Enable, and Clock options.
5. Select the **Output** options.

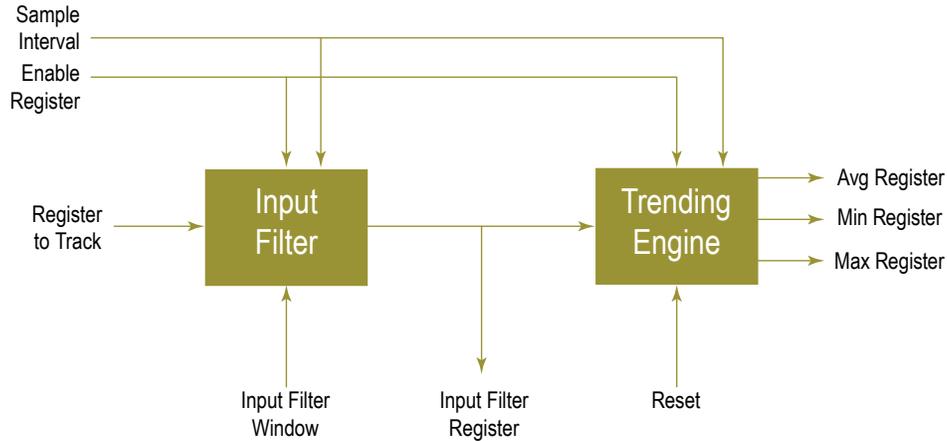
2.3.5 Trending and Filtering Rules

Use **Trending** rules to find the average, minimum, and maximum values for a specific register at a user-specified time interval. The Trend function collects data when the device boots up and accumulates the average, minimum, and maximum values for a register.

Local Registers locations to define include: Register to Track, Filter Register, Average Value Register, Minimum Value Register, Maximum Value Register, and Enable Register. When not using the Local Register definitions, leave them set to zero.



The filter and tracking is applied according to this flowchart.



The Trending rule has an optional filter input, and the filter output is the input to the Trending operation. The filter result is used before applying the Trending rule when you define the **Filter Register** parameter. The possible filters are:

Cumulative Average

Sums the samples and divides by the number of samples.

Exponential Moving Average

A moving average filter that applies weighting factors that decrease exponentially. The weighting for each older sample decreases exponentially, never reaching zero.

Lulu Smoother

Takes the minimum and maximum values of mini sequences of sample points. The mini sequence length is defined by the filter window parameter.

Median Average

Takes a least three sample points, sorts the sample points numerically, and selects the middle value as the result. A new sample point into the filter removes the oldest sample point.

Recursive Filter

Created from a percentage of the current sample plus a percentage of past samples. The number of samples is determined by the **Filter Window** parameter.

Simple Moving Average

Takes the number of samples defined by the **Filter Window** parameter and averages the samples for the result. When a new sample is taken, it becomes the latest sample point, and the oldest sample point is taken out of the filter averaging.

Weighted Moving Average

A moving average filter that applies a weighting factor to the data based on when the data was captured. More recent data has a higher weighted factor.

The other Trending parameters are listed. The Average, Minimum, and Maximum register fields are optional and can be left set to zero when using the Filter output register.

Register to Track

Local Register to provide the input to the filter and Trending rule.

Sample Interval

Defines the time interval of the samples to collect from the Local Register.

Filter

Defines the specific filter algorithm to apply to the front-end of the Trending rule.

Filter Window

Defines the number of samples for a filter algorithm.

For example, if the sample interval is 5 minutes and the Filter Window is 100, this will create a 100 point moving average over a 500 minute time period.

Filter Register

The Local Register that stores the output of the first stage filter. This is not required to connect the filter front-end to the Trending rule processing.

Average Value Register

The Local Register number to store the average value of the Trending rule. The average is calculated from the beginning of power-up or from the Reset Trending time. (Hourly, Daily)

Defining this Local Register is optional and can be left at zero.

Minimum Value Register

The Local Register number to store the Minimum value of the Trending rule. The Minimum is captured from the beginning of power-up or from the Reset Trending time. (Hourly, Daily)

Defining this Local Register is optional and can be left at zero.

Maximum Value Register

The Local Register number to store the Maximum value of the Trending rule. The Maximum is captured from the beginning of power-up or from the Reset Trending time. (Hourly, Daily)

Defining this Local Register is optional and can be left at zero.

Reset Trend Data

Use to reset the Trend data accumulation. (Hourly, Daily, or not at all)

Enable Register

The Enable Register is an optional register and can be left set to zero.

The Enable Register defines the address of a Local Register that can be used to turn on and off the trending function. Set the value of the Local Register defined to a 1 to turn on trending; set the value to zero to turn off trending. When disabled, the current data is held until the rule is enabled again.

Create a Basic Trend Rule

To create a basic Trend rule, follow these steps.

1. From the **Action Rules > Trending** screen, click **Add Trend Rule**.
2. In the **Name** field, name your Trend Rule.
3. Define the **Register to Track**.
4. Define the **Sample Interval** (hours:minutes:seconds) you'd like to collect the data.
5. Select **Filters** and your **Filter Window**.
6. Enter the Local Registers to copy the **Average, Minimum, and Maximum** values into.

2.3.6 Tracker Rules

Tracker rules monitor a Local Register and store the result of a function in another Local Register.

The possible functions are:

- Count the number of register transitions from 0 to non-zero value (rising edge). The speed at which it can count depends on how much work the DXM is performing. Typical tracker rates should be around 1 to 2 times per second.
- Track time in milliseconds the Local Register is in the non-zero state
- Track time in milliseconds the Local Register is in the zero state

Name	Register to Track	Transition to Track	Result Register
New tracker	1	None	Rising edge
			2

Create a Tracker Rule

1. From the **Action Rules > Trackers** screen, click **Add Tracker Rule**.
2. **Name** your new tracker rule.
3. Select the **Register to Track** from the list.
4. Select the **Transition to Track** from the drop-down list.
 - Rising edge—Transitioning from a zero to a non-zero value.
 - On time in ms—Length of time the register is on (non-zero), in milliseconds.
 - Off time in ms—Length of time the register is off (zero), in milliseconds.
5. Select the **Result Register** to store the count in.

2.4 Scheduler

Use the **Scheduler** screens to create a calendar schedule for local register changes, including defining the days of the week, start time, stop time, and register values.

Schedules are stored in the XML configuration file, which is loaded to the DXM. Reboot the DXM to activate a new schedule.

If power is cycled to the DXM in the middle of a schedule, the DXM looks at all events scheduled that day and processes the last event before the current time.

Right-click on most fields to access a context-sensitive menu. The most common options in the menus are **Cut**, **Copy**, **Paste**, **Delete**, and **Insert** to make data entry easier.

2.4.1 Create a Weekly Event

Use the **Scheduler > Weekly Events** screen to define weekly events.

1. Click **Add New Rule**.
A new schedule rule is created.

- Click on the arrow to the left of the new rule to expand the parameters into view. The user-defined parameters are displayed.
- Name your new rule.
- Enter the local register.
- Select the days of the week this rule applies to.
- Enter the starting value for the local register.
- Use the drop-down list to select the type of Start at time: a specific time or a relative time.
- Enter the starting time.
- Enter the end time and end value for the local register.

Register updates can be changed up to two times per day for each rule. Each rule can be set for any number of days in the week by clicking the buttons M, T, W, Th, F, S, or Su.

If two register changes are defined for a day, define the start time to be before the end time. Select **End Value** to enable the second event in a 24 hour period. To span across two days (crossing the midnight boundary), set the start value in the first day, without selecting **End Value**. Use the next day to create the final register state.

Start and end times can be specified relative to sunrise and sunset, or set to a specific time within a 24 hour period. When using sunrise or sunset times, set the GPS coordinates on the device so it can calculate sunrise and sunset.

2.4.2 Create a One-Time Event

Define one-time events to update registers at any time within a calendar year.

Similar to Weekly events, the times can be specific or relative to sunrise or sunset. Define one-time events using the **Scheduler > One Time Events** screen.

- Click on **Add One Time Event**. A new one-time event is created.
- Click on the arrow to expand the parameters into view. The user-defined parameters are displayed.
- Name your one-time event by clicking on the name link and entering a name.
- Enter the local register.
- Enter the starting time, date, and starting value for the local register.
- Enter the ending time, date, and ending value for the local register.

2.4.3 Create a Holiday Event

Use the **Scheduler > Holidays** screen to create date and/or time ranges that interrupt weekly events.

- Click on **Add Holiday**. A new rule is created.
- Enter a name your new holiday rule.
- Select the start date and time for the new holiday.
- Select the stop date and time for the new holiday.

