User Guide

Babel Buster 2

Model BB2-6010-Web
Modbus Web Portal Gateway
Rev. 1.0 – April 2016

BB2-6010-Web User Guide Contents

Babel Buster 2
Modbus
Network Gateway
Model BB2-6010-Web

Control Solutions, Inc.
Minnesota

1. Introduction

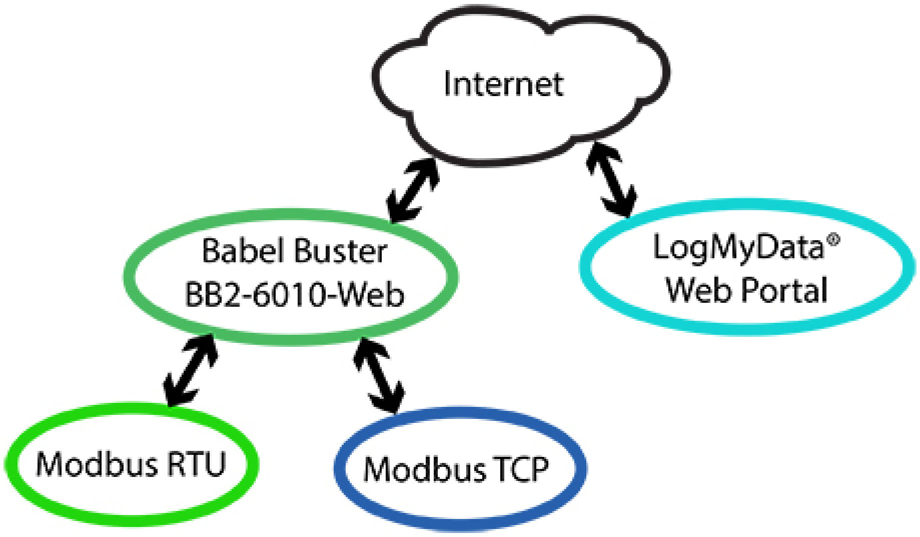
1.1 How to Use This Guide

This user guide provides background information on how the gateway works, and an overview of the configuration process. There are several sections for groups of tabs found in the web interface in the gateway which is accessed by opening a web browser and browsing to the IP address of the device.

You should at least read the configuration sections to gain an understanding of how the gateway is configured. You can use the remaining sections as reference material to look up as needed. There is a "Quick Help" section at the bottom of each web page in the gateway which is generally sufficient for quick reference in setting up the gateway.

1.2 Overview of Gateway Device

The Babel Buster BB2-6010-Web is a special type of Modbus gateway used to query Modbus devices and report their data to a web portal such as LogMyData®.



The Modbus data may be reported "on delta", which means report any change in level outside of a user defined threshold. Modbus data may be reported using its trending feature, which records minimum, maximum, and average over a window and those numbers are reported periodically. Alerts and alarms may also be set up to trigger immediate reporting if a register level goes outside of a user defined limit.

1.3 Important Safety Notice

Proper system design is required for reliable and safe operation of distributed control systems incorporating any Control Solutions product. It is extremely important for the user and system designer to consider the effects of loss of power, loss of communications, and failure of components in the design of any monitoring or control application. This is especially important where the potential for property damage, personal injury, or loss of life may exist. By using ANY Control Solutions, Inc., product, the user has agreed to assume all risk and responsibility for proper system design as well as any consequence for improper system design.

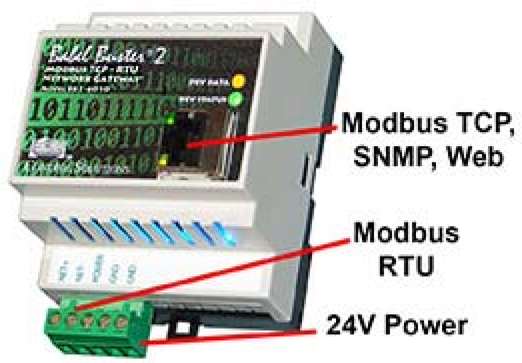
1.4 Warranty

This software and documentation is provided "as is," without warranty of any kind, either expressed or implied, including, but not limited to, the implied warranties of fitness or merchantability for a particular purpose. Control Solutions may make improvements and/or changes in this documentation or in the product(s) and/or the program(s) described in this documentation at any time. This product could include software bugs, technical inaccuracies, typographical errors, and the like. Changes are periodically made to the information herein; these changes may be incorporated in new editions of the software.

2. Connecting the BB2-6010-Web for the First Time

Follow these steps to make the initial connection to the BB2-6010-Web.

(a) Connect power. Apply +12 to +24VDC or 24VAC to the terminal marked “POWER”, and common or ground the the terminal marked “GND”.



(b) Connect a CAT5 cable between the RJ-45 jack on the gateway, and your network switch or hub. You cannot connect directly to your PC unless you use a “crossover” cable.

(c) Apply power.

A blue LED inside the case should light indicating power is present.

If the link LED on the RJ45 jack is not on, check your Ethernet cable connections. Both link and activity LEDs on the RJ45 jack will be on solid for a short time during boot-up. The entire bootup process will take 1-2 minutes, during which time you will not be able to connect with a browser.

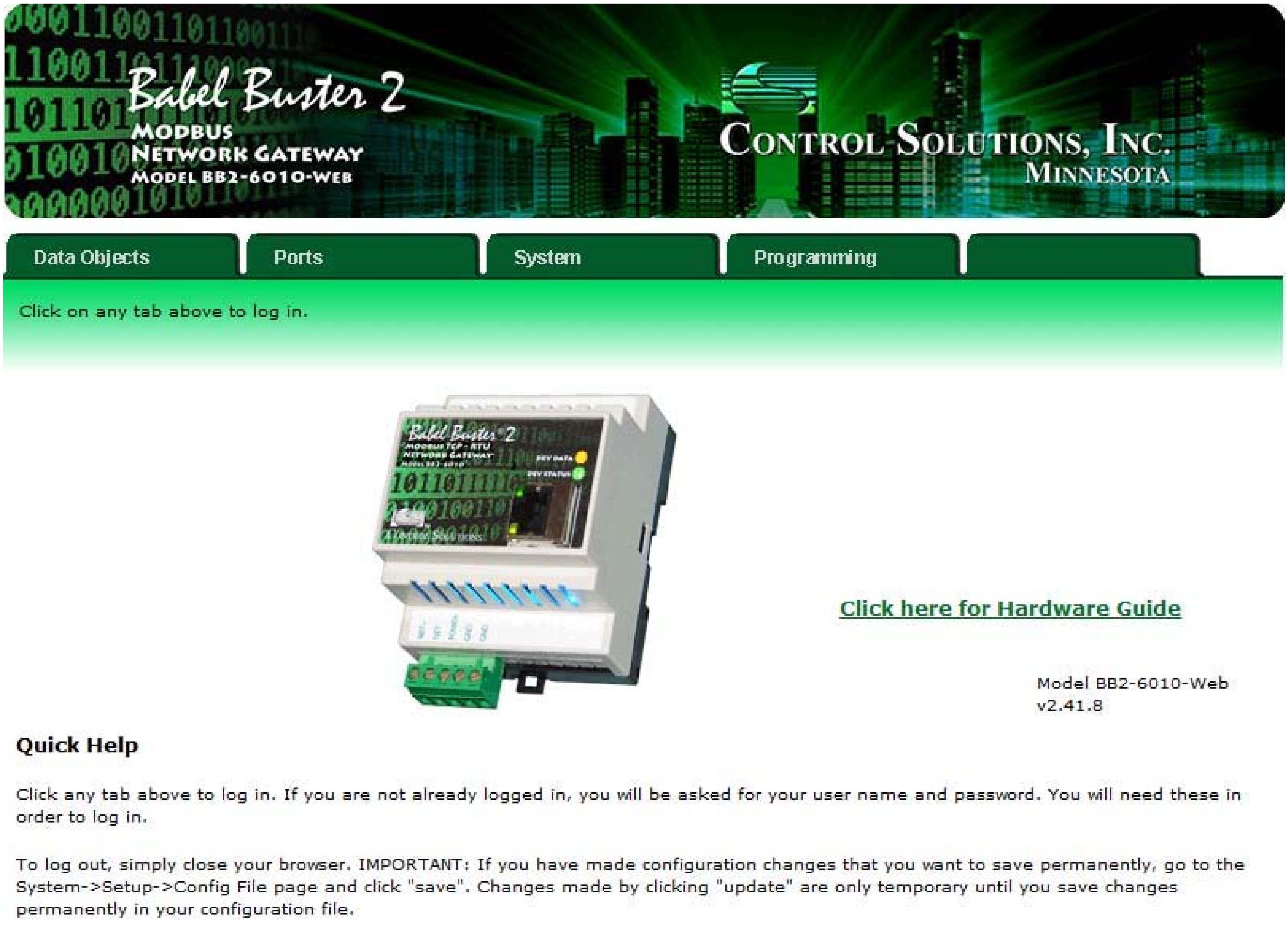
Ethernet link LED is the yellow LED integrated into the CAT5 connector. Ethernet activity LED is the green LED integrated into the CAT5 connector.

(d) The default IP address as shipped is 10.0.0.101. If your PC is not already on the 10.0.0.0 domain, you may need to add a route on your PC (XP only). Do this by opening a command prompt. First type “ipconfig” and note the IP address listed. This is your PC’s IP address. Now type the command

```
route add 10.0.0.0 mask 255.255.255.0 1.2.3.4
```

but substitute your PC’s IP address for 1.2.3.4.

This generally works, but if this fails, you will need to temporarily change your computer’s IP address to a fixed address that starts with 10.0.0. and ends with anything but 101.



Click on any tab above to log in.

Model BB2-6010-Web
v2.41.8

Quick Help

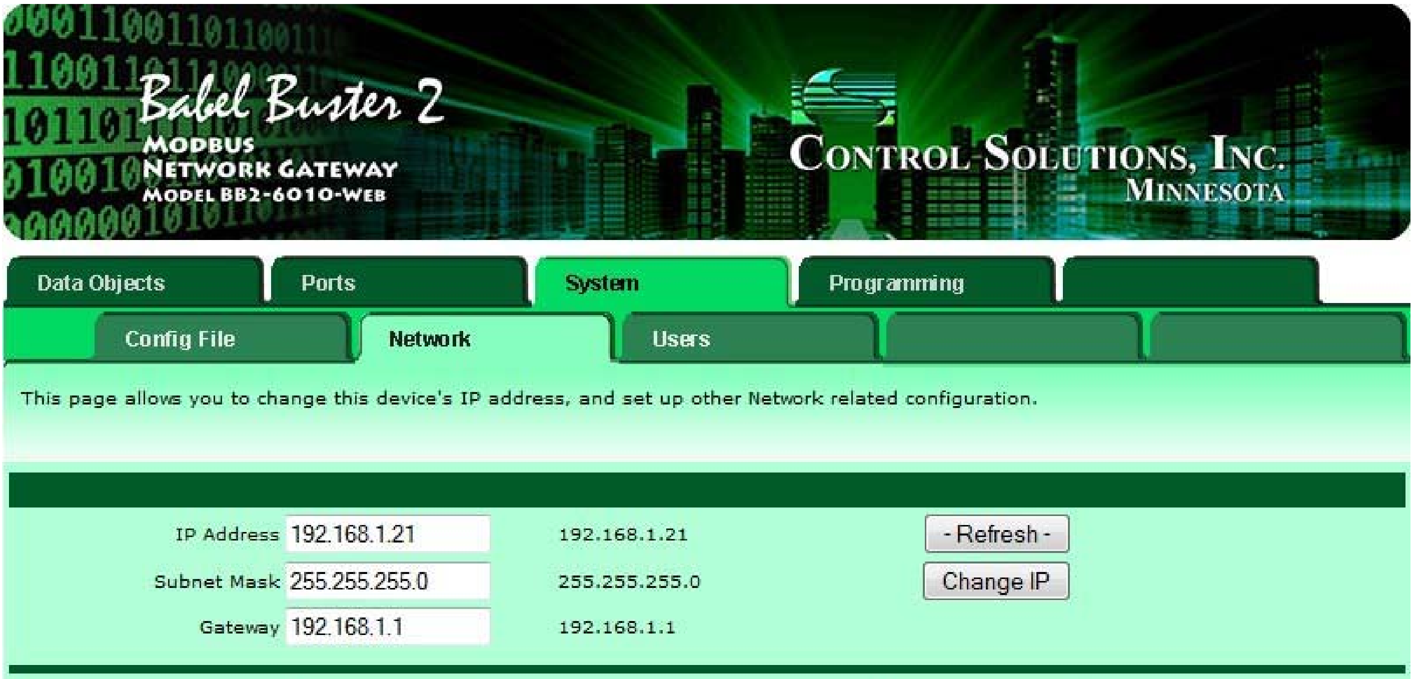
Click any tab above to log in. If you are not already logged in, you will be asked for your user name and password. You will need these in order to log in.

To log out, simply close your browser. IMPORTANT: If you have made configuration changes that you want to save permanently, go to the System->Setup->Config File page and click "save". Changes made by clicking "update" are only temporary until you save changes permanently in your configuration file.

(e) Open your browser, and enter “http://10.0.0.101/” in the address window. You should see a page with the “Babel Buster BB2-6010-Web” header shown above. From this point, you will find help on each page in the web site contained within the product.

(f) When you click on any of the page tabs such as System, you will be asked for a user name and password. The default login is user name “system” with password “admin”. You can also log in as “root” using password “buster”. You must log in as “root” if you will be changing the IP address.