2.4.4 Create a Dynamic Schedule

Use the **Dynamic Update** screen to create a scheduler update file. The scheduler update file is a configuration XML file that contains only scheduler data definitions. When you upload the scheduler update file to a DXM that is running a defined schedule, the scheduler update file replaces the running schedule.

To use the Dynamic scheduler feature, follow these steps.

1. Create the main XML configuration file with the entire configuration of the DXM, including default scheduler settings.
2. Go to **Scheduler > Dynamic Update** and select **Enable dynamic scheduler**.
3. Go to **Device > Save** to save the XML configuration file.
4. Load the main XML file into the DXM.
The DXM is running the main configuration file and will accept scheduler update files.
5. Modify the schedule settings, then click **Save scheduler update file**.
The scheduler update file is created.
6. To load the scheduler update file, click **Load scheduler update file**.
7. To send the scheduler update file to the DXM, click **Send scheduler update to device**.
The new schedule defined in the scheduler update file replaces the existing operating schedule.

Enable dynamic scheduler

Enables scheduler update files to be accepted.

This must be selected in main XML configuration file loaded into the DXM before it will accept scheduler update files.

Get scheduler update from device

Downloads the scheduler update file from the DXM.

Send scheduler update from device

Uploads a scheduler update file to the DXM, replacing the existing schedule running on the DXM.

Save Scheduler update file

Saves a scheduler update file to the PC. The file only contains the scheduler configuration data.

Load scheduler update file

Loads a scheduler update file from the PC to the DXM Configuration Tool software.

2.5 Register View

The Register View is a Modbus utility to help debug configurations and view register data for devices connected to the DXM.

The screenshot shows the 'Read/Write Source and Format' window. At the top, there are settings for 'Data Format' (Decimal), 'Register source' (Local registers), 'Slave ID' (199), and 'Timeout' (5 sec). Below these are checkboxes for 'Show local register names' and 'Apply sign and register scaling to values'. The main area is divided into two panels: 'Read Registers' and 'Write Registers'. Each panel has a 'Starting Register' and 'Number of registers' input field. The 'Read Registers' panel shows a table with 6 registers, each with an ID, name, and value. The 'Write Registers' panel shows a table with 6 registers, each with an ID, name, and value.

Starting Register	ID	Value
1	Analog IN 1	0
2	Discrete OUT 1	0
3	Analog IN 2	0
4	Discrete OUT 2	0
5	Humidity	0
6	Temp F	0

Starting Register	ID	Value
15	None	0
16	None	0
17	None	0
18	None	0
19	None	0
20	None	0

Follow these steps to select the registers to view.

1. Connect to the DXM using USB or Ethernet.
2. From the **Register source** drop-down list, select between Local Registers (DXM), ISM registers (Gateway), remote device, I/O registers, or display registers. Enter a Slave ID when accessing remote devices.
3. Select the **Starting Register** to display and the **Number of registers**.

Read Registers

To read the contents of a specific register or range of registers, select the starting register and the number of registers to read from.

Click **Read Registers** to read the data once. The register values display in the **Read Registers** section. Select **Enable Polling**, specify a polling rate, and click **Read Registers** to create a constant polling loop.

Write Registers

To write values to a specific register or range of registers, select the starting register and the number of registers to write to. Enter the value to write to these registers and click **Write Registers** to send these defined values to the selected registers.

2.6 Settings

Use the **Settings** screen to define general parameters, cloud services settings, logging parameters, Script Basic settings, networking and e-mail/SMS messages settings, administration parameters, notes, and I/O board parameters.

2.6.1 General

Use **Settings > General** to define the time zone, Modbus Master serial interface, Modbus Slave serial interface parameters, and to manually enter your GPS coordinates.

The screenshot displays the 'Settings' interface with the following sections:

- Device Time:** Shows 'Current device time' with a refresh icon, buttons for 'Get time from device' and 'Synch PC time with device', a dropdown for 'Device timezone (UTC)' set to 'UTC (UTC+00:00)', and a checkbox for 'Device observes DST'.
- RS-232 Settings:** Includes dropdowns for 'Baud rate' (19200), 'Parity' (None), and 'Stop bits' (1).
- Device Location/GPS:** Features input fields for 'Default latitude' and 'Default longitude' (both 0), a dropdown for 'Acquire using' (Default Lat/Lon), an 'External GPS address' field (0), a time picker for 'Acquire GPS data every (hh:mm:ss)' (00:00:00), and a checkbox for 'Send location to cloud'.
- Modbus Master Communications:** Includes dropdowns for 'Serial mode' (RS-485), 'Baud rate' (19200), and 'Parity' (None); time pickers for 'Communication timeout (hh:mm:ss.fff)' (00:00:05.000) and 'Maximum polling rate (hh:mm:ss.fff)' (00:00:00.050); and a dropdown for 'Automatic Radio Polling' (None).
- Modbus Slave Communications:** Includes dropdowns for 'Serial mode' (RS-485), 'Baud rate' (19200), and 'Parity' (None), and a checkbox for 'Wireless Modbus backbone'.
- Device Information:** Contains a 'Get device information' button and a table with two columns: 'Processor information' and 'IO board information'. The Processor information includes Serial number, Model number, Firmware number, and Version. The IO board information includes Serial number, EEPROM number, Model number, EEPROM version, Firmware number, and Firmware version.

Device Information

Click **Get Device Information** to have the DXM read and display its serial number, model number, version, and firmware numbers for both the processor board and the I/O board.

Device Location/GPS

Defines the latitude and longitude for the device. The GPS data can be entered manually for fixed assets or by defining an external GPS unit connected to the master Modbus RS485 bus. Select **Send Location to Cloud** to report this information to the Web site.

Device Time

Sets the time zone offset and displays the current time on the device or sync time with the PC.

Modbus Master Communications

Defines the Modbus Master serial port settings on the device (main RS-485 port). The default communications settings are 19200 baud, no parity, RS485. When changing the communications settings on the DXM, any devices attached to the bus also need to be changed, including the internal ISM radio. The ISM radio communication parameters should be changed first using the User Configuration Tool for DX80/Performance radios or the MultiHop Configuration Tool for MultiHop radios.

Communication timeout—Defines the length of time (hours:minutes:seconds) reserved for each Modbus master command to complete. The factory default of 5 seconds should be adequate for most Modbus slave devices and DX80 star wireless networks. This may need to be lengthened for MultiHop networks with battery powered devices or networks with many chained repeaters.

Maximum polling rate—Defines the time interval (hours:minutes:seconds) between Modbus messages. The default is set to 50 milliseconds. Some Modbus devices cannot communicate at this rate and may require greater than 50 milliseconds.

Modbus Slave Communications

Defines the Modbus Slave communications parameters and Modbus Slave ID when the RS-232 flow control signals are configured as a secondary Modbus serial port. Select **Wireless Modbus Backbone** to enable the Modbus slave port to come through a MultiHop HE5 module plugged into the SAM4 processor board (DXM150-S1R2P model only).

RS-232 Settings

Select the baud rate, parity, and stop bits for RS-232 communication.

Automatic Radio Polling

Automatic Radio Polling continually polls the embedded DX80 ISM radio for data changes.

Using automatic polling is optimal for data throughput from the radio, but monopolizes the bandwidth on the master RS-485 bus. If other resources are expected to share the master RS-485 bus, it may be better to access radio data using Register Mapping. Using **Automatic Radio Polling** allocates DXM Local Registers 1–768 for the DX80 ISM radio devices (the Gateway and Nodes 1–47)

The processor organizes the data in DXM **Local Registers** based on the user selections defined below.

None

Disables automatic radio polling.

The user must use Register Mapping or ScriptBasic to transfer register data with the ISM radio.

Group by Device - Inputs only

Only input data from the radio is collected and stored in the DXM Local Registers. The data is organized (grouped) by the device.

The first 16 registers (1–16) are the Gateway registers. Node 1 uses Local Registers 17–32, Node 2 uses Local Registers 33–48 and so on.

See the table for more details.

Group by Device - Inputs/Outputs

Input data from the radio is collected and stored in the DXM Local Registers. Local Registers defined as the outputs for devices are automatically written to the ISM radio when the contents change. The data is organized (grouped) by the device.

The first 16 registers (1–16) are the Gateway registers. Node 1 uses Local Registers 17–32, Node 2 uses Local Registers 33–48 and so on.

See the table for more details.

Group by In/Out - Inputs only

Only input data from the radio is collected and stored in the DXM Local Registers. The data is organized (grouped) by each input.

For example, all input 1 data is stored in the first 48 Local Registers (1–48), Gateway, Node 1–47). The next 48 registers (49–96) store input 2 data for all devices.