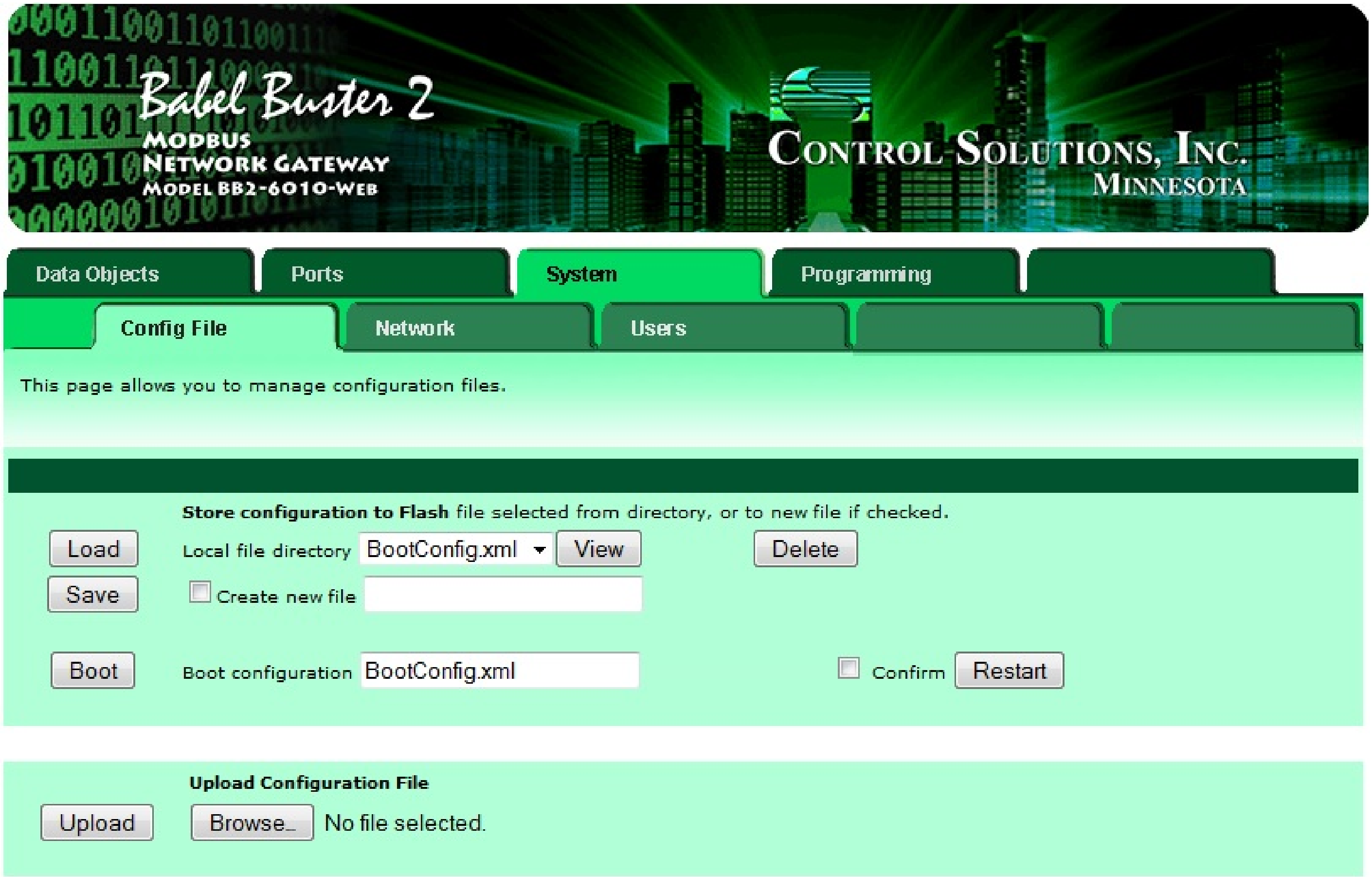
(g) To change the IP address of the gateway, go to the Network page under System :: Setup. The following page should appear. Change the IP address, and subnet mask and gateway if applicable. Click Change IP to save the changes. The process of programming this into Flash takes around half a minute. The new IP address only takes effect following the next system restart or power cycle. (Note: There are additional network settings that will need to be made before the BB2-6010-Web will begin reporting data to the web portal. These are covered in later sections.)



This page allows you to change this device's IP address, and set up other Network related configuration.

IP Address	192.168.1.21	192.168.1.21
Subnet Mask	255.255.255.0	255.255.255.0
Gateway	192.168.1.1	192.168.1.1

(h) Most changes are stored in an XML configuration file in the device’s Flash file system. Only a few are stored differently, and the IP address is one of those. Normally, clicking Update on any configuration page only stores that configuration information to a temporary RAM copy of the configuration file. To make your changes other than IP address permanent, you must click Save on the Config File page (System :: Setup :: Config File).



This page allows you to manage configuration files.

Store configuration to Flash file selected from directory, or to new file if checked.

Local file directory: BootConfig.xml

□ Create new file

Boot configuration: BootConfig.xml

□ Confirm

Upload Configuration File

No file selected.

3. Minimum BB2-6010-Web Setup

This user guide contains a lot of information to digest. To help sort through which parts are required reading, this brief section is provided as a checklist or overview of those areas that will need attention in order to get the BB2-6010-Web to start sending Modbus data to the web portal.

(a) Configure the Modbus port (section 4 or 5 as applicable).

(b) Configure one or more data objects to receive Modbus data (section 6). It is a good idea to review both sections 6 and 7 before proceeding with configuration.

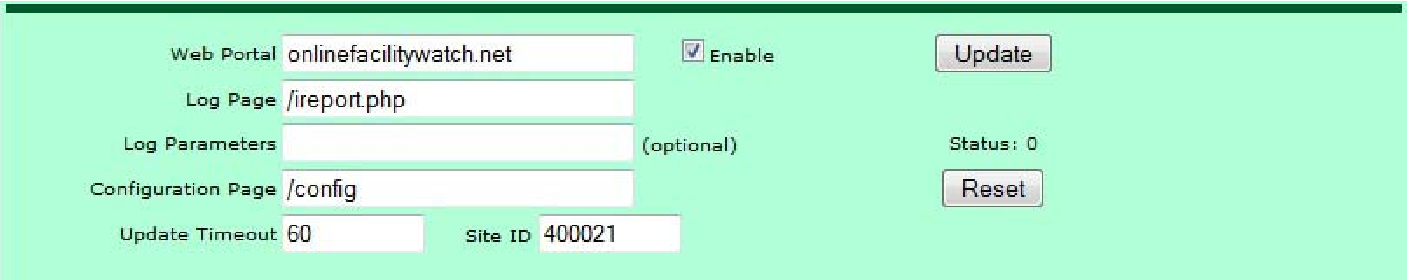
(c) Configure at least one report group (section 7) and be sure to reference that report group in objects configured as discussed section 6. Data objects that do not reference a valid report group will not report anything to the web portal.

(d) Export a copy of your configuration and upload it to your account on the web portal (section 8.1).

(e) Configure the Network parameters (section 8.2). You will have probably already set the IP address of the BB2-6010-Web by now. But you must also provide a valid DNS server before the BB2-6010-Web will be able to find the web portal. And of course, you must tell the BB2-6010-Web which server your account is found on and what this device's Site ID is.

Once you have gone live, you will be expected to only make subsequent configuration changes via the web portal. This is required in order to keep your device and your web portal account in sync. If you were permitted to make local changes unknown to the web portal after going live, both you and the server will end up confused about what data you are sending. If you later take the BB2-6010-Web offline and make additional configuration changes, you should repeat the process of exporting the configuration file and uploading it to the web portal.

IMPORTANT: You are "Live" any time the Enable box is checked on the Network setup page. DO NOT check this box until after you have done any pre-configuration you wish to do, and have uploaded that configuration to the web portal.



Web Portal onlinefacilitywatch.net ☑ Enable

Log Page /ireport.php

Log Parameters (optional)

Status: 0

Configuration Page /config

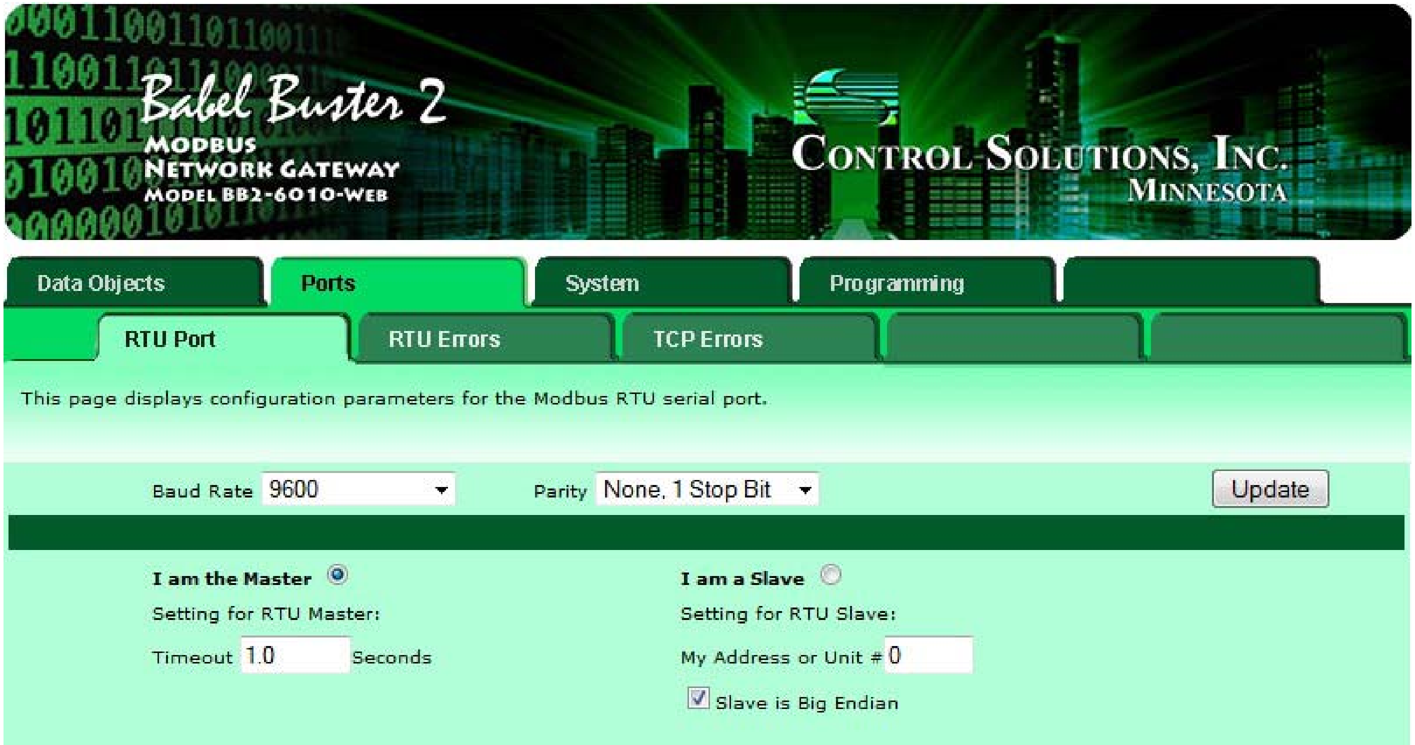
Update Timeout 60 Site ID 400021

NOTE: It is not necessary to do any pre-configuration of objects or reporting groups before going "Live". You would then do all configuration for the first time via the web portal. If you do not wish to do any pre-configuration, proceed directly to Network configuration (section 8.2).

4. Configuring BB2-6010-Web as Modbus Master

4.1. RTU Port Settings

Select the baud rate that your RTU network is running at. It is rare for RTU to run at anything other than N81, so leave that setting for parity if uncertain. Select "I am the Master". Be sure to enter a timeout greater than zero. If the slave device does not respond within this time, it will be treated as an error even if the slave did respond later.



This page displays configuration parameters for the Modbus RTU serial port.

Baud Rate 9600 Parity None, 1 Stop Bit

I am the Master ●

Setting for RTU Master:

Timeout 1.0 Seconds

I am a Slave ○

Setting for RTU Slave:

My Address or Unit # 0

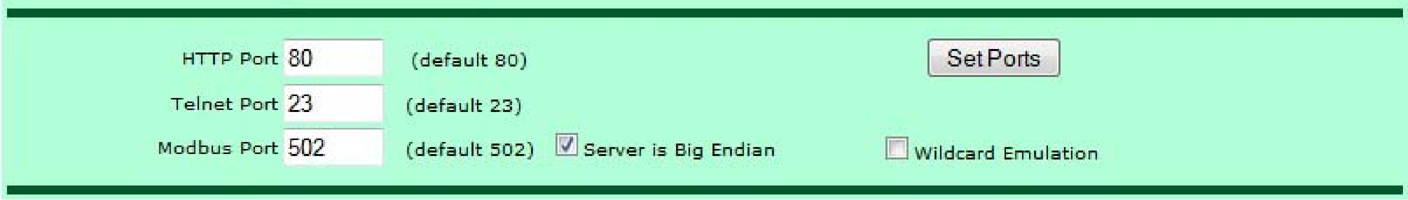
☑ Slave is Big Endian

There is an option for "RTU disabled" under baud rate settings. If RTU is disabled, the RS-485 port is then available to Script Basic for use as a generic serial communications port, making it possible to write a program to interface with a proprietary serial protocol.

4.2. TCP Port Settings

Modbus TCP is inherently both Master and Slave at the same time, so there is no selection needed for master versus slave. The only thing to be certain of is that the

Modbus Port on the Network page is set to a non-zero value, usually 502 (the standard Modbus TCP port). If set to zero, Modbus TCP will be disabled after the next restart. This ability is included for security conscious users that wish to disable Modbus TCP access via the network.



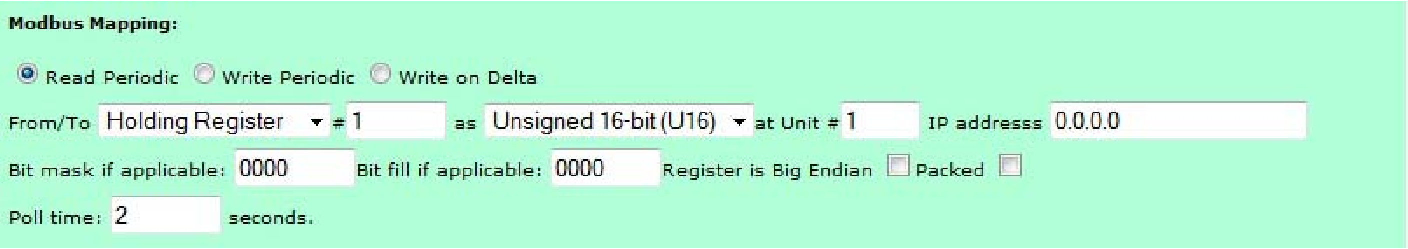
HTTP Port 80 (default 80)

Telnet Port 23 (default 23)

Modbus Port 502 (default 502) ☑ Server is Big Endian □ Wildcard Emulation

4.3. Mapping Objects for Modbus Master

You will need to enter appropriate parameters in the Modbus Mapping section of objects that should generate polls to Modbus slaves. The mapping section of the objects is what causes the actual polling of Mobus slaves. Full details on mapping configuration are provied in Section 6.



Modbus Mapping:

● Read Periodic ○ Write Periodic ○ Write on Delta

From/To Holding Register # 1 as Unsigned 16-bit (U16) at Unit # 1 IP address 0.0.0.0

Bit mask if applicable: 0000 Bit fill if applicable: 0000 Register is Big Endian □ Packed □

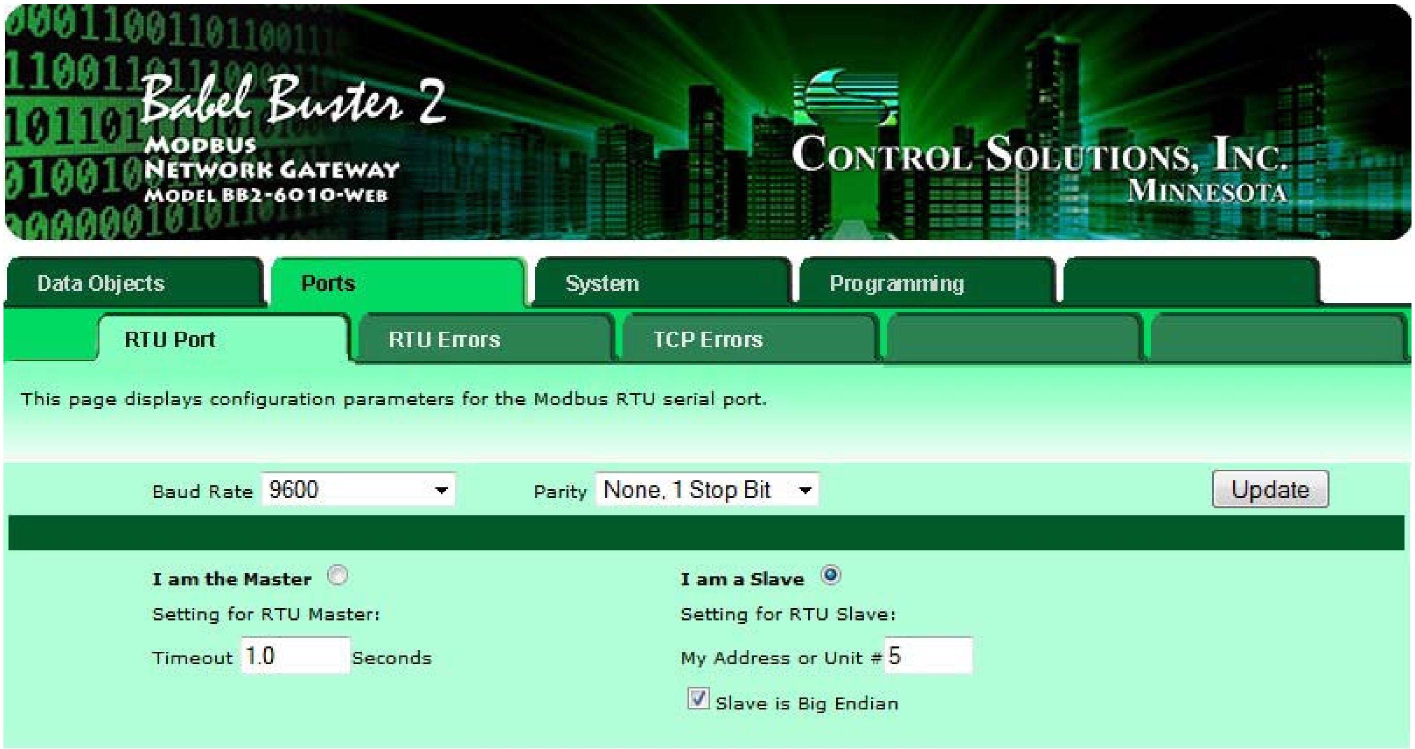
Poll time: 2 seconds.

If you wish to use a data object for something other than Modbus (e.g. data is generated by a Script Basic program), then be sure to select "None" for register type, and "None" for data format in the Modbus Mapping section. Selecting "None" will cause the Modbus engine to skip this object in its poll scan.

5. Configuring BB2-6010-Web as Modbus Slave

5.1. RTU Port Settings

Select the baud rate that your RTU network is running at. It is rare for RTU to run at anything other than N81, so leave that setting for parity if uncertain. Select "I am a Slave". Be sure to enter a slave address (or unit number) to which this device should respond on the network.



This page displays configuration parameters for the Modbus RTU serial port.

Baud Rate 9600 Parity None, 1 Stop Bit

I am the Master ○

Setting for RTU Master:

Timeout 1.0 Seconds

I am a Slave ●

Setting for RTU Slave:

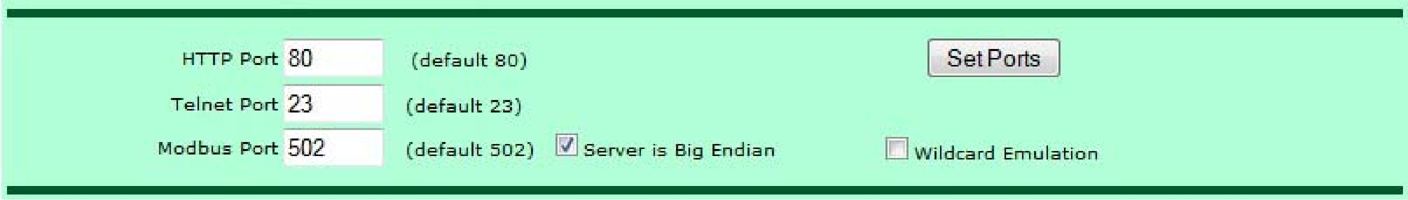
My Address or Unit # 5

☑ Slave is Big Endian

The "Big Endian" selection does not matter if you are reading coils, discrete inputs, or any input or holding register that is a single 16-bit register. However, in order to read 32-bit integers or floating point (also 32-bit) via Modbus, you will actually be reading two consecutive 16-bit registers that are concatenated to form a 32-bit value. However, the order in which those two registers should be interpreted (i.e. most significant data first or second) is not standardized. Therefore, you have the ability to change the BB2-6010-Web to work either way as needed. Select "Slave is Big Endian" if the most significant or high order data should be presented in the first register (or lower numbered register of the pair).

5.2. TCP Port Settings

Modbus TCP is inherently both Master and Slave at the same time, so there is no selection needed for master versus slave. The only thing to be certain of is that the Modbus Port on the Network page is set to a non-zero value, usually 502 (the standard Modbus TCP port). If set to zero, Modbus TCP will be disabled after the next restart. This ability is included for security conscious users that wish to disable Modbus TCP access via the network.



HTTP Port 80 (default 80)

Telnet Port 23 (default 23)

Modbus Port 502 (default 502) ☑ Server is Big Endian ☐ Wildcard Emulation

5.3. Mapping Objects for Modbus Slave

The Modbus Mapping section found in the configuration of each object only applies when the BB2-6010-Web is Modbus master. You can disregard this area when configuring as a Modbus slave.

All data objects known to the BB2-6010-Web are always accessible when it is operating as a Modbus slave. Refer to the register map below to determine register numbers to use to read and write the objects from some other Modbus master.

5.4. Modbus Slave Register Map

All of the data objects in the BB2-6010-Web may be read in multiple ways by an external Modbus RTU master when the BB2-6010-Web is configured as slave, and at any time by Modbus TCP when the Modbus TCP port is enabled. Data objects may be read as 16-bit integer, 32-bit integer, or floating point (IEEE 754 32-bit), and these may be read as input registers or holding registers. These data formats may only be written as holding registers. In addition, all data objects can be read as coils or discrete inputs, and written as coils, although the data will only be zero or one in that case. (If an object with a value greater than 1 is read as coil or discrete input, it will be returned as 1 since values are only a single bit for these register types.)

The data format of the object in the BB2-6010-Web does not restrict how it is read via Modbus. Any necessary data conversion is performed on the fly, such that floating point objects may be read as 16-bit integer, etc. One must be aware, however, that errors could occur in data values if a floating point whose value exceeds the range of a 16-bit integer is read as a 16-bit value.

The register numbers used to access data objects are calculated according to how you want to access the objects. In addition, a number of "virtual" registers exist for checking alert and alarm states.

Object number N is used to calculate register numbers for accessing object data (1 <= N <= 400)

Register 1 to N = 16-bit integer presentation of the object data
Register 1001 to 1000 + N = unsigned 16-bit integer presentation of data
Register 2001 to (2000 + N * 2 - 1) = floating point presentation of object data (accessible as register pair only)
Register 3001 to (3000 + N * 2 - 1) = 32-bit integer presentation of object data (accessible as register pair only)

Note that only registers 1 to N are valid for access to objects as coils or discrete inputs.

Virtual registers return only a zero or one indicating alert or alarm inactive or active. For purposes of reporting to the web portal, these are redundant since alerts and alarms are already reported. These registers are primarily useful to Script Basic programs for checking alert/alarm status. These register numbers are calculated as follows:

Register 4000 + object # = alert state (0=inactive, 1=alert active)
Register 5000 + object # = alarm state (0=inactive, 1=alert active)
Register 6000 + object # = alarm OR alert state (0=inactive, 1=alert or alarm is active)

Modbus error codes are directly accessible to Script Basic using virtual registers as follows. These are accessible only as integers. Registers beyond 8001 are not useful unless the BB2-6010-Web is functioning as Modbus master, but are included here for reference.

8001-8246 -- Modbus error status by slave ID when BB2-6010-Web is master
8001 -- Modbus error status when BB2-6010-Web is slave

Virtual register value for Modbus error status registers:

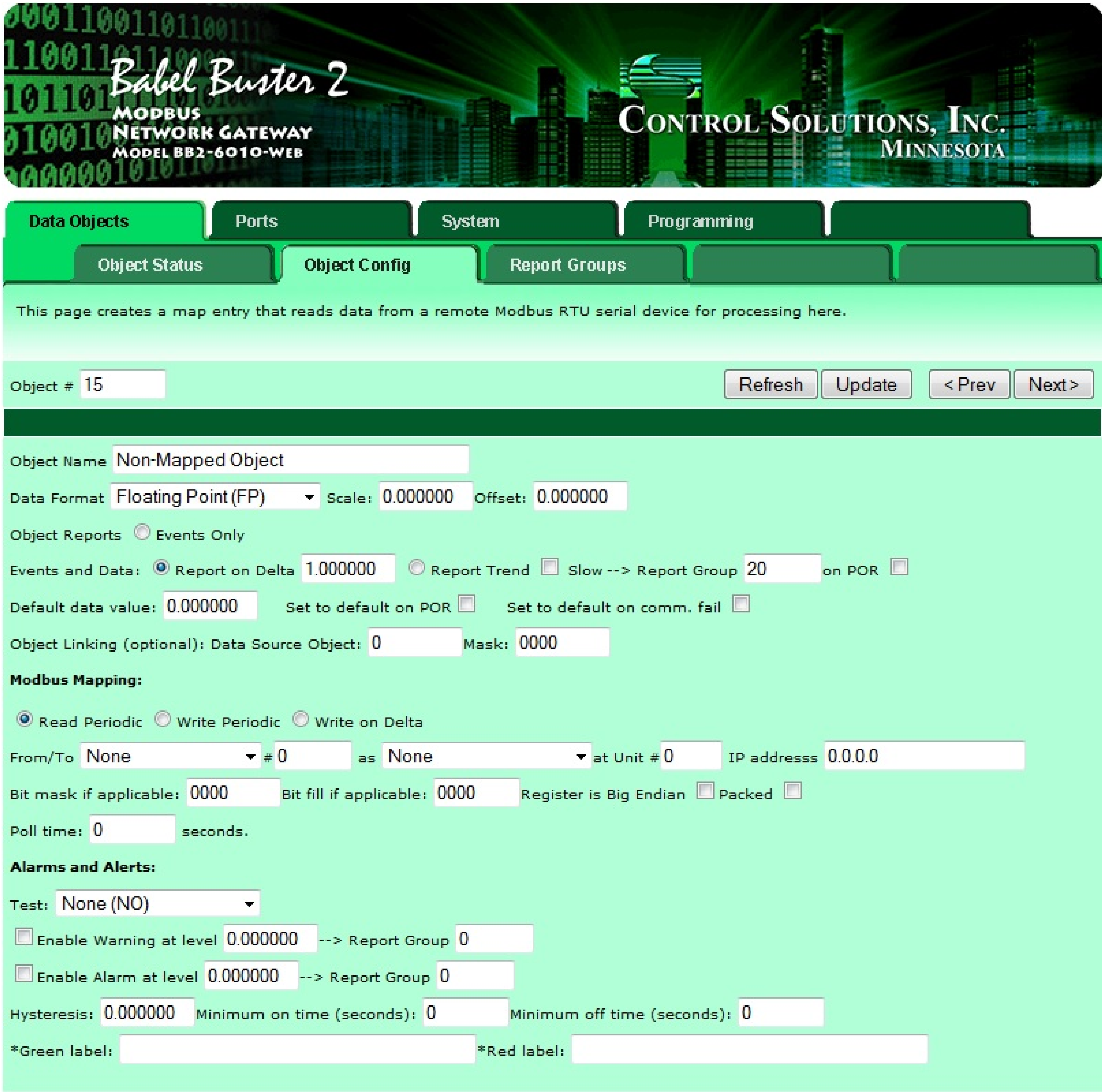
0..127 = exception code
0x80 (128) = configuration error (128)
0x81 (129) = no-responce count is at maximum (15)
0x82 (130) = CRC error count is at maximum (15)
0x83 (131) = both no-response and CRC are at maximum count

6. Configuring Data Objects for Reporting to Web Portal

6.1. Definition of Data Object

The BB2-6010-Web contains a set of 400 data objects which form the core elements responsible for gathering data and sending it to the web portal. Each object is accessible to external Modbus masters, can poll Modbus slaves, and is accessible to user written Script Basic programs. Each object can be set up to report changes in data, report data trends, and report alerts and alarms.

The minimum requirement for configuring a data object is to give it a name, and select its data format. If an object is given a name and format other than "None", then it will be included and displayed on the web portal when the configuration file from the BB2-6010-Web is uploaded to the web portal.



This page creates a map entry that reads data from a remote Modbus RTU serial device for processing here.

Object # 15

Object Name Non-Mapped Object

Data Format Floating Point (FP) Scale: 0.000000 Offset: 0.000000

Object Reports ○ Events Only

Events and Data: ● Report on Delta 1.000000 ○ Report Trend □ Slow --> Report Group 20 on POR □

Default data value: 0.000000 Set to default on POR □ Set to default on comm. fail □

Object Linking (optional): Data Source Object: 0 Mask: 0000

Modbus Mapping:

● Read Periodic ○ Write Periodic ○ Write on Delta

From/To None # 0 as None at Unit # 0 IP addresss 0.0.0.0

Bit mask if applicable: 0000 Bit fill if applicable: 0000 Register is Big Endian □ Packed □

Poll time: 0 seconds.

Alarms and Alerts:

Test: None (NO)

□ Enable Warning at level 0.000000 --> Report Group 0

□ Enable Alarm at level 0.000000 --> Report Group 0

Hysteresis: 0.000000 Minimum on time (seconds): 0 Minimum off time (seconds): 0

*Green label: *Red label:

Scale and offset are optional. If a non-zero value is provided, then the "raw data" is multiplied by the scale factor and added to offset before being saved in this object as its final data value. The raw data will typically be contents of a Modbus register read from a Modbus slave. Scale and offset will also be applied when the raw data source is a linked object, but only if the object data format is floating point. When the object is mapped to write to a Modbus register, then the scaling process is reversed.

An object can be configured to exist by selecting a data format other than "None", yet be configured to not report anything to the web portal. If an object is configured to report "Events Only" but the alarms and alerts test is set to "None", then no reporting can occur. Data will be sent to the web portal in one (or more) of three ways: (1) Report on Delta; (2) Report Trend ; (3) Report Events Only with alarms and alerts defined. Alerts and alarms are automatically included in Report on Delta or Trend if the alarm test is defined as anything other than "None".

Report on Delta means report changes in data value. The window next to "Report on Delta" is where you enter the margin of change that should be reported. Report on delta of zero will report any change. Report on delta of "5.0" would report any change in value of 5 or greater, and so on. If you set Report on Delta of zero, you should be aware that a lot of reporting may occur, possibly more than you really want. You have the ability to throttle the reporting rate via configuration of the report group. You also have the option of forcing periodic reporting at some minimum rate even if there is no change in value, and this is also configured in the report group settings.

Report Trend results in capturing minimum, maximum, and average value over some window of time. The window of time, and the sample rate within that window, is configured in the report group. At the end of each window period, the accumulated data is transmitted to the web portal. Each data set transmitted includes minimum, maximum, and average, in that order, over the trend window period.

The trending will most often apply to floating point values (average temperature, average pressure, etc.), or possibly integer values. If the data object is defined as "Boolean (BIT)", then the trend action takes on a new meaning. Since there is no meaningful average of a series of on/off states, the trending for a boolean object will log last off time, last on time, and total on time. Time here means duration, not time of day. Therefore if the numbers reported were "5,12,82", and the boolean object was tracking on/off status of a motor for example, these numbers would say the motor was most recently off for 5 seconds, on for 12 seconds, and had run for a total of 82 seconds since the last report. Now obviously seeing a motor run for seconds is not practical. That is where the "Slow" check box comes into play. This box only applies when the object is defined as boolean, and results in times being reported in minutes rather than seconds.

A Report Group is required for either type of data reporting. Without a valid report group defined and referenced here, no data will be sent to the web portal. The report group defines the time constraints on data reporting and effectively enables reporting.

The "On POR" following the report group only applies to Report on Delta, and should be checked if you want the BB2-6010-Web to report its value immediately upon startup rather than waiting for the first change in data.