Local Registers 1–384 store inputs 1–8, Local Registers 385–768 store outputs 1–8.

See the table for more details.

Group by In/Out - Inputs/Outputs

Input data from the radio is collected and stored in the DXM Local Registers. Local Registers defined as the outputs for devices are automatically written to the ISM radio when the contents change. The data is organized (grouped) by each input.

For example, all input 1 data is stored in the first 48 Local Registers (1–48), Gateway, Node 1–47). The next 48 registers (49–96) store input 2 data for all devices.

Local Registers 1–384 store inputs 1–8, Local Registers 385–768 store outputs 1–8.

See the table for more details.

Table 1: DXM Local Registers organized by Device (GW=0, Nodes 1–47) where DXM Local Register address = (Device# × 16) + DevReg#

Device Type	Device Reg		Gateway	Node 1	Node 2	Node 3	Node 4		Node 47
Input 1	1	→	1	17	33	49	65		753
Input 2	2	→	2	18	34	50	66		754
Input 3	3	→	3	19	35	51	67	...	755
Input 4	4	→	4	20	36	52	68		756
Input 5	5	→	5	21	37	53	69		757
Input 6	6	→	6	22	38	54	70		758
Input 7	7	→	7	23	39	55	71		759
Input 8	8	→	8	24	40	56	72		760
Output 1	9	→	9	25	41	57	73		761
Output 2	10	→	10	26	42	58	74		762
Output 3	11	→	11	27	43	59	75	...	763
Output 4	12	→	12	28	44	60	76		764
Output 5	13	→	13	29	45	61	77		765
Output 6	14	→	14	30	46	62	78		766
Output 7	15	→	15	31	47	63	79		767
Output 8	16	→	16	32	48	64	80		768

Table 2: DXM Local Registers organized by Input/Output (GW=0, Nodes1–47) where DXM Local Register address = ((DevReg# - 1) × 48) + (Device# + 1)

Device Type	Device Reg		Gateway	Node 1	Node 2	Node 3	Node 4		Node 47
Input 1	1	→	1	2	3	4	5		48
Input 2	2	→	49	50	51	52	53		96
Input 3	3	→	97	98	99	100	101	...	144
Input 4	4	→	145	146	147	148	149		192
Input 5	5	→	193	194	195	196	197		240
Input 6	6	→	241	242	243	244	245		288
Input 7	7	→	289	290	291	292	293		336
Input 8	8	→	337	338	339	340	341		384
Output 1	9	→	385	386	387	388	389		432
Output 2	10	→	433	434	435	436	437		480
Output 3	11	→	481	482	483	484	485	...	528
Output 4	12	→	529	530	531	532	533		576

Device Type	Device Reg		Gateway	Node 1	Node 2	Node 3	Node 4		Node 47
Output 5	13	→	577	578	579	580	581		624
Output 6	14	→	625	626	627	628	629		672
Output 7	15	→	673	674	675	676	677		720
Output 8	16	→	721	722	723	724	725		768

2.6.2 Cloud Services

Use **Cloud Services** to define the parameters to send register data to the website.

Cell Configuration

Select the DXM's cellular modem from the **Cell module** drop-down list.

- CE910 2G CDMA—A Telit CE910 CDMA cellular modem card that operates on the Verizon wireless network.
- HE910 3G GSM/HSPA—A Telit HE910 3G GSM cellular modem card that operates on various carriers worldwide. The modem supports 2G GSM if 3G is not available.
- LE910 4G LTE—A Telit LE910 4G LTE cellular modem card that operates on the Verizon LTE network. The modem does not have a fall back to 3G network.

APN, **APN Username**, and **APN Password** are user-defined fields. The APN setting for the LTE cellular modem using a Banner Connected Data Solutions wireless plan is "vzwinternet".

Cloud Push

The **Cloud Push Interval** defines the time interval for cyclical data pushes sent to a webserver or host system. Setting the **Cloud Push Interval** to zero disables cyclical push messages. When using pushes created by a threshold rule, the **Cloud Push Interval** is ignored. Cloud pushes created from ScriptBasic are independent from these settings.

Sample count defines how many times between the **Cloud Push Interval** to save data. The sample points are saved, then sent at the next **Cloud Push Interval**.

For example, with a **Cloud Push Interval** set to 1 hour and the sample count set to 10, the DXM saves the register data every 6 minutes and pushes all data to the web server every hour.

The **Push Port** defines the HTTP push data port on the web server. The factory default setting is port 80. Set the port to 443 when you are using HTTPS.

HTTPS

Selecting **HTTPS** indicates to the DXM to use TLS (Transport Layer Security) as a sub-layer under regular HTTP application layering. HTTPS encrypts and decrypts user data to and from the web server. The webserver is required to carry a certificate. Select **Use HTTPS** to enable TLS services. Not all servers/communication networks support TLS.

HTTPS (SSL) only works with ethernet and LE910 (4G) cell modems. Any other combination disables SSL. Enabling SSL push automatically sets your push port to 443 (this is still user configurable).

A **Certificate CN** is required to be associated with one or more domain names, called common name (CN). A single name Certificate CN is typically www.yoursite.com. A wildcard certificate includes single level sub-domains. The certificate CN is a wildcard certificate; *.bannercds.com. If nothing is entered into this field, the BannerCDS CN will be used; *.bannercds.com.

The DXM uses the **Network Interface** settings to determine where to send cyclic report data, alert messages, or log files. Selecting Ethernet or Cell also requires setting up the parameters under the **Settings > Network** tab.

When Cellular is selected, the optional VPN (Virtual Private Network) is available. This requires the private cellular plan and allows for direct point-to-point communication between the DXM and host system. Enabling VPN forces the DXM to constantly listen for messages from a host system.

Network Interface

Ethernet Retries per push interval—Used when pushing HTTP data to a webserver using Ethernet and defines the number of attempts made to send HTTP data to the web server before it will save the message into a local file to be pushed up at the next push interval. (default 5)

Push interface—Select Ethernet or Cell from the drop-down list.

Web Server

Under the **Web Server** section, a **Host Header** allows for multiple domains to reside at the same server address. Similar to how many people can share one phone number, dialing one number connects you to the phone but to talk to someone specific you need another piece of information. This is where the **Host Header** field is used. An IP address can get you to a server and the **Host Header** field allows for multiple domains residing at that server. Not all servers require this field, when left blank the intended target of the push message will be the server name. The website does not require this field.

The **Server Name/IP** is the webserver address used by the DXM when pushing data to the cloud. Enter the domain name, `push.bannercds.net`. The DXM defaults to using a public DNS (domain name server) to resolve domain names. To use a specific DNS enter the IP address under the **Settings > Network** tab.

Enter the **Page** that directs incoming data at the web server. (/push.aspx)

From the **Site ID** drop-down list, select either GUID or User Defined Text. For custom web servers the Site ID string can be changed from a GUID (Global Unique Identifier number) to a user defined string of numbers or letters. The user-defined text is a unique site string defined by the web server when a site is created. Copy the string from the website into this field in the DXM configuration tool.

Users can create up to five **Custom HTTP Headers** to provide custom information to the web server. The custom headers are sent with every push to the web server. **Custom HTTP Headers** are text fields the user can define that will be included in the HTTP header when the data is pushed to the web server.

When **Include XML GUID on first push** is selected, the first initial push to the web server includes the configuration file GUID (Global Unique Identifier) number. The web server can then verify that the XML configuration file that is loaded into the DXM is the same as what is loaded into the web server.

Web Server Authentication

Webserver Authentication defines a username and password to be sent to the webserver with every push data set to validate the sending device before storing any data in the database. If the webserver is expecting login credentials, the DXM must be programmed with the username and password. This only provides login credentials for authentication to the server; this does not secure the data payload.

The DXM must be connected to the computer. Select **Require Authentication**, then enter a username and password. Select **Send Authentication** to send the username and password to the DXM to be stored in non-volatile memory. Manually cycle power to the DXM after the username and password are written to it. The credentials cannot be read from the DXM.



Important: After **Require Authentication** is selected and the username and password are sent to the DXM to be stored, go to **File > Save** to save the XML configuration file. Then send the new XML configuration file to the DXM. If the configuration file loaded into the DXM does not have **Require Authentication** selected, the username and password will not be sent.