The Default Value is an optional value that you may force this object to be set to upon either of the two conditions noted by the check boxes that follow the default value. Set to default on POR means immediately set this object to the default value upon startup. Set to default on comm. fail means set this object to the default value in the event Modbus polling fails (this option only applies if the object is mapped to poll a Modbus register, and the BB2-6010-Web is acting as Modbus master).

Object linking is discussed in its own section below.

6.2. Modbus Mapping

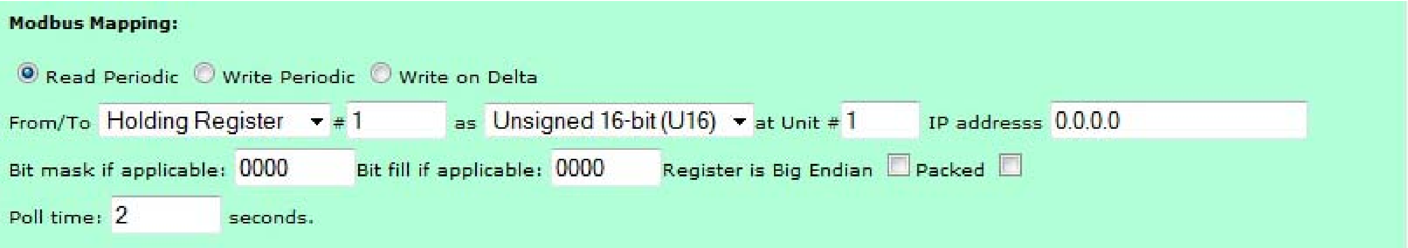
The BB2-6010-Web is primarily a Modbus to web portal gateway, and as such, most objects are going to be configured to query Modbus devices. The Modbus Mapping section of the data object is where this action is configured. If the BB2-6010-Web is to be only a Modbus slave, then skip this section and be sure to leave the selections here set to "None". The discussion that follows applies when the BB2-6010-Web will be Modbus master.

The most frequent application will be to read a register from a Modbus slave. Selet Read Periodic. Then select the register type (e.g. holding register), register number, data format (e.g. unsigned 16-bit), and slave address.

Important note about Modbus register numbers: If your Modbus device documentation refers to its first register as 40001, it is using an older convention for documenting registers. The offset of 40000 is short hand for telling you it is a holding register. The register number in this case is actually 1. To configure the BB2-6010-Web, do not enter 40001. Select "holding register" and enter the actual register number of just "1". Another source of confusion is whether the documentation gives register numbers or raw addresses. A register number of 1 will be transmitted on the wire as 0. If the documentation said 40001, what gets transmitted in the raw binary address field is 0. If the documentation give an address of 0 for a register, then add 1 to all addresses in that document to arrive at the correct register number to enter here.

Modbus input and holding registers are always exactly 16 bits. If your device provides a 32-bit integer or floating point value, then Modbus is actually going to automatically read two consecutive registers to produce the entire 32-bit value. The order in which these two registers appear in the slave is not standardized. Therefore, you have the option of selecting the order. Check the "Big Endian" box if the high order data comes first, i.e. most significant bits are in the first or lower numbered register.

When doing periodic reads, you need to specify a poll time in seconds. The register will be read this often. If all poll times are zero, then the BB2-6010-Web will simply poll as fast as it can, but will always poll in round robin fashion. If there is a very long list of registers to poll and all have been assigned a short poll time, this poll time will not necessarily be honored. Polling is always done in round robin fashion. The BB2-6010-Web will poll as fast as possible but pause and hold off if the poll time has not expired by the time polling comes back around to this object.



Modbus Mapping:

● Read Periodic ○ Write Periodic ○ Write on Delta

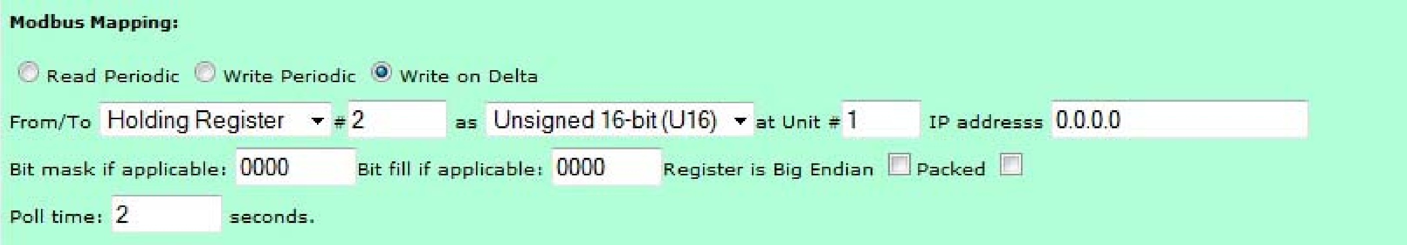
From/To Holding Register ▾ # 1 as Unsigned 16-bit (U16) ▾ at Unit # 1 IP addresss 0.0.0.0

Bit mask if applicable: 0000 Bit fill if applicable: 0000 Register is Big Endian □ Packed □

Poll time: 2 seconds.

Modbus registers may also be written by the BB2-6010-Web. Select Write Periodic if the register should simply be rewritten at a regular pace defined by the poll time. The more useful and more often used option is Write on Delta. This means the Modbus register will only be written when the data object changes value (e.g. new value sent from the web portal).

Select a register type, register number, register data format, and slave address just as you would for reading a register. However, in this case, be sure the register type is writeable. Modbus input registers are "read only" and your slave device will return an exception error if you try to write to an input register.



Modbus Mapping:

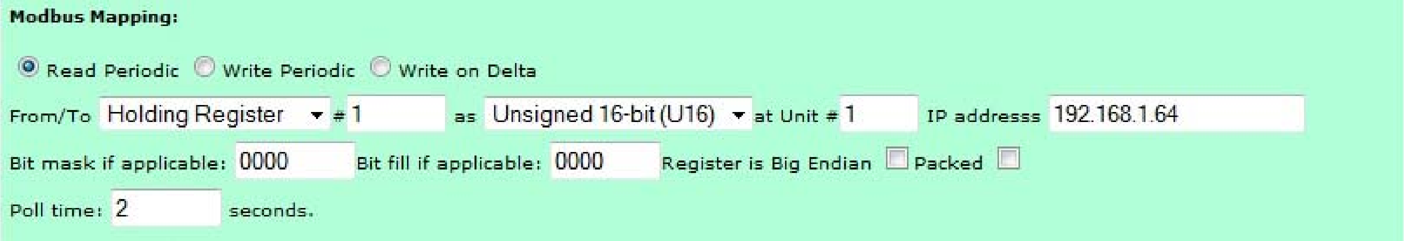
○ Read Periodic ○ Write Periodic ● Write on Delta

From/To Holding Register ▾ # 2 as Unsigned 16-bit (U16) ▾ at Unit # 1 IP address 0.0.0.0

Bit mask if applicable: 0000 Bit fill if applicable: 0000 Register is Big Endian □ Packed □

Poll time: 2 seconds.

The "Unit #" is the slave address for Modbus RTU. If this object should poll Modbus RTU, then leave the IP address set to 0.0.0.0. However, if this object should poll Modbus TCP instead, then enter the IP address of the Modbus TCP slave. Entering a non-zero IP address will switch polling from RTU to TCP.



Modbus Mapping:

● Read Periodic ○ Write Periodic ○ Write on Delta

From/To Holding Register ▾ # 1 as Unsigned 16-bit (U16) ▾ at Unit # 1 IP address 192.168.1.64

Bit mask if applicable: 0000 Bit fill if applicable: 0000 Register is Big Endian □ Packed □

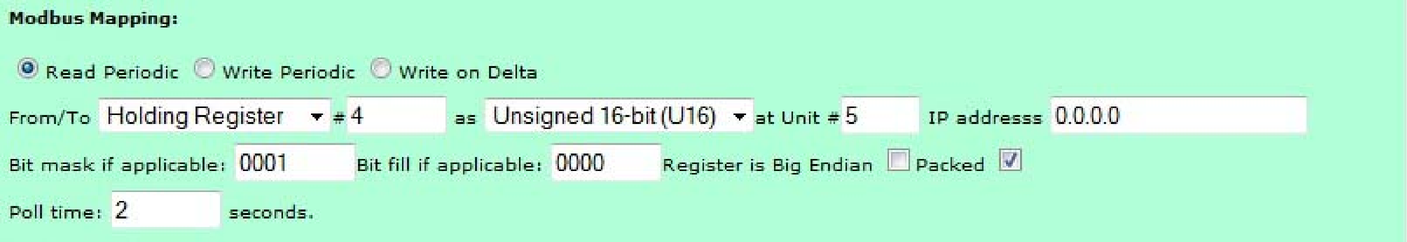
Poll time: 2 seconds.

It is common for Modbus devices to "pack" a bunch of status bits into a single holding register. For alarm reporting purposes, this single conglomerated value is not very useful. To report alarms upon the occurrence of a specific status bit, you need the ability to mask the bits. This feature is provided in each data object. If you enter a hexadecimal bit mask, the contents of the holding register (or input register) referenced will be logically AND-ed with the mask, and the result will be right justified. If a single bit is given in the mask, then the resulting object value will always be zero or one.

The mask can be multiple bits. Sometimes two 8-bit values are packed into a single holding register. In this case, a mask of FF00 would get the first value, and 00FF would get the second value.

You have the ability to make Modbus polling more efficient by reading the packed register once, and then distributing its contents to multiple data objects. Do this by defining all of the data objects belonging to the same holding register in contiguous order, and then check the Packed box in each of them. If the Packed box is not checked, or the objects are not contiguous, then the same register will be read multiple times, once for each mask value.

You may also write to packed registers. The mask is defined the same way. In this case, the value of the object will be shifted into the position indicated by the mask, and logically OR-ed into the accumulated value ultimately written to the Modbus slave. The optional bit fill value is only used when writing a packed register. If any bits are set in the fill mask, then these are logically OR-ed into the value written, meaning these bits will always be unconditionally set in the Modbus register. It is very important that objects be contiguous and the Packed box be checked when writing a packed register. Multiple reads will only result in inefficient polling, but multiple writes to the same register will result in erratic and undesired behavior in the Modbus slave being written.



Modbus Mapping:

● Read Periodic ○ Write Periodic ○ Write on Delta

From/To Holding Register # 4 as Unsigned 16-bit (U16) at Unit # 5 IP addresss 0.0.0.0

Bit mask if applicable: 0001 Bit fill if applicable: 0000 Register is Big Endian □ Packed ☑

Poll time: 2 seconds.

The bit mask shown in the Modbus Mapping is a 4 digit hexadecimal (16 bit) value used to mask out one or more bits in a register. The selected bits will be right justified, so a single bit regardless of where positioned in the source register will be stored locally as 0 or 1. The hex bit mask values would be as follows:

B0/D0/bit 0 mask = 0001
B1/D1/bit 1 mask = 0002
B2/D2/bit 2 mask = 0004
B3/D3/bit 3 mask = 0008
B4/D4/bit 4 mask = 0010
B5/D5/bit 5 mask = 0020
B6/D6/bit 6 mask = 0040
B7/D7/bit 7 mask = 0080
B8/D8/bit 8 mask = 0100
B9/D9/bit 9 mask = 0200
B10/D10/bit 10 mask = 0400
B11/D11/bit 11 mask = 0800
B12/D12/bit 12 mask = 1000
B13/D13/bit 13 mask = 2000
B14/D14/bit 14 mask = 4000
B15/D15/bit 15 mask = 8000

Some Modbus devices also back two 8-bit values into a single 16-bit register. The two values will typically be documented as “high byte” and “low byte” or simply have “H” and “L” indicated. If you run into this scenario, the masking for bytes is as follows:

High byte mask = FF00
Low byte mask = 00FF

When the mask value in a BB2-6010-Web is more than just one bit, the mask is still logicalle AND-ed with the data from the Modbus slave, and the entire resulting value is right justified to produce an integer value of less than the original bit width of the original register.

There have been a few instances of documenting packed bits in a 32-bit register. Although Modbus protocol is strictly 16-bit registers, some implementations force you to read pairs of registers. If your device documents 32 packed bits, then you would insert 0000 in front of each mask above, and the remainder of the list would be as follows:

B16/D16/bit 16 mask = 00010000
B17/D17/bit 17 mask = 00020000
B18/D18/bit 18 mask = 00040000
B19/D19/bit 19 mask = 00080000
B20/D20/bit 20 mask = 00100000
B21/D21/bit 21 mask = 00200000
B22/D22/bit 22 mask = 00400000
B23/D23/bit 23 mask = 00800000
B24/D24/bit 24 mask = 01000000
B25/D25/bit 25 mask = 02000000
B26/D26/bit 26 mask = 04000000
B27/D27/bit 27 mask = 08000000
B28/D28/bit 28 mask = 10000000
B29/D29/bit 29 mask = 20000000
B30/D30/bit 30 mask = 40000000
B31/D31/bit 31 mask = 80000000

When the data object is defined as 32 bits, and the Modbus Mapping is configured to read (or write) 32 bits, then mask values are processed as 32 bits.

6.3. Alarms and Alerts

One of the most useful features of the BB2-6010-Web combined with its web portal support is the ability to report alarms. Each data object has a 2-level alarm associated with it. To activate an alarm, select a test other than "None", provide thresholds as applicable, and provide a report group. An alarm without a valid report group will not be reported.

Alarms are defined as "warning" and "alarm". The intent is that you should receive a warning (or pre-alarm) first, then an alarm if the condition escalates. Therefore, thresholds should be selected such that the warning occurs first (if used).

Test determines how the data object's present value will be compared to the threshold given. If the test is "Reg > Thresh", that basically defines a high alarm, meaning the alarm will be triggered when the object's present value goes above the threshold. If the test is "Reg < Thresh", then you have a low alarm, meaning the alarm will be triggered when the object's present value goes below the given threshold.

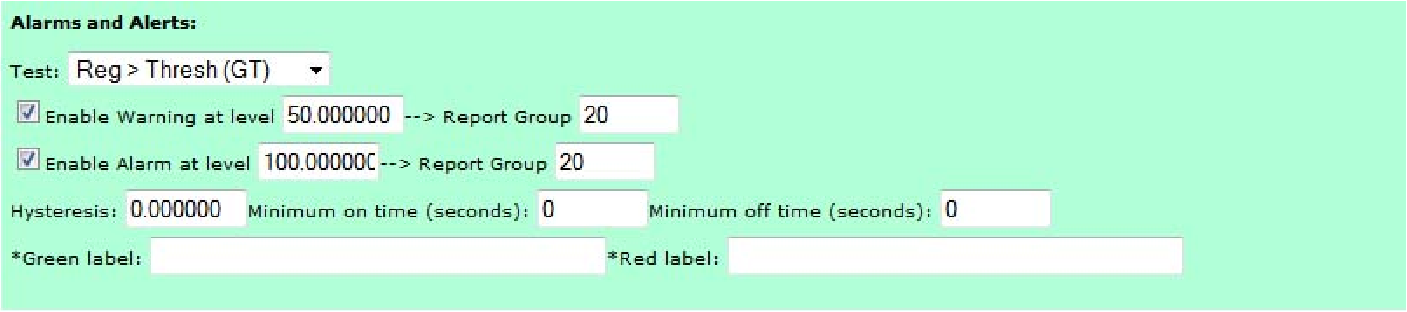
You can also trigger an alarm when value equals the threshold; however, this would not be practical for an analog gauge value such as pressure. The "Reg = Thresh" would typically only be used for a Boolean type object to check for a status bit being on (versus off), and you would then typically only enable just the alarm (leave warning disabled) since there are only two possible values.