Configure the Cloud Settings

1. To configure the connection to the web server, go to the **Settings > Cloud Services** screen.

The screenshot shows the 'Cloud Services' configuration screen. It is divided into several sections:

- Network Interface:** Push interface (Ethernet), Ethernet retries per push interval (5).
- Cell Configuration:** Cell module (CE910 2G CDV), Enable Server Mode (checkbox), APN, APN Username, APN Password.
- Cloud Push:** Cloud push interval (None), Sample count (1), Push port (80), Print push debug messages to serial console (checkbox).
- Web Server Authentication:** Require Authentication (checkbox), Username, Password, Send Authentication button.
- Web Server:** Server name/IP (push.bannercds.com), Page (/push.aspx), Host header, Site ID is (GUID), Clear button (00000000-0000-0000-0000-000000000000), Custom HTTP Headers, Initial Push Options (Include XML GUID on first push checkbox).
- HTTPS:** Use HTTPS (checkbox), Certificate CN (*.sensonix.net).

2. Copy and paste the **Site ID**.

The Site ID is that long string of numbers and letters from the Banner Connected Data Solutions website.

3. Verify the **Server Name/IP** is set to `push.bannercds.com` and the **Page** is set to `/push.aspx` for sending to the website. To push to the demo website, set the Server Name/IP to `demopush.sensonix.net`.
4. Set the **Cloud Push Interval** to a value appropriate for your application.

The **Cloud Push Interval** determines how often the device pushes the data to the web. The faster the push interval, the more data is sent to the site. Cellular plans can only push at an interval of 5 minutes or longer, while Ethernet connections can push at an interval of 1 minute or longer. The **Sample Count** specifies how many times the data is gathered within the **Cloud Push Interval**.

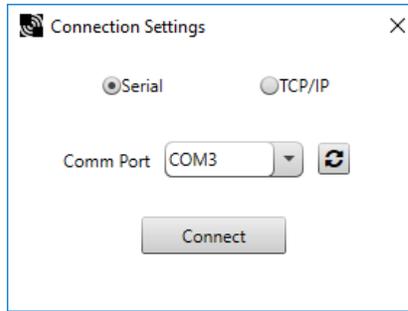
For example, if the **Cloud Push Interval** is 15 minutes and the **Sample Count** is set to 3, then during each data push (every 15 minutes), 3 samples are sent to the web. This is one sample every 5 minutes.

5. Save the configuration file by going to **File > Save**.

File names must be no more than 30 characters long, and should not contain any spaces or special characters.

6. With a USB cable connected to the device, go to the **Device > Connection Settings** menu.

7. Select the appropriate **Comm Port** and click **Connect**.



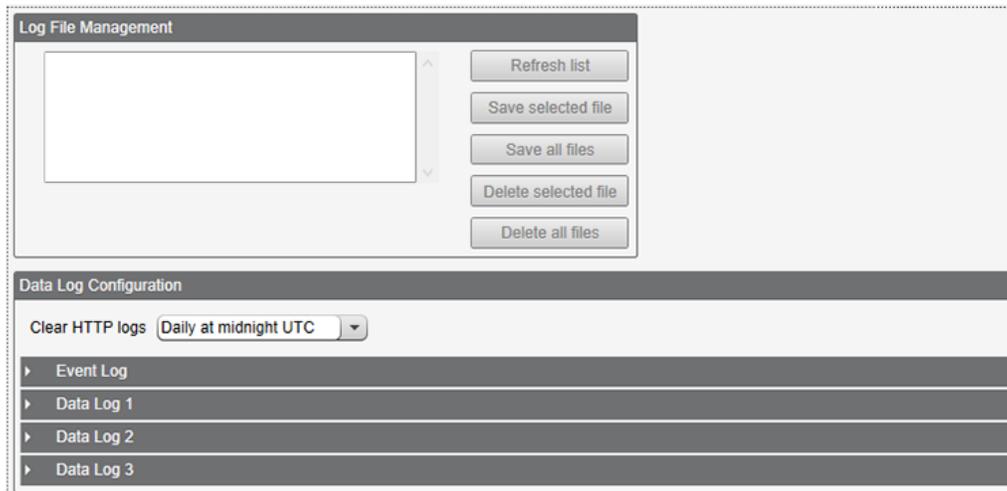
If multiple comm ports are visible, try each one until the software is able to connect to the device.

8. Go to **Device > Send Configuration to the Device** to upload the new XML file.

2.6.3 Logging

Use **Settings > Logging** to define the local logging setup for the on-board SD card of the DXM. Up to three cyclical logs and one event log can be defined for threshold events. Click on the arrow to the left to expand the log configuration parameters.

The string length limit for each line in the Log file is 1,000 characters. Lines typically start with the time stamp followed by the values of each saved register. Discrete values (0/1) are short, but analog values could be up to 10 characters each. Taking an average of 6 characters for each value stored, the practical limit will be about 160 registers per log file.



The data log configuration parameters define the log file name, size, and what to do when the size is exceeded. Files can be sent via email when full.

View, save, or delete files stored on the micro SD card using the **Log file Management** section. The DXM needs to be connected to a PC for these features to operate.

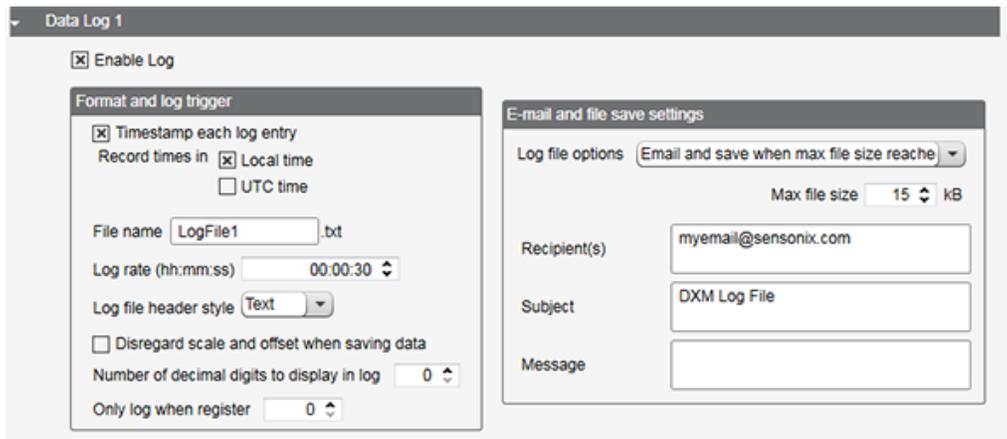
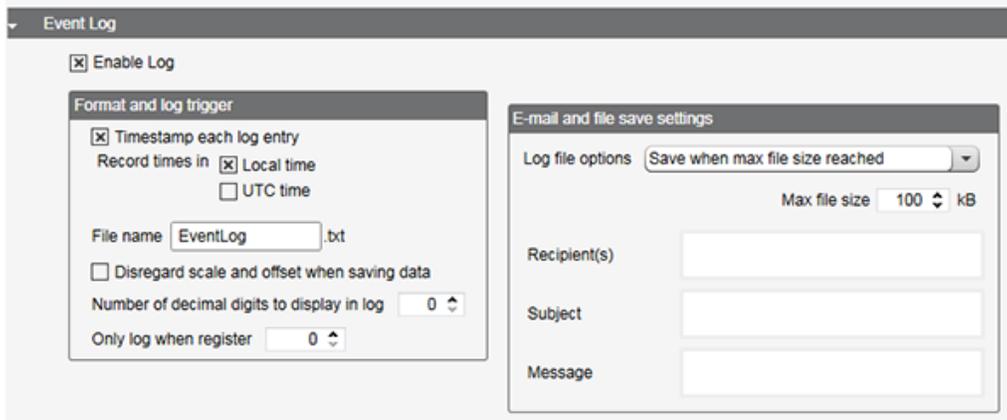
An HTTP Log file is created for a data packet pushed to the web server. Successful pushes cause the HTTP log to be time-stamped and placed into a dated folder under the "_sxi" directory. Typically, these logs are deleted daily.

Transmissions that fail to make it to the web server accumulate in the HTTP Log file, retrying the failed attempts at the next cyclic interval. After 10 failed attempts, the HTTP Log file is time-stamped and saved on the SD card in a separate directory ("_sav") and must be retrieved manually.



Create an Event Log or Data Log

Use the **Event Log** to track data only when the contents of the specified registers reaches the defined value. The **Data Logs** record the contents of the selected register at the defined rate.



1. Select **Enable Log** to enable the log file creation. The register data to be saved in the log is defined under the **Local Registers** tab.
2. Select **Timestamp each log entry** to date/time stamp each log entry. Select whether you want the time stamps to be in local time or UTC time.
The local time is determined from the device's time zone selection under on the **Settings > General** screen.
3. Name your Event or Data Log.
A date/time stamp is added to the end of the user-defined **File Name** then stored on the Micro SD card. New files will be created with a new date/time stamp.