You have the option of enabling the warning, alarm, or both. One or both must be selected before any notification will be generated.

Hysteresis provides a means of preventing multiple spurious alarms when a value is hovering near the threshold. If Hysteresis is set to something greater than zero, then once triggered, the present value must move away from the threshold by this amount before the warning or alarm will be reset.

Minimum on time means the condition must exist for at least this long before it will be reported. Minimum off time means the condition must not exist for at least this long before the triggered alarm will be cleared or reset.

Labels are optional and are not used locally by the BB2-6010-Web. These fields are only used for pre-configuring the device, and will be included in the configuration file uploaded to the web portal when the device is first commissioned. Once loaded into the web portal, they may be changed. Changes made to labels on the web portal are not reflected here (all other changes will be reflected here). These labels are only used for the automated authoring of emails and text messages on the web portal.



Alarms and Alerts:

Test: Reg > Thresh (GT)

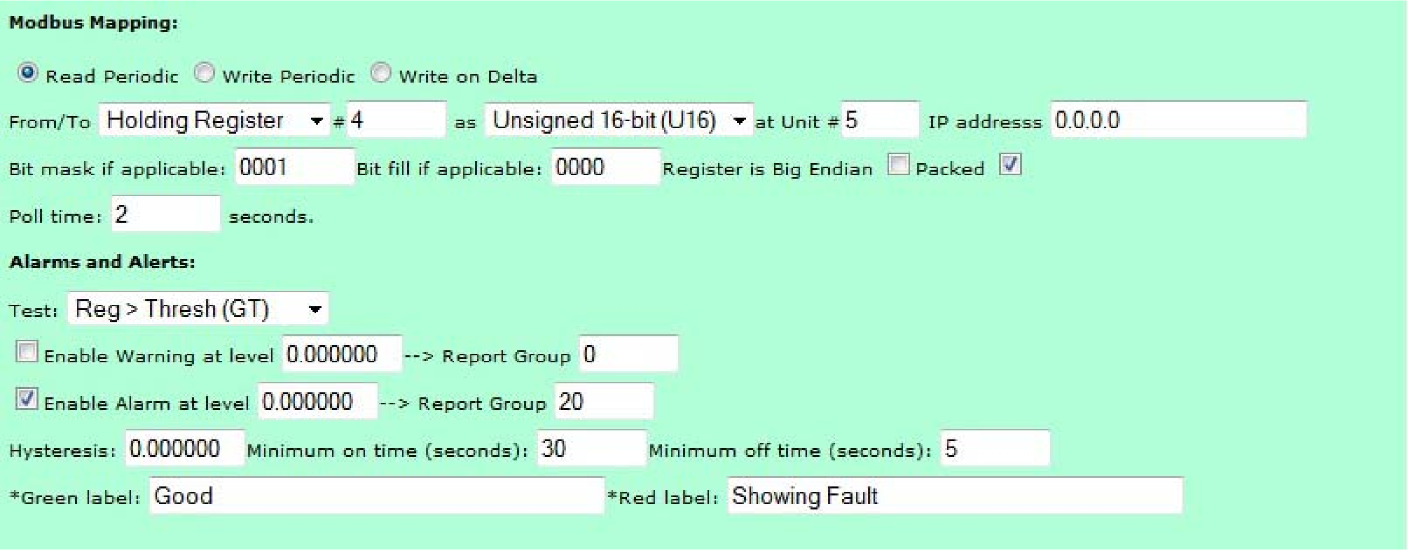
☑ Enable Warning at level 50.000000 --> Report Group 20

☑ Enable Alarm at level 100.00000 --> Report Group 20

Hysteresis: 0.000000 Minimum on time (seconds): 0 Minimum off time (seconds): 0

*Green label: *Red label:

Use of the bit mask was discussed above in the Modbus Mapping section. Setting up the special case of alarm on a specific bit within a packed register is illustrated below. You could use a test of "Reg = Thresh" and set the threshold to one, or you can use "Reg > Thresh" with a threshold of zero. Either one will report an alarm when the specified bit in the packed register is found to be set. In the example below, we want to see that bit set for at least 30 seconds before we call it an alarm. Once triggered, we want to see the bit clear for at least 5 seconds before we consider the condition resolved and reset or clear the alarm. Both the setting and clearing of an alarm are reported to the web portal, and subsequently reported to users if so configured on the web portal.



Modbus Mapping:

● Read Periodic ○ Write Periodic ○ Write on Delta

From/To Holding Register # 4 as Unsigned 16-bit (U16) at Unit # 5 IP addresss 0.0.0.0

Bit mask if applicable: 0001 Bit fill if applicable: 0000 Register is Big Endian □ Packed ☑

Poll time: 2 seconds.

Alarms and Alerts:

Test: Reg > Thresh (GT)

□ Enable Warning at level 0.000000 --> Report Group 0

☑ Enable Alarm at level 0.000000 --> Report Group 20

Hysteresis: 0.000000 Minimum on time (seconds): 30 Minimum off time (seconds): 5

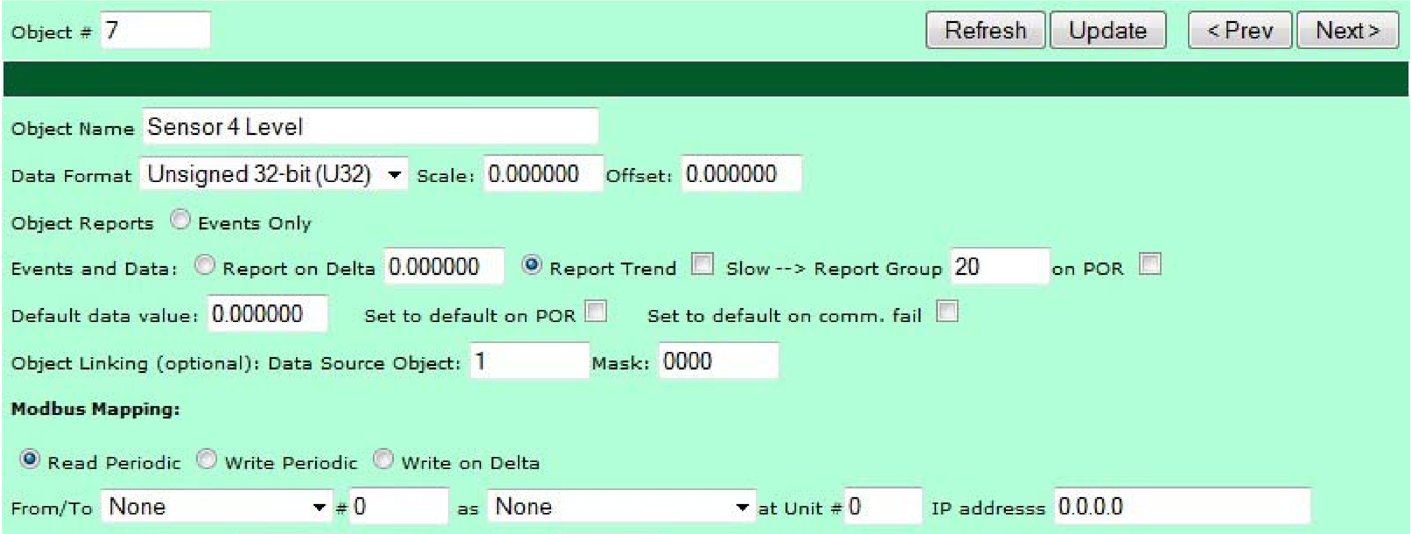
*Green label: Good *Red label: Showing Fault

The Enumberated (ENUM) test requires special explanation. Some devices provide status codes in a Modbus register, and it is desirable to report these as alarms. The Enumerated test works much like Report on Delta, except that an alarm is triggered by a change in value rather than just a routine reporting of a data value. Any non-zero value found in an Enumerated object will constitute an alarm. Return to zero is treated as clearing of the alarm. If a different value than previously reported appears in the data object, it is treated as a re-triggering of the alarm, but with a different alarm value. Additional configuration on the web portal side allows assigning a different text string to each possible status code that may be found in the monitored object. These enumeration text strings (lines) are used in generating alarm notifications (email, text, voice) to users. The enumeration strings are set up on the web portal. The only thing selected in the device locally during pre-configuration is the fact that this alarm is Enumerated.

6.4. Object Linking

Object Linking provides a means of having an object simply copy data from another object. One example of application of linking is configuring multiple alarms for the same data point. Suppose you wanted both high and low alarms for a certain Modbus register. You would set up the first object to poll the Modbus register, and also set the high alarms on that object. Then use linking to duplicate that data, and set low alarms on the linked object.

To set up object linking, simply configure a new object, and enter some other object number as the Data Source Object on the Object Linking line. The example below shows Object 7 being a duplicate copy of data obtained by Object 1.



Object # 7

Object Name Sensor 4 Level

Data Format Unsigned 32-bit (U32) Scale: 0.000000 Offset: 0.000000

Object Reports ○ Events Only

Events and Data: ○ Report on Delta 0.000000 ● Report Trend □ Slow --> Report Group 20 on POR □

Default data value: 0.000000 Set to default on POR □ Set to default on comm. fail □

Object Linking (optional): Data Source Object: 1 Mask: 0000

Modbus Mapping:

● Read Periodic ○ Write Periodic ○ Write on Delta

From/To None # 0 as None at Unit # 0 IP addresss 0.0.0.0

Another potential application for object linking is to pick certain bits out of a packed register. You can set up masking as outlined above. But you could also use linking to mask individual bits from the source that is polling the packed Modbus register. In this case, you would not select "Packed" in the Modbus mapping, nor would you enter any mask in the Modbus mapping. Instead, you would include that same type of mask on the Object Linking line. What you would potentially gain by this is to have the source register set up to trigger an alarm report when any bits in the packed register are set (test for not equal to zero), and have additional linked objects trigger additional alarms on specific individual bits. You then have the option of providing notifications to different user groups (on the web portal) based on "any alarm" versus a specific alarm.

6.5. Accessing Data Objects from Script Basic

You may read any of the local objects using the getreg function, and write them using the setreg command. Note that getreg is a function while setreg is not; it is a command and therefore requires no parenthesis. Consider the following example:

```
MyData = getreg (22)
MyData = MyData * 2.5
setreg 24, MyData
```

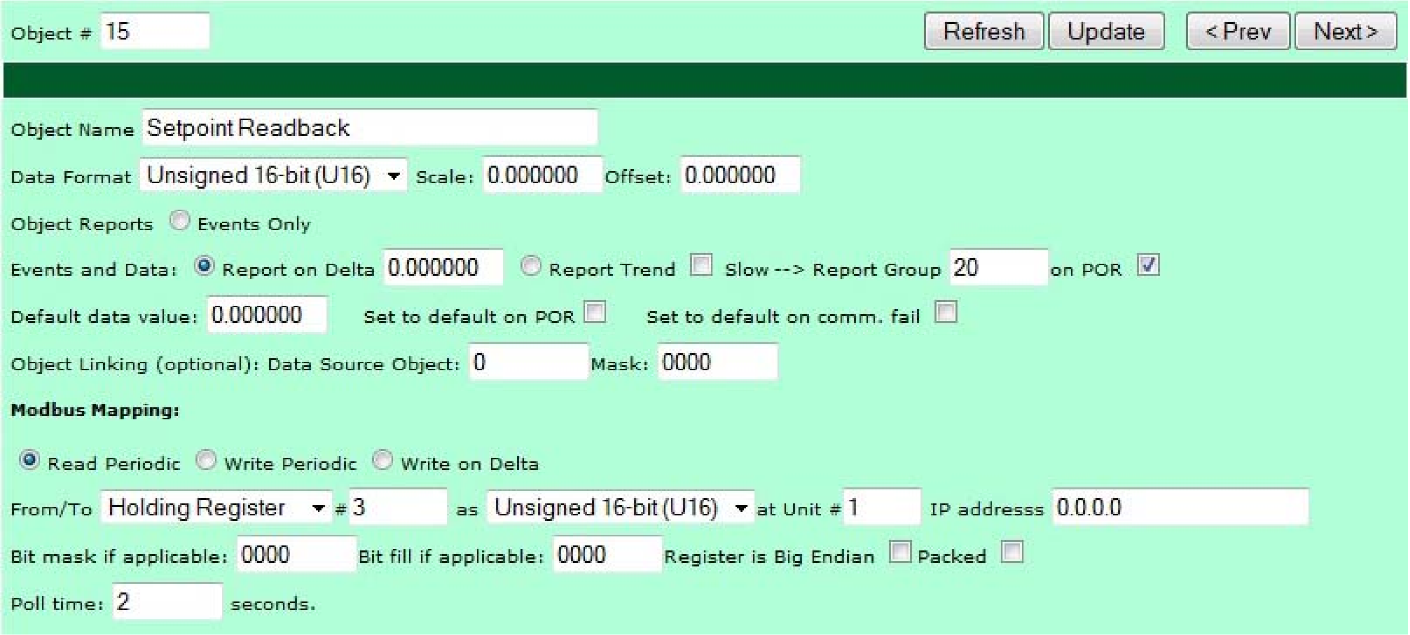
The above example will place the contents of register #22 in the variable MyData, then multiply it by 2.5, then place that value in register #24. Variables may be used in place of the constants used in the above example.

The register numbers referenced in getreg and setreg are the same numbers outlined in Section 5.4, Modbus Slave Register Map. Internally, the Script Basic program treats all register data as floating point regardless of what format the object was configured as. Any data format conversion needed will be done automatically on the fly. However, to avoid losing range for floating point data, register numbers referring to objects as floating point should be used for floating point objects.

6.6. Reading and Writing Same Modbus Register

There is sometimes a need to read always and write infrequently to the same Modbus register. For example, this is useful when an external device contains a setpoint that can be set locally via an operator panel, and yet you want to not only display that setpoint remotely but have the option of changing it remotely. You cannot simply set up a periodic write, or even a "write on delta" because this would result in always overriding whatever had been entered via the local operator panel. Solving this problem is done with the use of two carefully configure objects that work as a team.

The first object should be set up to periodically read the setpoint. This is set up in the same manner as reading any other Modbus register, as discussed under Modbus Mapping above.