4. Select whether or not you want to **Disregard scale and offset when saving data** for the register or apply the scale and offset defined on the **Local Register Configuration** screen.
Local Registers 1 through 845 are integer registers and are always stored as integers. Floating point Local Registers 1001 through 2000 are stored as floating point numbers.
5. Define the **Number of decimal points to display in log** (floating point data log only).
6. Define the **Email Parameters** for sending a log file when the log file is full.
7. For cyclical log files (log1-log3), specify the **Log Rate**.
8. Set the **Maximum File Size** before the DXM creates a new file. When emailing log files, keep the file sizes below 50 KB.
9. Select from the following **Log file options**.
 - Save when max file size reached—Save the log file to the internal DXM SD card when the defined Max file size is reached. The user must retrieve the file.
 - Save when register is non-zero—Save the log file to the internal DXM SD card when the defined Local Register is set to a non-zero value. The Local Register can be controlled by Action Rules, ScriptBasic, Scheduler, or Website. The user must retrieve the file.
 - Save Daily—The log file will be saved daily at 00:00 UTC time. The user must retrieve the file.
 - Email and Save when max file size reached—Email and save the log file to the internal DXM SD card when the defined Max file size is reached. You may define up to three recipients to receive the email. Define the SMTP parameters on the **Settings > Network** screen.
 - Email and Save when register is non-zero—Email and save the log file to the internal DXM SD card when the defined Local Register is set to a non-zero value. The Local Register can be controlled by Action Rules, ScriptBasic, Scheduler, or Website. You may define up to three recipients to receive the email. Define the SMTP parameters on the **Settings > Network** screen.
 - Email and Save Daily—The log file will be emailed and saved daily at 00:00 UTC time. You may define up to three recipients to receive the email. Define the SMTP parameters on the **Settings > Network** screen.
10. Enter the **Recipient** email address, **Subject**, and **Message** for the log file.
You may define up to three email recipients, separated by commas.

2.6.4 Scripting

The DXM can run one ScriptBasic program. Use the **Scripting** screen to save, load, or delete script files on the DXM.

Files other than ScriptBasic files can be uploaded to the DXM. Files are placed in the root directory and can only be accessed by a ScriptBasic program.

Enabling memory Usage Tracking enables an internal memory watcher program to track the memory allocation for a ScriptBasic program. The results are displayed on the DXM's LCD, under **System Info > Script**.

ScriptBasic Options

TCP server enabled—ScriptBasic can operate as a server watching for Ethernet packets. Use the FILEIN command in ScriptBasic Default port is 8845.

TCP Client enabled—ScriptBasic can send Ethernet packets when this is enabled. Enter IP address of destination. Default port is 8845, port is selectable.

UDP enabled—Enable ScriptBasic to send/receive UDP data traffic. Enter IP address of destination.

The screenshot displays two main sections of the configuration tool:

- Script File Management:** This section includes an "Upload file" button on the left, a large empty rectangular area in the center, and a vertical stack of four buttons on the right: "Refresh list", "Save selected", "Delete selected", and "Add selected to startup". At the bottom of this section, there is a label "Script to run at startup" and a "Clear startup script" button.
- Script Basic options:** This section contains three rows of settings:
 - TCP server enabled Port: 1
 - TCP client enabled Port: 1 IP Address: 0.0.0.0
 - UDP enabled Port: 1 IP Address: 0.0.0.0

Select a ScriptBasic file from the current ScriptBasic files window and select **Add Selected to Startup Scripts** to define which program should be run at boot time. Save the XML configuration file before loading to the device. A reboot is required to start a new script program.

To upload files:

1. Click **Upload**. A pop-up box appears.
2. Select the pull-down menu to select the file types: .sb, .esb or *.*.

2.6.5 Network

Use the **Network** screen to define the Ethernet connectivity settings, Cellular firewall settings (for VPN), and DNS settings.

Any network parameters adjusted using the **LCD Display** screen override the settings in the XML configuration file. The user must reset the network LCD setting on the LCD display before the network parameters in the XML configuration will be used.

Current Device IP IP Address 0.0.0.0 Subnet 0.0.0.0 Gateway address 0.0.0.0 Device MAC 00:00:00:00:00:00 <input type="button" value="Get Settings from Device"/>		Cell Firewall Cell firewall IP address 0.0.0.0 Cell firewall netmask 255.255.255.0	
IP Address DHCP <input type="button" value="v"/> IP address 0.0.0.0 Subnet 255.255.255.0 Gateway address 0.0.0.0		Network Parameters <input type="checkbox"/> Enable UDP console Ethernet Connection Acquisition Connection retry 5 Connection retry wait 00:00:10.000 Max delay between retries 00:01:00.000 Cellular Connection Acquisition Connection retry 5 Connection retry wait 00:00:10.000 Activity timeout 00:00:10 Watchdog timeout 00:00:10.000	
DNS Primary DNS IP address 0.0.0.0 Secondary DNS IP address 0.0.0.0 Primary cell DNS IP address 0.0.0.0 Secondary cell DNS IP address 0.0.0.0		Cell Software Firewall <input checked="" type="checkbox"/> Open software firewall *If there are no valid entries for a section, the firewall is left open for that category of messages. Phone Number <input type="text"/> <input type="button" value="Add"/> <input type="button" value="Remove"/> Open to all SMSs IP Address <input type="text"/> <input type="button" value="Add"/> <input type="button" value="Remove"/> Open to all IPs Email Address <input type="text"/> <input type="button" value="Add"/> <input type="button" value="Remove"/> Open to all Emails	

Cellular Firewall and Cellular Software Firewall

The cellular modem has a built-in firewall to restrict which host(s) communicates with the DXM using a private network. Define an IP address for a single host connection or use the netmask to create multiple IP connections (0.0.0.0 is open to any IP address).

Select **Open software firewall** to accept incoming messages on a cellular connection. If **Open software firewall** is selected and there are no entries present, the software firewall allows all traffic. Entries within the firewall settings will only allow the devices specified to communicate with the DXM.

Current Device IP Settings

The current IP address (when connected via Ethernet) can be read from the device or by using the LCD menu on the DXM Controller. Click **Get Settings From Device** when the device is connected via the USB port.

DNS

The DXM defaults to using a public DNS service to resolve domain names. Enter the IP address of a primary and secondary DNS server to select a particular DNS server.

The DXM defaults to using the carrier DNS service. To redirect the DXS requests to a different DNS service, enter a primary and secondary cell DNS.

IP Address

Select a static IP address or select the automatic assignment of an IP address by selecting DHCP from the drop-down menu. If Static IP is selected, enter the IP and subnet addresses.

You can also enter static IP addresses using the DXM's LCD menu system. Entering IP Addresses using the menu system overrides the IP addresses in the XML configuration files. Clear the IP addresses in the menu system to use the IP address in the XML configuration file.

Network Parameters

The Network Parameters define certain aspects of the user selected network, cellular or Ethernet, within the DXM. In most cases, the factory default parameters are sufficient for typical applications and do not need to be changed.

Ethernet Connection Acquisition—These parameters govern the operation of the DXM when it tries to connect to an Ethernet network.

- **Connection Retry**—Defines the number of attempts to join an Ethernet network (default 5). Only use this parameter with DXM firmware later than version 1.0. Typical settings are 0 through 5 seconds.
- **Connection Retry Wait**—Defines the amount of time in seconds to wait between the first and second retries. For every failed attempt to connect to the network the wait time increases by a multiple of the wait time. For example, if the wait time is 5 seconds, the delay between the first and second retry will be 5 seconds. The next delay is 10 seconds, then 15 seconds, then 20 seconds, up to the maximum time specified. Only use with DXM firmware later than version 1.0. Typical settings are 0 through 5 seconds.
- **Max Delay between Retries**—Defines the maximum amount of time to wait between retries. After every failed attempt, the wait time is increased by the original retry wait time until it reaches the **Max Delay between Retries**.

Cellular Connection Acquisition—Defines the interaction with the wireless network when trying to establish a connection. The default parameters should not need to be changed.

- **Connection Retry**—Defines how many attempts are to be made to create a connection with the wireless network.
- **Connection Retry Wait**—When a cellular connection fails, the **Connection Retry Wait** parameter defines how long the system waits until it attempts to create another cellular connection.

The following Cellular Server Mode parameters only apply to DXMs using a cellular network connection in Server Mode. Server Mode turns on the cellular modem continuously to allow for connections from a host system to the DXM. Do not configure the DXM to send push data automatically in Server Mode.

- **Activity Timeout**—An internal setting on the cellular modem that watches the incoming activity. If there isn't activity coming into the cellular modem for the specified amount of time, the connection closes and the device returns to listening mode. This catches the issue with the host system dropping a connection but the cell modem not aware of the dropped connection. Always set the Activity Timeout parameter to a time longer than the expected polling rate. If the polling rate is 1 hour this setting should be greater than 1 hour (e.g. 4000 seconds). The factory default setting is disabled (0).
- **Watchdog Timeout**—Used in an application routine running in the DXM processor. If there is no incoming cellular traffic within the watchdog time period, the cell modem resets. The Watchdog Timeout should be greater than the Activity Timeout; the watchdog routine is a failsafe mechanism to detect if the cellular modem is left in a bad state from the wireless network. The factory default setting is disabled (0).

2.6.6 Mail and Messaging

The DXM can email alarm conditions using Threshold rules and can email internal log files when the log files are full. The DXM must have an Ethernet or cellular network connection for e-mail alarm conditions.

The screenshot displays the configuration interface for the DXM. It is divided into three main sections:

- Mail Server Settings:** This section includes fields for 'SMTP server' (smtp.gmail.com), 'SMTP server port' (587), 'User name' (MyGmail@gmail.com), and 'Password' (MyPassword). There is a checkbox for 'Enable SMTP authentication' which is checked. Two radio buttons are present for encryption: 'No encryption' and 'Situational encryption', with 'Situational encryption' selected. A 'Send SMTP Password' button is also visible.
- E-Mail Recipients:** This section contains a list of ten recipients, each with a 'Address' field. The first recipient, 'E-mail Recipient 1', has the address 'receiver.work@gmail.com' entered. The other nine recipients have empty address fields.
- SMS Recipients:** This section is currently collapsed, indicated by a downward-pointing arrow.

E-Mail Recipients

Enter the list of email addresses to send message and alerts to. Include the subject line and message.

This is the global list of email addresses available to be used by the **Action Rules > Thresholds**. Select individual email addresses within the Threshold Rule. Email addresses for Logging are specified on the **Settings > Logging** screen.

Mail Server Settings

The server settings specify a SMTP (Simple Mail Transfer Protocol) server as well as provide the login credentials to the SMTP mail server. The SMTP server port defines where to send the e-mail message for submission. Port 25 or 587 is for typical sending e-mail. Some servers are set up to reject all messaging on port 25, but valid users authenticating on port 587 are allowed to relay mail to any valid address.

For mailing larger groups, create groups on the mail server. Define whether or not to use encrypted (secure) communications with the SMTP server. If your SMTP server does not require an encryption connection, select **No encryption**. If your SMTP server can select or de-select encryption as the connection between the SMTP server and client is negotiated, select **Situational encryption**.

If you enable authentication, you must enter the user name and password and click **Send SMTP Password**. This password is not stored in the XML configuration file and must be sent directly to the DXM by filling in the username and password and clicking **Send SMTP Password**.

SMS Recipients

Enter the list of people to send text messages to. Note that each cell phone server provider may have a slightly different address to send text messages to. Enter the message.

2.6.7 Administration

The DXM manages two passwords that are defined on the **Settings > Administration** screen. Defining a **File Protection** password requires a password to be entered before a configuration file or ScriptBasic program file is loaded to the DXM. The LCD protection password requires the user to enter a password on the DXM LCD to unlock the LCD display of the DXM.

File Protection—Setting a password requires that a user enters a password before loading a configuration file or ScriptBasic file. The password is stored in the DXM and requires the DXM to be connected to the computer via USB port or Ethernet to set the password.

LCD Protection—The LCD passcode locks the DXM LCD display menu from operation. A valid passcode must contain 1–9 numbers with the first number being non-zero. The user has access to the full menu system when the passcode has been entered using the DXM LCD menu. The user can re-lock the display by selecting the **Display Lock** menu or the DXM will automatically re-lock the display after 15 minutes with no activity. If the DXM reboots, the device prompts the user for the passcode.

The screenshot displays two side-by-side panels. The left panel, titled 'File Protection', contains a 'Get Device Status' button with the text 'Unknown' next to it. Below this is a text input field for 'Current admin password'. An 'Action' dropdown menu is set to 'Set/Change admin password'. There are two more text input fields for 'New admin password' and 'Verify new password', followed by a 'Submit' button. The right panel, titled 'LCD Protection', features a text input field for 'LCD passcode' containing the value '5039'. To its right are two buttons: 'Send LCD passcode' and 'Blank LCD passcode'.

1. With the device connected, click on **Get Device Status**.
2. The device reports back if it is locked or unlocked.
3. To change, set, or clear a password select the appropriate action, fill in the required fields then click **Submit**.

Reset the Controller if the Password is Lost

If the device is locked and the password is lost, go to the DXM to clear the password. Clearing the password from the controller also erases the configuration file and all ScriptBasic programs. To clear the password:

1. Remove the power from the device.
2. Set DIP switch 4 to the ON position.
3. Disconnect the Ethernet cable.
4. Apply power to the device.
5. Immediately after powering up, press and hold the reset button on the processor board.
After 10 to 15 seconds, all LEDs on the processor board flash quickly for a second. All files are erased and the password is cleared.
6. Remove the power from the device.
7. Set DIP switch 4 to the OFF position.
8. Reconnect the Ethernet cable.
9. Apply power to the device.
This erases all files from the device.
10. Reload your configuration files.

2.6.8 Notes

Use the **Notes** screen to enter informational text about the configuration file or notes specific to the application.

The **Notes** screen can save up to 4096 characters of alphanumeric characters and some special characters (~ ! @ # \$ ^ () _ - = + ? / }] { [). The notes are stored in the beginning of the XML file.

2.6.9 I/O Board

The DXM I/O boards come in various combinations of inputs and outputs. The selection of the base board type can be set to **Manual** or **Auto detect**. When modifying parameters, always read (GET) the parameters first, edit, and then write (SEND) the new parameters back to the I/O board.

Depending upon the amount of parameter data to see, click on one of the following buttons to read the current parameter settings:

GET all parameters and IO

Reads all the I/O board device level parameter data plus the input and output parameter settings.

GET parameters

Reads only the I/O board device parameters. Device level parameters effect the I/O board operation in general and are not specific to I/O points.

GET all IO points

Reads the parameters for all the inputs and outputs on the I/O board.

GET

Reads parameters for each individual I/O point, from one input or output.

Based on the number of changes made to the parameter data select one of the following buttons to write the data to the DXM I/O board:

SEND all parameters and IO

Writes all the I/O board device level parameter data plus the input and output parameter settings.

SEND parameters

Writes only the I/O board device parameters. Device level parameters effect the I/O board operation in general and are not specific to I/O points.

SEND all IO points

Writes the parameters for all the inputs and outputs on the I/O board.

SEND

Writes parameters for each individual I/O point, from one input or output.

Name

Assign a name to each device to it easier to track the device in the configuration software.

The **Name** must be less than 18 characters and contain only the standard alpha-numeric characters, the following standard ASCII characters: *, +, -, /, <, >, or a space.

After making any changes, click **SEND parameters** to write the changes. The status bar at the bottom of the screen indicates when the process is complete.

Default Conditions

Default conditions are the conditions under which outputs are sent to their defined default state. These conditions apply to all outputs.

Selecting **Start Up** sets this radio's outputs to their default values when this radio is powered up.

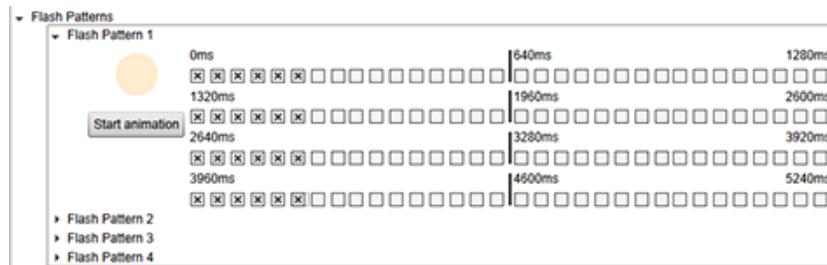
A **Comms Timeout** refers to the communication between the host system and the selected radio (which may be any radio within the network). Selecting Communication Timeout sets this radio's outputs to their selected default values when the host system has not communicated with this radio within the time specified. Set the communication timeout in seconds.

After making any changes, click **SEND parameters** to write the changes. The status bar at the bottom of the screen indicates when the process is complete.

After the radio problems are resolved with the parent radio or communication with the host system has resumed, the host system is responsible for writing the operating output value.

Flash Patterns

Setting the flash pattern establishes an on and off pattern that can be used for a discrete output or switch power.