Object # 15

Object Name Setpoint Readback

Data Format Unsigned 16-bit (U16) Scale: 0.000000 Offset: 0.000000

Object Reports ○ Events Only

Events and Data: ● Report on Delta 0.000000 ○ Report Trend □ Slow --> Report Group 20 on POR ☑

Default data value: 0.000000 Set to default on POR □ Set to default on comm. fail □

Object Linking (optional): Data Source Object: 0 Mask: 0000

Modbus Mapping:

● Read Periodic ○ Write Periodic ○ Write on Delta

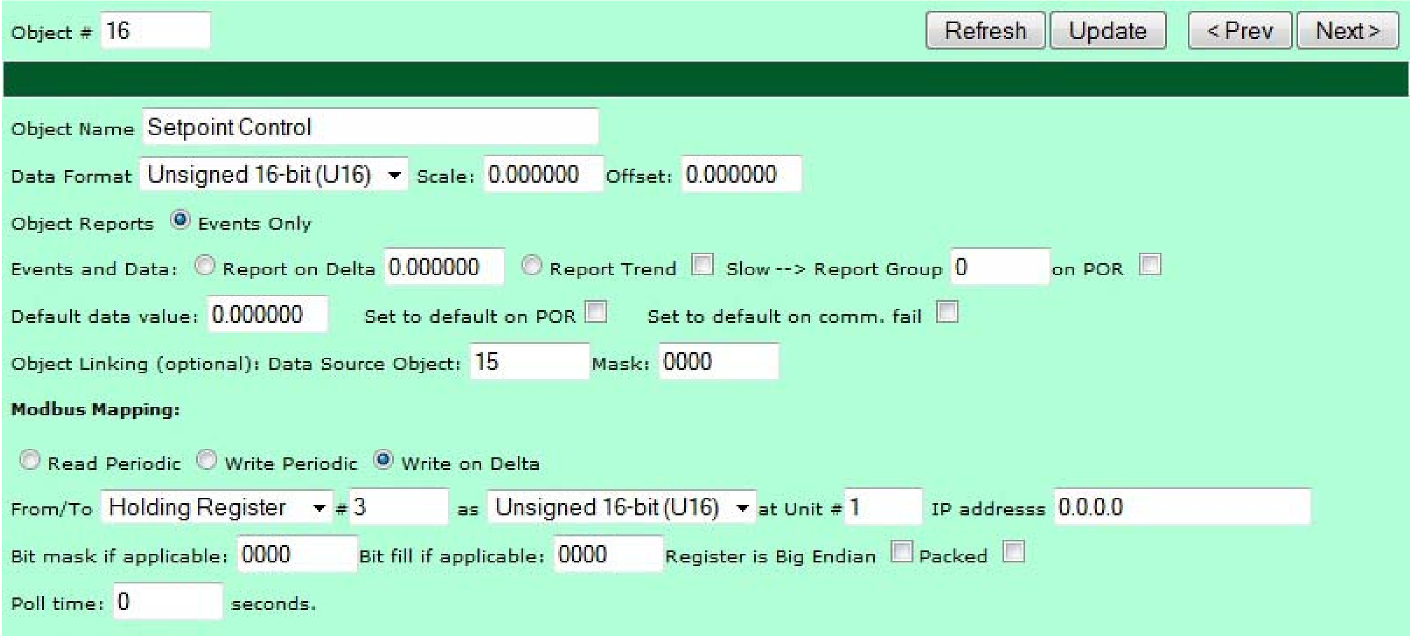
From/To Holding Register # 3 as Unsigned 16-bit (U16) at Unit # 1 IP addresss 0.0.0.0

Bit mask if applicable: 0000 Bit fill if applicable: 0000 Register is Big Endian □ Packed □

Poll time: 2 seconds.

The second object is set up for Write on Delta, but with a special trick. The object doing the reading should be referenced as the data source on the Object Linking line for the object doing the writing.

The write object should not be configured to report to the web portal since it would only report changes you sent, and you already know what those are. The read object should be the one set to Report on Delta. This way, you will see whatever the register value is at any time, and changes made via the local operator panel will be reported in addition to any changes you make remotely.



Object # 16

Object Name Setpoint Control

Data Format Unsigned 16-bit (U16) Scale: 0.000000 Offset: 0.000000

Object Reports ● Events Only

Events and Data: ○ Report on Delta 0.000000 ○ Report Trend □ Slow --> Report Group 0 on POR □

Default data value: 0.000000 Set to default on POR □ Set to default on comm. fail □

Object Linking (optional): Data Source Object: 15 Mask: 0000

Modbus Mapping:

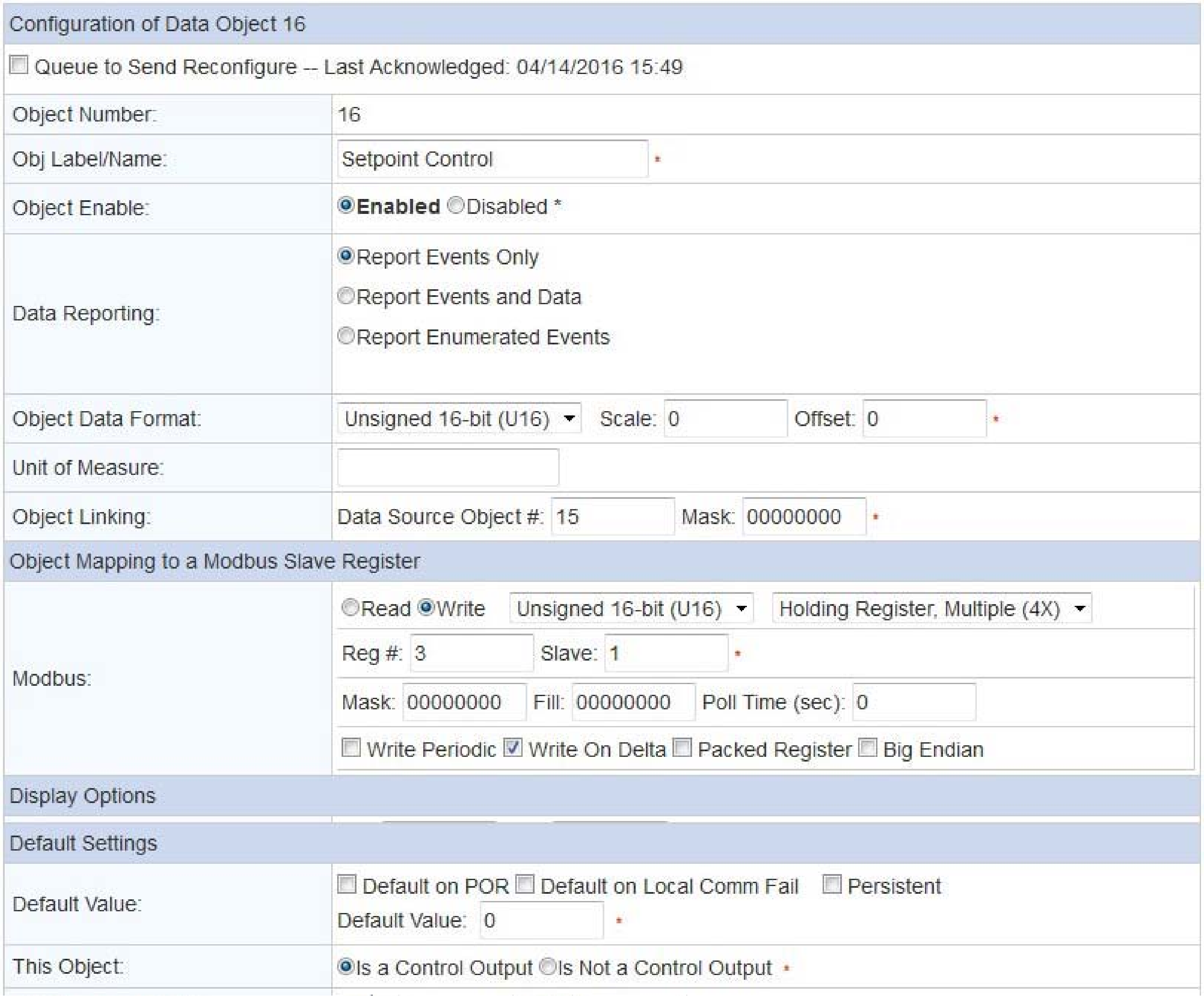
○ Read Periodic ○ Write Periodic ● Write on Delta

From/To Holding Register # 3 as Unsigned 16-bit (U16) at Unit # 1 IP addresss 0.0.0.0

Bit mask if applicable: 0000 Bit fill if applicable: 0000 Register is Big Endian □ Packed □

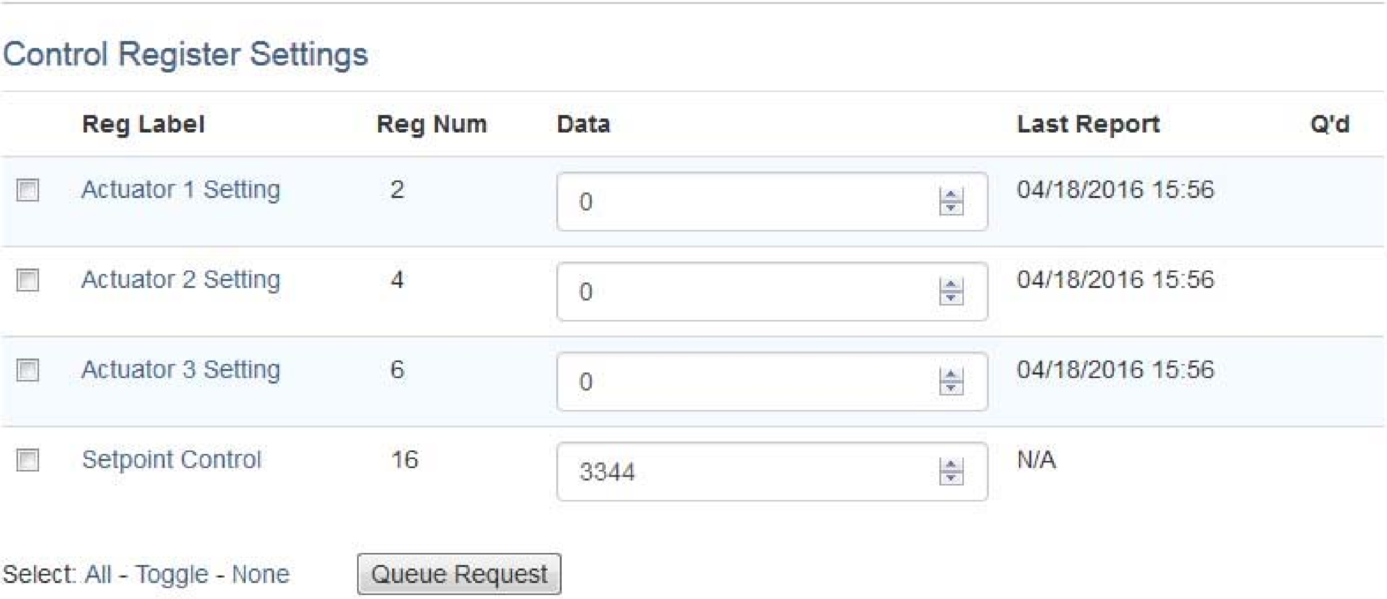
Poll time: 0 seconds.

The view of the "trick object" doing the writing as it will appear on the web portal is illustrated below. Everything about the configuration is the same, but there is one additional setting on the web portal that needs to be made. Select "Is a Control Output" at the bottom of the page (and click Update to save that setting).



Configuration of Data Object 16	
☐ Queue to Send Reconfigure -- Last Acknowledged: 04/14/2016 15:49	
Object Number:	16
Obj Label/Name:	Setpoint Control *
Object Enable:	● **Enabled** ○ Disabled *
Data Reporting:	● Report Events Only ○ Report Events and Data ○ Report Enumerated Events
Object Data Format:	Unsigned 16-bit (U16) ▾ Scale: 0 Offset: 0 *
Unit of Measure:	
Object Linking:	Data Source Object #: 15 Mask: 00000000 *
Object Mapping to a Modbus Slave Register	
Modbus:	○ Read ● Write Unsigned 16-bit (U16) ▾ Holding Register, Multiple (4X) ▾ Reg #: 3 Slave: 1 * Mask: 00000000 Fill: 00000000 Poll Time (sec): 0 ☐ Write Periodic ☑ Write On Delta ☐ Packed Register ☐ Big Endian
Display Options	
Default Settings	
Default Value:	☐ Default on POR ☐ Default on Local Comm Fail ☐ Persistent Default Value: 0 *
This Object:	● Is a Control Output ○ Is Not a Control Output *

Once you have selected this object as a Control Output, it will appear in the Control Register Settings section of the Diagnostics page. This is where you go to remotely change the setpoint in the above example. To make the change, enter a new value in the Data window, check the box at the start of the respective row, and click Queue Request. Your change will be sent to the BB2-6010-Web device the next time it pings the server, which normally happens once every 60 seconds or sooner.



Control Register Settings

	Reg Label	Reg Num	Data	Last Report	Q'd
□	Actuator 1 Setting	2	0	04/18/2016 15:56	
□	Actuator 2 Setting	4	0	04/18/2016 15:56	
□	Actuator 3 Setting	6	0	04/18/2016 15:56	
□	Setpoint Control	16	3344	N/A	

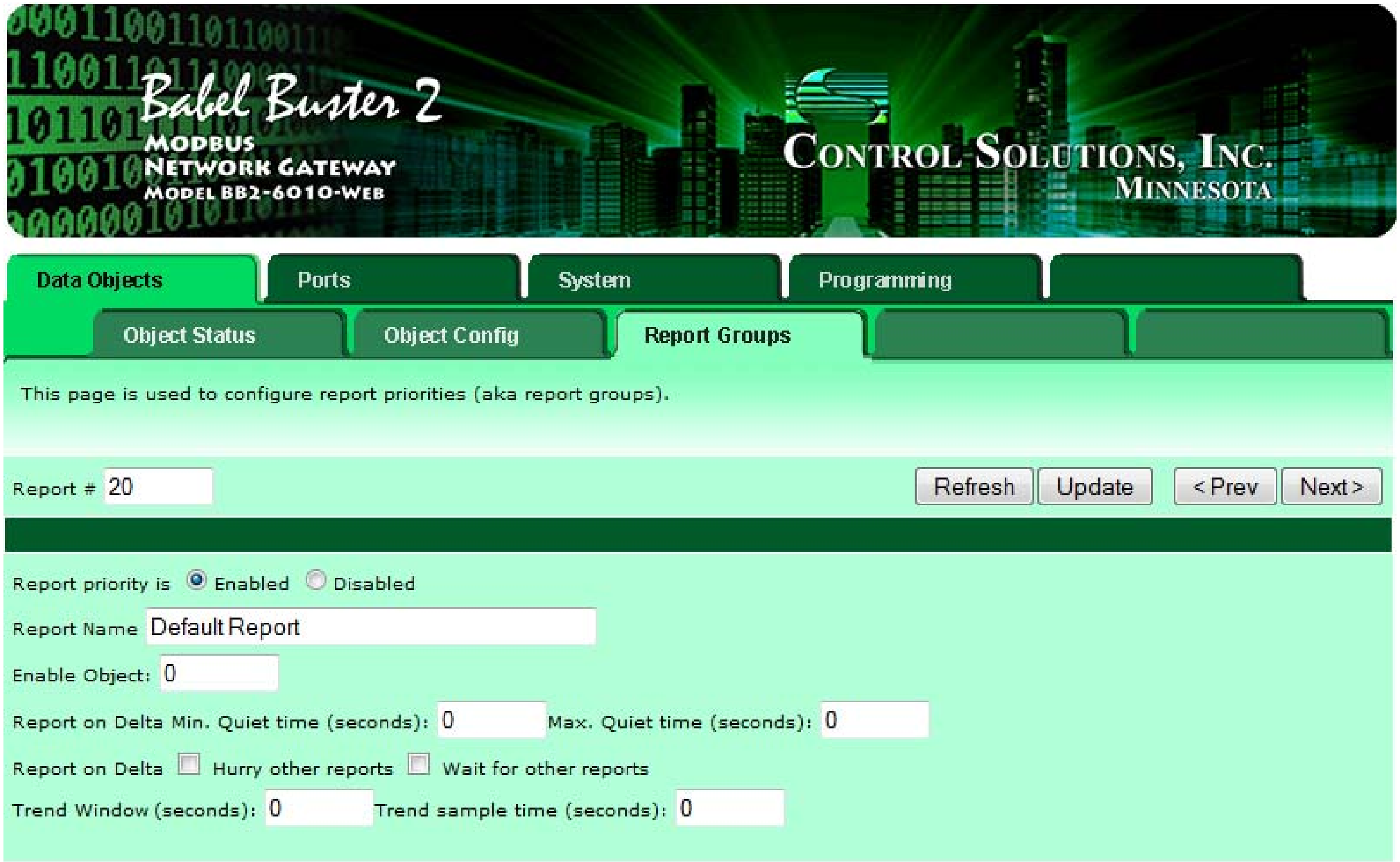
7. Configuring Report Groups

7.1. Minimum Requirements

Report group numbers are more than just a number. They are a priority level. Reporting assigned to group or priority #1 will always get reported first, followed by group/priority 2, and so on. To leave room for adding higher priority levels later, it is suggested that you start somewhere in the middle of the list for your first group. We tend to default to starting with report group 20.