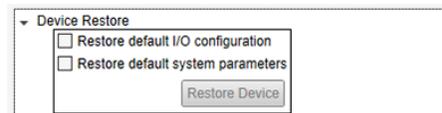
Define **Flash Patterns** by selecting specific timeslots to turn the output on or off. While originally the flash pattern was designed to turn on and off an indicator light, the flash pattern can be set for any discrete output or switch power. Each slot represents one frame size, which may vary from radio to radio. The default frame is 40 milliseconds. Users may configure up to four different flash patterns per radio device.

In the example shown, Flash Pattern 1 is configured to turn on the EZ-LIGHT for 240 milliseconds, then turn off the light for 1040 milliseconds. To preview the flash pattern, click on the **Start Animation** button. The EZ-LIGHT graphic flashes on and off in the programmed flash pattern.

After making any changes, click **SEND parameters** to write the changes. The status bar at the bottom of the screen indicates when the process is complete.

Device Restore

Use these commands to restore default settings to the selected device.



To restore default device parameters:

1. At the top of the screen, select a MultiHop device.
2. Click on **GET All Parameters** to get that device's I/O configuration.
3. In the **Device Restore** section, select the appropriate checkboxes.
 - **Restore Default I/O Configuration.** Restores default I/O configuration parameters to the values indicated in the device's datasheet, such as sample and report rates.
 - **Restore Default System Parameters.** Restore default system parameters, such as the radio configuration and binding code.
4. Click on **Restore Device** to restore the selected default values to your device.

Remap Registers or Register Aliasing

Use the **Remap Registers** section of the **Configure Device** screen to map registers to contiguous register locations to optimize Modbus read/write functions.

Device Restore

Remap Registers

Alias Registers (601-616)		Register Content (101-116)	
GET	SEND	GET	SEND
1	5	101	0
2	7	102	0
3	9	103	0
4	10	104	0
5	8	105	0
6	501	106	0
7	502	107	0
8	1	108	0
9	1	109	0
10	1	110	0
11	1	111	0
12	1	112	0
13	1	113	0
14	1	114	0
15	1	115	0
16	1	116	0

1. Verify the desired MultiHop Radio **Address** is selected in **Configuration Address**.
2. Click the arrow next to **Remap Registers**.
3. Fill in the source registers you would like to alias. The **Alias Registers** rows 1 through 16 are the user-defined entries of addresses of the registers to alias (rearrange). This alias table is stored in the MultiHop radio register addresses 601 through 616. In the example, source registers addresses 5, 7, 9, 10, 8, 501, and 502 are entered into the table.

The aliased **Register Contents** are in registers 101 through 116. For this example, when a host system reads Modbus registers 101 through 107 of the MultiHop radio, the register contents come from register 5, 7, 9, 10, 8, 501 and 502.

Inputs

The following parameters are used to configure the inputs.

The input parameters vary, depending on the enabled and available input types.

Input Configuration

Select **Enable** to turn on this input.

Sample Rate—The sample interval, or rate, defines how often the Sure Cross device samples the input. For battery-powered applications, setting a slower rate extends the battery life. Set the sample rate/interval in hours:minutes:seconds.

Universal Configuration

After enabling the input, configure the universal input type.

Count on falling values

Select to count on a falling transition.

Count on rising values

Select to count on a rising transition.

Input Type

Select the input type from the drop-down list.

- NPN
- PNP
- 0-20mA
- 0-10V
- Thermistor
- Potentiometer
- Bridge
- NPN Raw Fast

Preset Value (Counter)

Enter a number in the selection box and press the Set Value button to write a preset counter value to the register.

Analog Configuration

Enable push on change of state

Enables push registers for this input. When the discrete input changes state, the register value is pushed to the master radio if this register is configured to be a push register. For analog inputs, use the threshold and hysteresis parameters to define "on" and "off" points.

Enable register full scale

Turning Fullscale ON sets the entire register range of 0 through 65535 to represent the selected minimum through maximum input values. With Fullscale turned on, a register value of 0 represents the selected minimum value in microamps (for current inputs). A register value of 65535 represents the selected maximum value in microamps. For example, a register value of 0 is 0 and the register value of 65535 represents 20 mA (or 20,000 microamps). With Fullscale turned OFF, the register value represents unit-specific input readings. For units of current (mA), register values are stored as microAmps. Voltage values are stored as millivolts. A sensor reading of 15.53 mA is stored as 15530.

Signal Conditioning

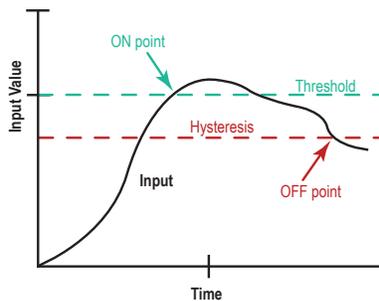
Delta

The delta parameter defines the change required between sample points of an analog input before the analog input reports a new value. To turn off this option, set the Delta value to 0. To use the delta function, the push registers must be defined.

Hysteresis and Threshold (Analog)

Threshold and hysteresis work together to establish the ON and OFF points of an analog input. The threshold defines a trigger point or reporting threshold (ON point) for a sensor input. Setting a threshold establishes an ON point. Hysteresis defines how far below the threshold the analog input is required to be before the input is considered OFF. A typical hysteresis value is 10% to 20% of the unit's range.

In the example shown, the input is considered on at 15 mA. To consider the input off at 13 mA, set the hysteresis to 2 mA. The input will be considered off when the value is 2 mA less than the threshold.



Median Filter

When the median filter is turned on, three samples are taken for each analog sensor reading. The high and low values are discarded and the middle value is used as the analog value. Set to zero (0) to turn off the median filter. Set to one (1) to turn on the median filter.

Sample High and Sample Low

For analog inputs, the sample high parameter defines the number of consecutive samples the input signal must be above the threshold before a signal is considered active. Sample low defines the number of consecutive samples the input signal must be below the threshold minus hysteresis before a signal is considered deactivated. The sample high and sample low parameters are used to avoid unwanted input transitions.

Tau Filter

Set to zero (0) to turn off the tau filter. Set to 1 (weakest filter) through 6 (strongest filter) to turn on the tau filter. (In the DX80 products, the Low Pass Filter is a combination of the median filter and the tau filter.)

Temperature Settings

Enable full scale (Temperature)—Turning Fullscale OFF sets the register range of 0x8000 (-32767) through 0x7FFF (+32768) to represent the range of input values. With Fullscale turned OFF, a register value of 1450 represents 72.5 degrees (register values = temperature × 20). With Fullscale turned ON, users can specify the register minimum and maximum range of values. These min/max values are represented in the register as 0 (min) and 65535 (max).

Temperature resolution—Select high to store temperatures values in the registers as the measured temperature $\times 20$. Set to low to store temperature values in tenths of a degree (measured temp $\times 10$). For example, if the measured temperature is 20.5 degrees, turning temperature scaling to high stores the temperature value as 410 while use low resolution stores the temperature as 205.

Units—Select either Celsius or Fahrenheit for your temperature readings.

Outputs

The following parameters are used to configure the outputs, including the switch power outputs.

The output parameters vary, depending on the enabled and available output types. To begin configuring your outputs, **Enable** the output.

Discrete Output

Default Output Value

Select the default output value. When the selected default condition occurs and Hold Last State parameter is set to OFF, this output is set to the selected default output value (e.g. out of sync, communication timeout, start up).

Hold Last State

Retains its last value during the selected default condition (out of sync, communication timeout, start up).

Use Flash Pattern

To use a programmed flash pattern, set the Enable Flash Pattern to ON, then select the appropriate flash pattern from the drop-down list. The flash patterns are defined in the Device Parameters section of this screen. Select from one of the four user-defined flash patterns. Set the flash patterns in the Device Parameters section of this screen.

Switch Power Output

When linking a switch power output to a specific input, select the **Enable** checkbox and set the **Enable default state** to OFF. Use the settings for the specific input to link the switch power output and set the voltage and warm-up time.

For continuous switch power, set the voltage on this screen and set the default state to ON. Verify the default "start-up" conditions are set in the device parameters screens.

Enable default state

When enabled, this switch power output remains on during the selected default condition (e.g. out of sync, communication timeout, start up). When disabled, the switch power cycles off during the selected default condition.

Hold Last Voltage

When set, the switch power output retains its last value during the selected default condition (e.g. out of sync, communication timeout, start up).

Use Flash Pattern

To use a programmed flash pattern, set the Enable Flash Pattern to ON, then select the appropriate flash pattern from the drop-down list. The flash patterns are defined in the Device Parameters section of this screen. Select from one of the four user-defined flash patterns. Set the flash patterns in the Device Parameters section of this screen.

Voltage

To set a voltage for the switch power output, select a value. When configured for continuous voltage output, this switch power output no longer cycles on, warms up the sensors, then cycles back down. Because the output voltage remains constant, continuous voltage is typically used with solar power installations.

Universal Out

After enabling the output, configure the universal output type.

Output is

Select the output type from the drop-down list.

0-20mA

0-10V

Enable register full scale

Turning Fullscale ON sets the entire register range of 0 through 65535 to represent the selected minimum through maximum input values. With Fullscale turned on, a register value of 0 represents the selected minimum value in microamps (for current inputs). A register value of 65535 represents the selected maximum value in microamps. For example, a register value of 0 is 0 and the register value of 65535 represents 20 mA (or 20,000 microamps). With Fullscale turned OFF, the register value represents unit-specific input readings. For units of current (mA), register values are stored as microAmps. Voltage values are stored as millivolts. A sensor reading of 15.53 mA is stored as 15530.

Hold last state (output)

Retains its last value during the selected default condition (out of sync, communication timeout, start up, etc).

H-Bridge Configuration

Enable this output and the H-Bridge configuration to configure its parameters.

Cap warmup time

Similar to the switch power warm-up time, the h-bridge capacitor warm-up time (hours:minutes:seconds) is the time allotted to charge the capacitor used to activate the h-bridge and latching solenoid.

Switch time

Time (hours:minutes:seconds) needed to activate the h-bridge and latching solenoid.

Use Flash Pattern

To use a programmed flash pattern, set the Enable Flash Pattern to ON, then select the appropriate flash pattern from the drop-down list. The flash patterns are defined in the Device Parameters section of this screen. Select from one of the four user-defined flash patterns. Set the flash patterns in the Device Parameters section of this screen.

Voltage

Voltage required to activate the h-bridge and latching solenoid.

2.6.10 Reprogram

Use the DXM Configuration Tool to updating the DXM's firmware. Previous firmware must be version 2.01 or later. For earlier firmware versions, contact Banner Engineering support.

1. Apply power to the DXM,
2. Connect the DXM to your PC using USB or Ethernet.
3. On the menu bar, go to **Device > Connection Settings** and select the appropriate COMM port or enter the IP address.
4. From the **Settings > Reprogram** screen, click **Select upgrade file**.
5. Browse to the firmware HEX file location and select the file.
Depending upon the connection, USB or Ethernet, the programming takes from 5 to 15 minutes to complete.

3 Additional Information

3.1 Contact Us

Corporate Headquarters

Address:
Banner Engineering Corporate
9714 Tenth Avenue North
Minneapolis, Minnesota 55441, USA

Phone: +1 763 544 3164
Website: www.bannerengineering.com

Europe

Address:
Banner Engineering EMEA
Park Lane, Culliganlaan 2F, bus 3
1831 Diegem, Belgium

Phone: +32 (0)2 456 0780
Website: www.bannerengineering.com
Email: mail@bannerengineering.com

Turkey

Address:
Banner Engineering Elk. San. Ve Tic. Ltd. Şti.
Şerifali Mah. Münevver Sok. Ekomed Plaza No:10 Kat:4
Ümraniye / İstanbul, Türkiye

Phone: +90 216 688 8282
Website: www.bannerengineering.com
Email: turkey@bannerengineering.com.tr

India

Address:
Banner Engineering India Pune Head Quarters
Office No. 1001, 10th Floor Sai Capital, Opp. ICC Senapati Bapat Road
Pune 411016, India

Phone: +91 (0) 206 640 5624
Website: www.bannerengineering.com
Email: salesindia@bannerengineering.com

Mexico

Address:
Banner Engineering de Mexico Monterrey Head Office
Edificio VAO Av. David Alfaro Siqueiros No.103 Col. Valle Oriente C.P.66269
San Pedro Garza Garcia, Nuevo Leon, Mexico

Phone: +52 81 8363 2714 or 01 800 BANNERE (toll free)
Website: www.bannerengineering.com
Email: mexico@bannerengineering.com

Brazil

Address:
Banner do Brasil
Rua Barão de Teffé nº 1000, sala 54
Campos Eliseos, Jundiaí - SP, CEP.: 13208-761, Brasil

Phone: +55 11 2709 9880
Website: www.bannerengineering.com
Email: brasil@bannerengineering.com

China

Address:
Banner Engineering Shanghai Rep Office
Xinlian Scientific Research Building Level 12, Building 2
1535 Hongmei Road, Shanghai 200233, China

Phone: +86 212 422 6888
Website: www.bannerengineering.com
Email: sensors@bannerengineering.com.cn

Japan

Address:
Banner Engineering Japan
Cent-Urban Building 305 3-23-15 Nishi-Nakajima Yodogawa-Ku
Osaka 532-0011, Japan

Phone: +81 (0)6 6309 0411
Website: www.bannerengineering.com
Email: mail@bannerengineering.co.jp

Taiwan

Address:

Banner Engineering Taiwan
8F-2, No. 308 Section 1, Neihu Road
Taipei 114, Taiwan

Phone: +886 (0)2 8751 9966

Website: www.bannerengineering.com

Email: info@bannerengineering.com.tw

Index

A

- action rule
 - processing 22–30
- administration 45, 46
- alias registers 48
- analog configuration 50

C

- cap warmup time 52
- cellular firewall 42
- clear HTTP logs 39, 40
- cloud services 36–38
- cloud settings 38
- comms timeout 47
- configuration file
 - save 4
 - upload 4
- configure
 - cloud settings 38
 - local registers 6, 8
- control logic rule
 - create 27
- copy
 - register 24
- count on falling values 49
- count on rising values 49
- create
 - control logic rule 27
 - data log 40
 - dynamic schedule 32
 - event log 40
 - holiday event 31
 - math rule 25
 - one-time event 31
 - read rule 13
 - register copy rule 24
 - threshold rule 24
 - tracker rule 30
 - trending rule 29
 - weekly event 30
 - write rule 15

D

- data log 40
- default conditions 47
- default output value 51
- delta 50
- device restore 48
- DNS 42
- DXM display 11
- DXM driver
 - install manually 3
- dynamic schedule 32

E

- email recipients 44

enable

- default state 51
- full scale (temperature) 50
- register full scale (universal out) 51

- enable Log 39, 40
- enable push on change of state 50
- enable register full scale 50
- encryption 44
- event log 40

F

- filtering 28, 29
- flash pattern 48

G

- general settings 33, 34

H

- h-bridge 52
- hold last state 51
- hold last voltage 51
- holiday event 31
- hysteresis 50

I

- i/o board 46–52
- input configuration 49
- install
 - DXM driver 3
- IP settings 42

L

- local registers
 - configuration 6, 8
 - in use 11
- log file management 39, 40
- logging 39, 40

M

- mail 44
- mail server settings 44
- mapping
 - registers 12–21
- math and logic rule
 - create 25
- median filter 50
- messaging 44
- modify
 - local registers in use 11
 - multiple registers 9, 10
 - registers individually 8

N

- name 47

- network 42
- notes 46

O

- one-time event 31
- out of sync 47

P

- parameters
 - discrete output 51
 - h-bridge 52
 - i/o board 46–52
 - universal out 51
- processing rules 15
- protocol conversion 12

R

- radio polling 34
- read rules
 - create 13
- register copy
 - rule 24
- register mapping 12–21
- register view 32
- registers
 - alias 48
 - modify individually 8
 - modify multiple 9, 10
 - multiple registers 10
 - remap 48
- remap registers 48
- restore
 - device defaults 48
- rule
 - action 22–30
 - binary logic 25–27
 - control logic 25–27
 - filter 28, 29
 - math and logic 25
 - processing 15
 - read 12, 13
 - register copy 24
 - threshold 23, 24
 - tracker 30
 - trending 28, 29
 - write 14, 15

S

- sample high and sample low 50
- sample rate 49
- save
 - configuration file 4
- scheduler 30–32
- scripting 41
- signal conditioning 50

- SMS recipients 44
- start up 47
- startup scripts 41
- switch power output 51
- switch time 52

T

- tau filter 50
- temperature settings
 - full scale 50
 - resolution 50
 - units 50
- threshold 50

- threshold rule
 - create 24
- tracker rule
 - create 30
- trending 28, 29
- trending rule
 - create 29

U

- universal configuration 49
- universal out 51
- upload
 - configuration file 4
- use flash pattern 51, 52

V

- view
 - registers 32
- voltage (h-bridge) 52
- voltage (switch power) 51

W

- weekly event 30
- write rule 14, 15