Having said that, it is worth noting that report priorities are not that important for a wired device like the BB2-6010-Web. The priority scheme was originally created for wireless devices having both cellular and satellite modems. These mobile devices would automatically roll over to satellite when cellular coverage was lost. However, since satellite data is much more expensive than cellular, it was desirable to have the means to only report the most important events via satellite, and report the rest later when cellular coverage was reestablished. The report priorities where then used to structure what got reported when, and under what circumstances.

The minimum requirement for creating a report group that will enable sending data from the device to the web portal is to simply give it a name and enable it. Everything else is technically optional.



This page is used to configure report priorities (aka report groups).

Report # 20

Report priority is ● Enabled ○ Disabled

Report Name Default Report

Enable Object: 0

Report on Delta Min. Quiet time (seconds): 0 Max. Quiet time (seconds): 0

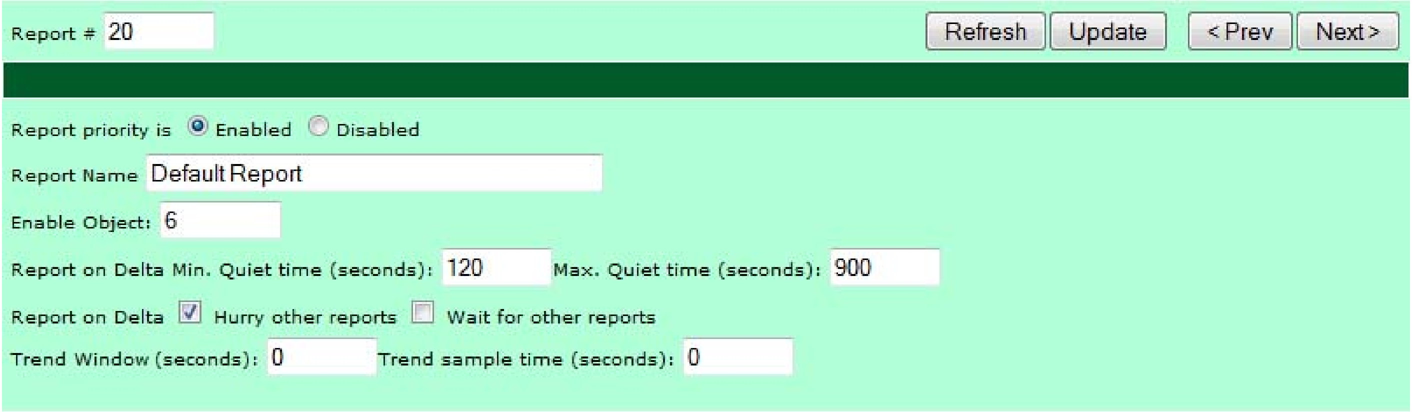
Report on Delta □ Hurry other reports □ Wait for other reports

Trend Window (seconds): 0 Trend sample time (seconds): 0

7.2. Conditional Reporting

You have the option of establishing conditional reporting, meaning this report group will only be active under certain circumstances. To selectively enable a report group, enter a non-zero object number as the Enable Object. This report now becomes active only when this object contains a non-zero number.

One example of use of this feature is reporting power data from a backup generator. Since reporting zero volts while the generator is not running is not useful, the Enable Object is set to be an object that monitors a "generator running" status bit. Now power data is only reported when power is being generated.



Report # 20

Report priority is ● Enabled ○ Disabled

Report Name Default Report

Enable Object: 6

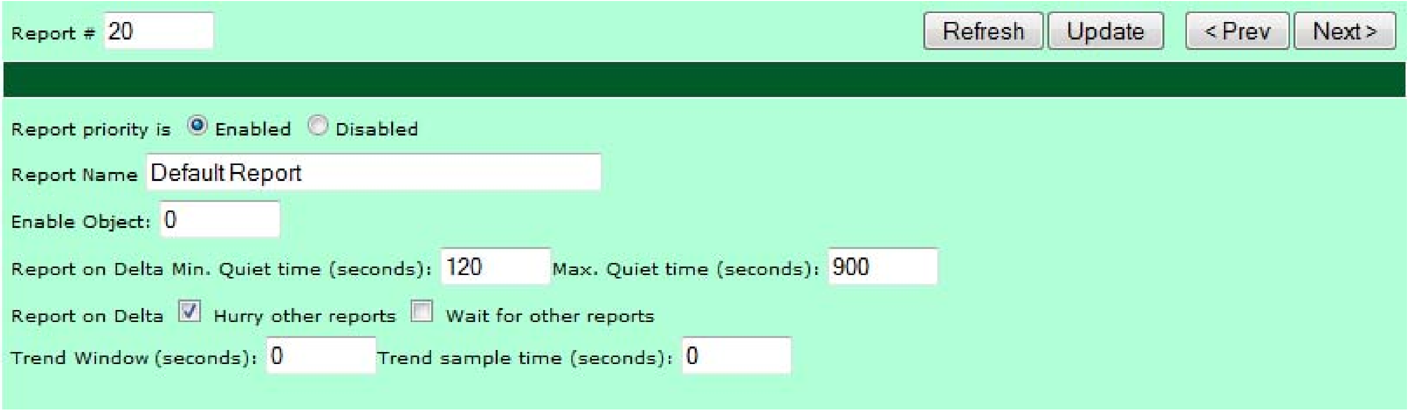
Report on Delta Min. Quiet time (seconds): 120 Max. Quiet time (seconds): 900

Report on Delta ☑ Hurry other reports □ Wait for other reports

Trend Window (seconds): 0 Trend sample time (seconds): 0

7.3. Report Rate Limiting

You have the ability to limit the rate at which data objects Report on Delta. (Alarms and alerts always report immediately upon occurrence, and are never limited in any way.) Limiting the reporting rate is of substantial importance on wireless devices where cost is proportional to data usage. Rate limiting is of less concern with wired devices (unless the "wired device" is wired to a cellular modem!).



Report # 20

Report priority is ● Enabled ○ Disabled

Report Name Default Report

Enable Object: 0

Report on Delta Min. Quiet time (seconds): 120 Max. Quiet time (seconds): 900

Report on Delta ☑ Hurry other reports ☐ Wait for other reports

Trend Window (seconds): 0 Trend sample time (seconds): 0

The minimum quiet time means this amount of time must elapse before another Report on Delta will be sent to the web portal, regardless of how much or how frequently the data changed in the mean time.

The maximum quiet time is useful for even wired devices, and forces a report at least this often even if the data value has not changed at all.

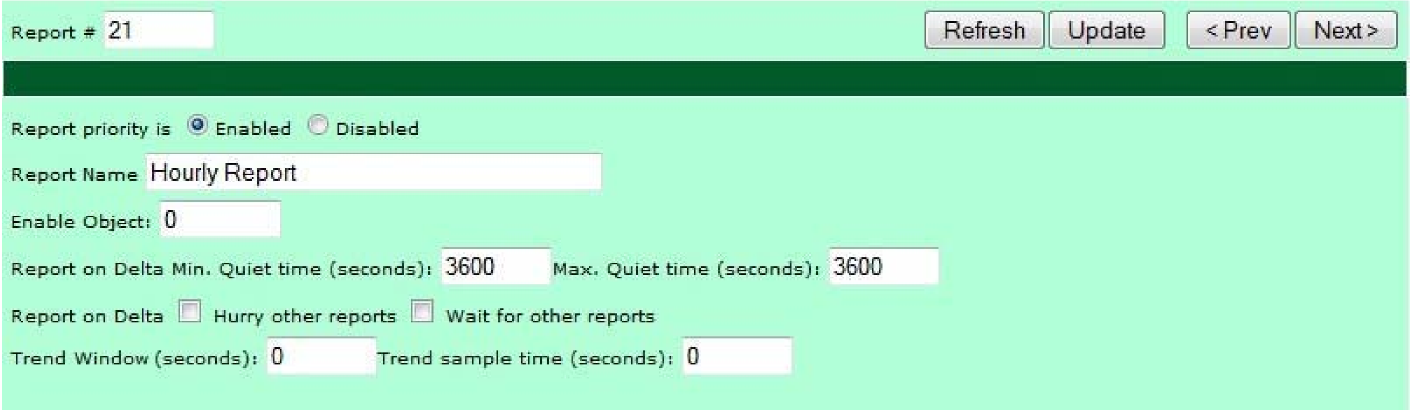
The "hurry" or "wait" options decide if and how reports are grouped into a single report to the web portal. If the "hurry" option is set, then all Report on Delta points in this report group will be reported, even if no change has occurred, if their minimum quiet time is at least half expired.

The "wait" option means Report on Delta will be held up until all quiet times in the group have expired (a feature not often used).

7.4. Forcing Periodic Reporting, Including On the Hour or Daily

You will automatically force strictly periodic reporting of all Report on Delta objects assigned to this group if you set the minimum and maximum quiet times to exactly the same time.

Furthermore, you will force the periodic reporting to be synchronized with real time (SNTP must have been set up) and report every hour on the hour if you set minimum and maximum quiet times to exactly 3600 seconds (one hour).



Report # 21

Report priority is ● Enabled ○ Disabled

Report Name Hourly Report

Enable Object: 0

Report on Delta Min. Quiet time (seconds): 3600 Max. Quiet time (seconds): 3600

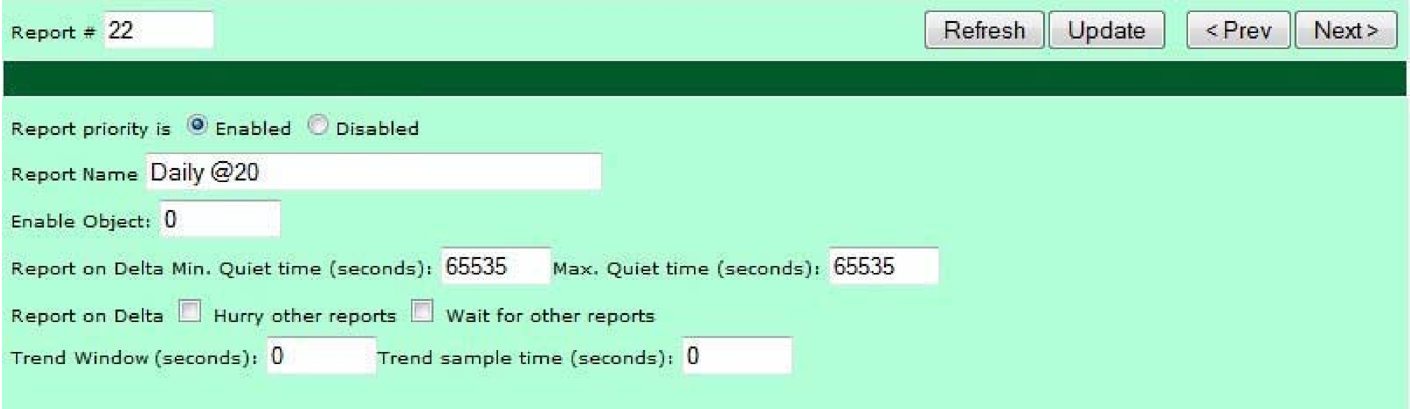
Report on Delta □ Hurry other reports □ Wait for other reports

Trend Window (seconds): 0 Trend sample time (seconds): 0

The other periodic reporting "trick" available is daily reporting. To create a daily report, set minimum and maximum quiet times to 65535 seconds. This special time value triggers daily report processing. The report name requires special consideration for daily reporting. The name can be anything, but must include "@N" where N is the hour at which the daily report should be made. The hour should be 0 to 23, and will be synchronized to the real time as established by SNTP.

The daily reporting applies to Report on Delta. This is typically used where the register being monitored by this data object is some type of totalizing meter, such as a total flow meter.

The example below says that any Report on Delta points assigned to report group 22 will be reported daily at 8PM local time (local time as determined by SNTP settings).



Report # 22

Report priority is ● Enabled ○ Disabled

Report Name Daily @20

Enable Object: 0

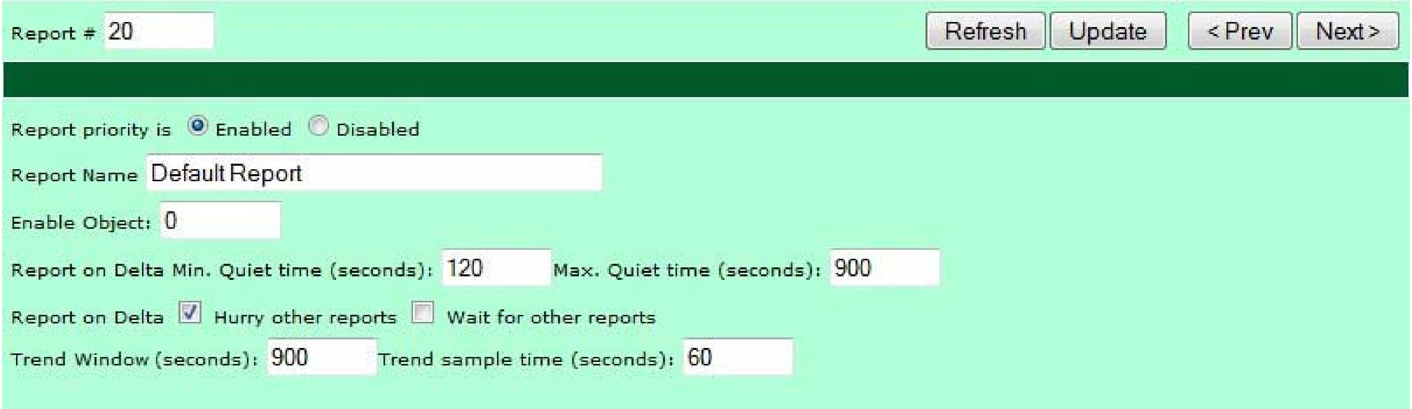
Report on Delta Min. Quiet time (seconds): 65535 Max. Quiet time (seconds): 65535

Report on Delta □ Hurry other reports □ Wait for other reports

Trend Window (seconds): 0 Trend sample time (seconds): 0

7.5. Trend Reporting

Trend reporting accumulates a minimum, maximum, and average value over a window of time defined as the Trend Window. During this window, the data object's value will be sampled periodically at the trend sample time rate. In other words, if the trend window is 900 and sample time is 60, then the average will be the sum of 15 values, divided by 15. The minimum will be the smallest of those 15 values, and maximum will be the largest of those 15 values. Upon expiration of the trend window time, the data values will be sent to the web portal for logging.



Report # 20

Report priority is ● Enabled ○ Disabled

Report Name Default Report

Enable Object: 0

Report on Delta Min. Quiet time (seconds): 120 Max. Quiet time (seconds): 900

Report on Delta ☑ Hurry other reports □ Wait for other reports

Trend Window (seconds): 900 Trend sample time (seconds): 60

Babel Buster 2
Modbus
Network Gateway
Model BB2-6010-Web

Control Solutions, Inc.
Minnesota

8. Connecting to the Web Portal

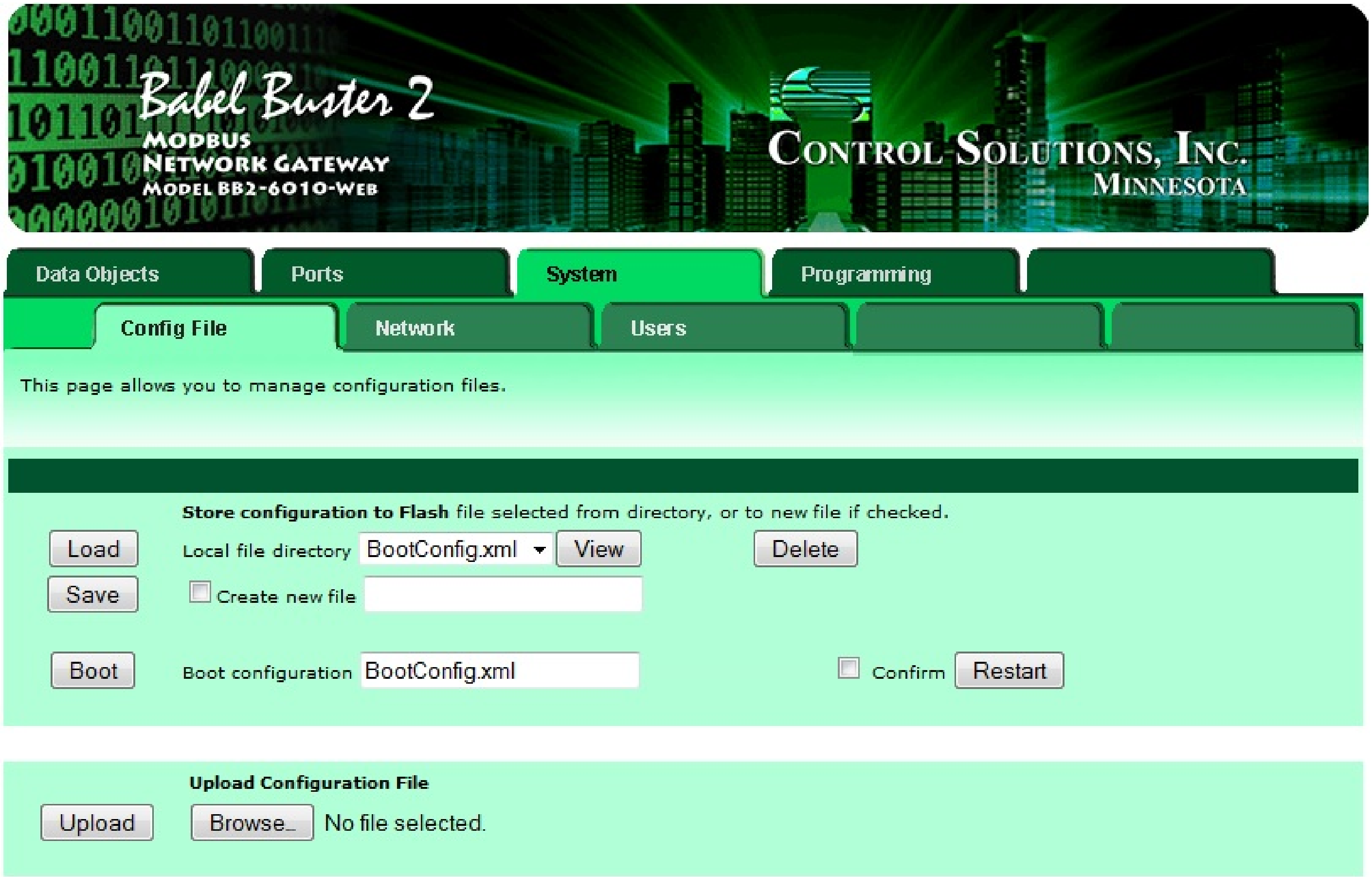
8.1. Uploading Configuration to Web Portal

Once you have entered all of your configuration settings into the BB2-6010-Web, you need to go to the Config File page and click Save to save your changes. The file named in the "Boot configuration" window is the active file that will be reloaded when the BB2-6010-Web reboots. Select this file name when saving your configuration (or create a new file and change the Boot configuration file name).

The file named as the Boot configuration file is also the file where changes made remotely via the web portal will be saved.

Entering configuration locally via the web pages in the BB2-6010-Web should be thought of as "pre-configuration". You will upload this pre-configuration to the web portal, but once the device goes online, all configuration from that point on should only be done via the web portal.

It is not necessary to do any pre-configuration locally. You do have the option of going straight to network setup to place the BB2-6010-Web online immediately, and then do all configuration including initial configuration via the web portal. This is especially useful if the new BB2-6010-Web going online should function exactly like another BB2-6010-Web already online. You can simply copy that other device's configuration to the new device in one simple step on the web portal.



This page allows you to manage configuration files.

Store configuration to Flash file selected from directory, or to new file if checked.

Local file directory BootConfig.xml

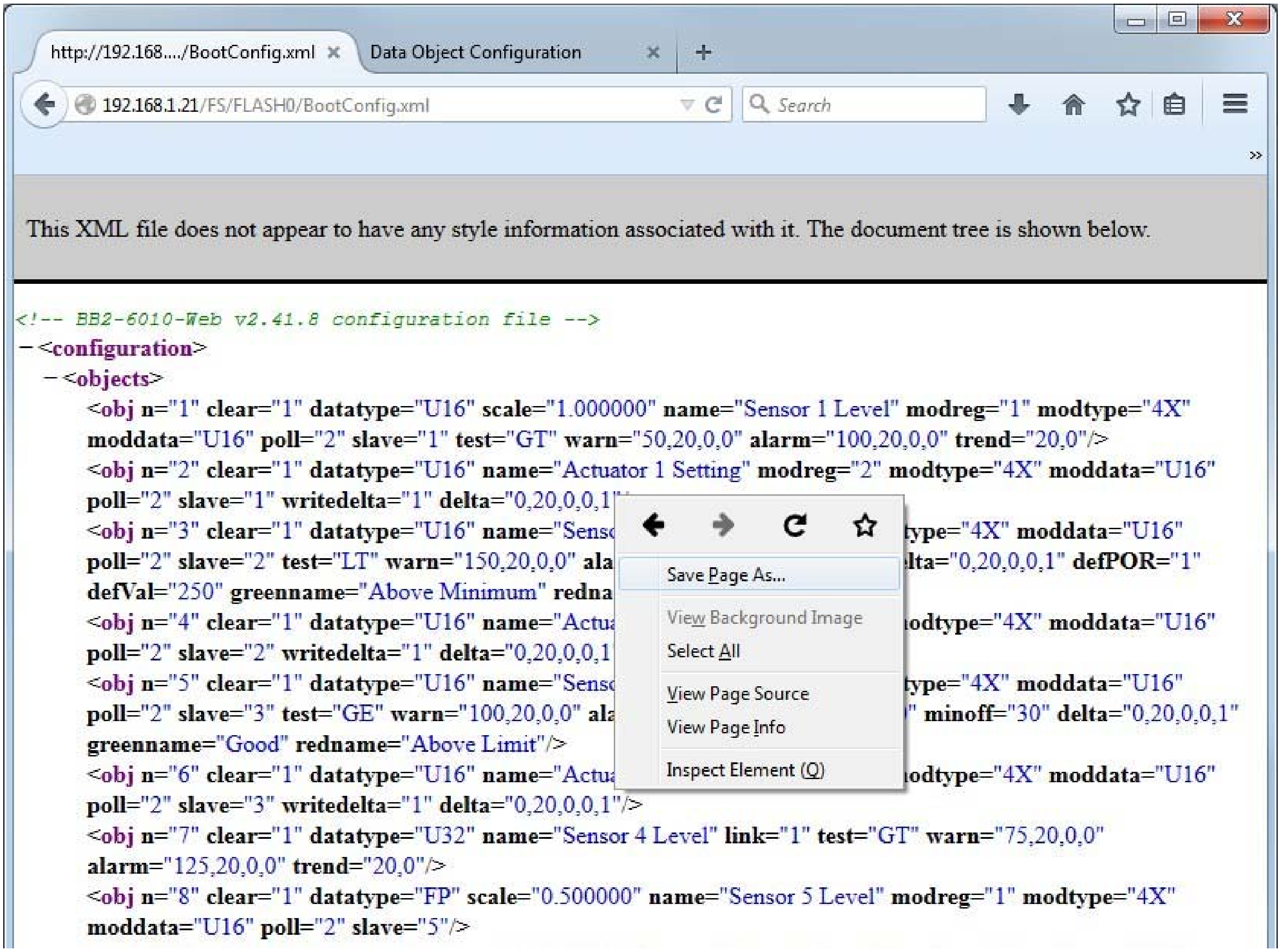
□ Create new file

Boot configuration BootConfig.xml

□ Confirm

No file selected.

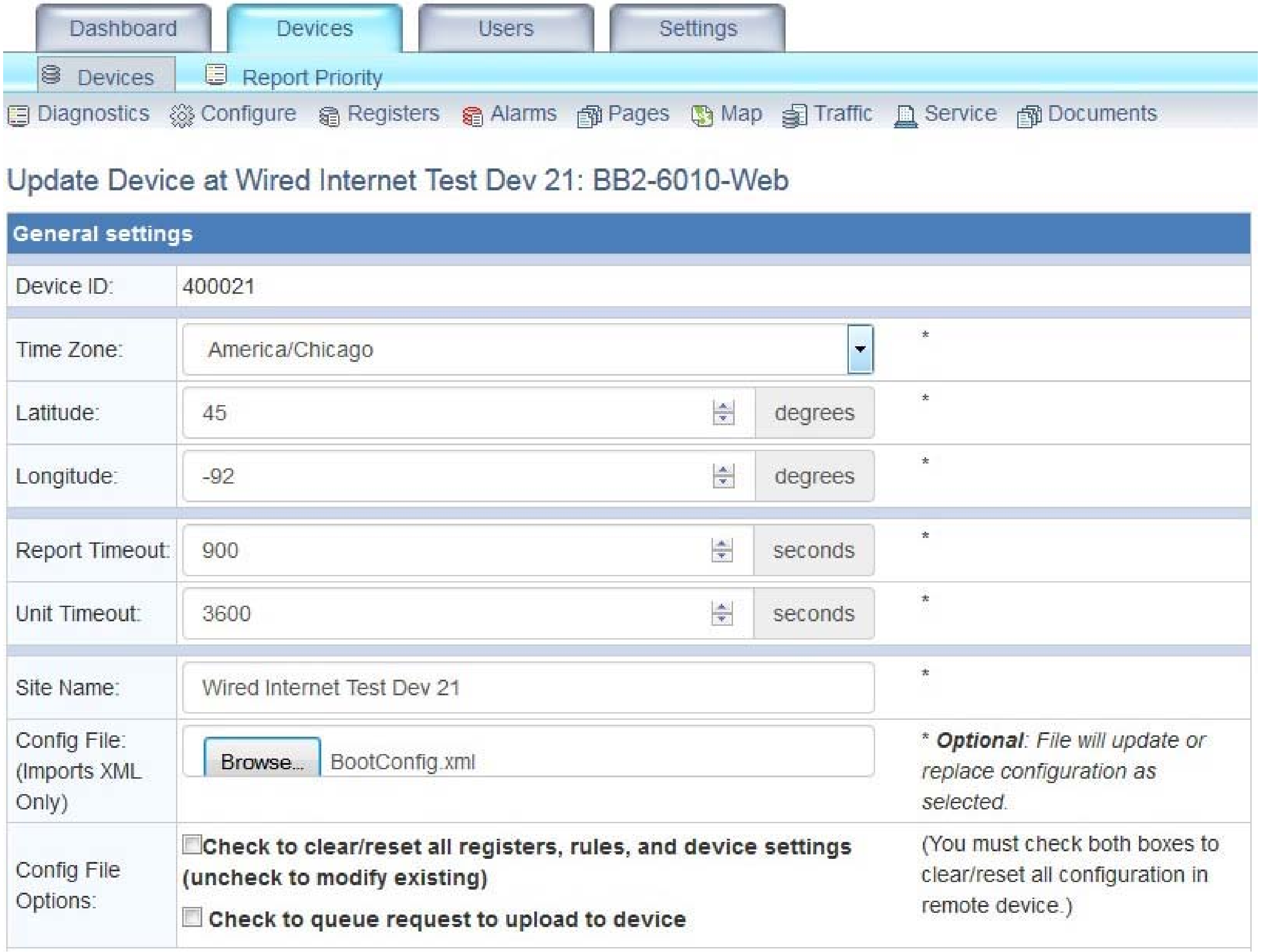
Assuming you did pre-configure the BB2-6010-Web locally and now want to export its configuration to upload to the web portal, start by going to the Config File page and click the View button with the Boot configuration file selected. Your browser should now display XML that appears something like the illustration below. Now right click and select "Save Page As..." (or find your browser's version of the same function). The browser should default to saving the file named with a *.xml suffix. Once saved, double click on the new file on your PC - if correctly saved, it should automatically open and display the same XML without errors. Do not copy/paste text from the browser window - this will not create a valid useable XML file.



This XML file does not appear to have any style information associated with it. The document tree is shown below.

```
<!-- BB2-6010-Web v2.41.8 configuration file -->
- <configuration>
  - <objects>
      <obj n="1" clear="1" datatype="U16" scale="1.000000" name="Sensor 1 Level" modreg="1" modtype="4X"
      moddata="U16" poll="2" slave="1" test="GT" warn="50,20,0,0" alarm="100,20,0,0" trend="20,0"/>
      <obj n="2" clear="1" datatype="U16" name="Actuator 1 Setting" modreg="2" modtype="4X" moddata="U16"
      poll="2" slave="1" writedelta="1" delta="0,20,0,0,1"
      <obj n="3" clear="1" datatype="U16" name="Senso                type="4X" moddata="U16"
      poll="2" slave="2" test="LT" warn="150,20,0,0" ala              lta="0,20,0,0,1" defPOR="1"
      defVal="250" greenname="Above Minimum" redna
      <obj n="4" clear="1" datatype="U16" name="Actu              odtype="4X" moddata="U16"
      poll="2" slave="2" writedelta="1" delta="0,20,0,0,1
      <obj n="5" clear="1" datatype="U16" name="Senso                type="4X" moddata="U16"
      poll="2" slave="3" test="GE" warn="100,20,0,0" ala              " minoff="30" delta="0,20,0,0,1"
      greenname="Good" redname="Above Limit"/>
      <obj n="6" clear="1" datatype="U16" name="Actu              odtype="4X" moddata="U16"
      poll="2" slave="3" writedelta="1" delta="0,20,0,0,1"/>
      <obj n="7" clear="1" datatype="U32" name="Sensor 4 Level" link="1" test="GT" warn="75,20,0,0"
      alarm="125,20,0,0" trend="20,0"/>
      <obj n="8" clear="1" datatype="FP" scale="0.500000" name="Sensor 5 Level" modreg="1" modtype="4X"
      moddata="U16" poll="2" slave="5"/>
```

To upload the newly saved XML file to the web portal, go to the Configure page (on the web portal) for your device, click on the Browse button on the Config File line, and find your file. Select the file, then click the Save button at the bottom of the web page (scroll down as necessary). You must also select user groups to be notified before clicking the Save button. These appear in more than one place on the Configure page (not illustrated here for brevity).



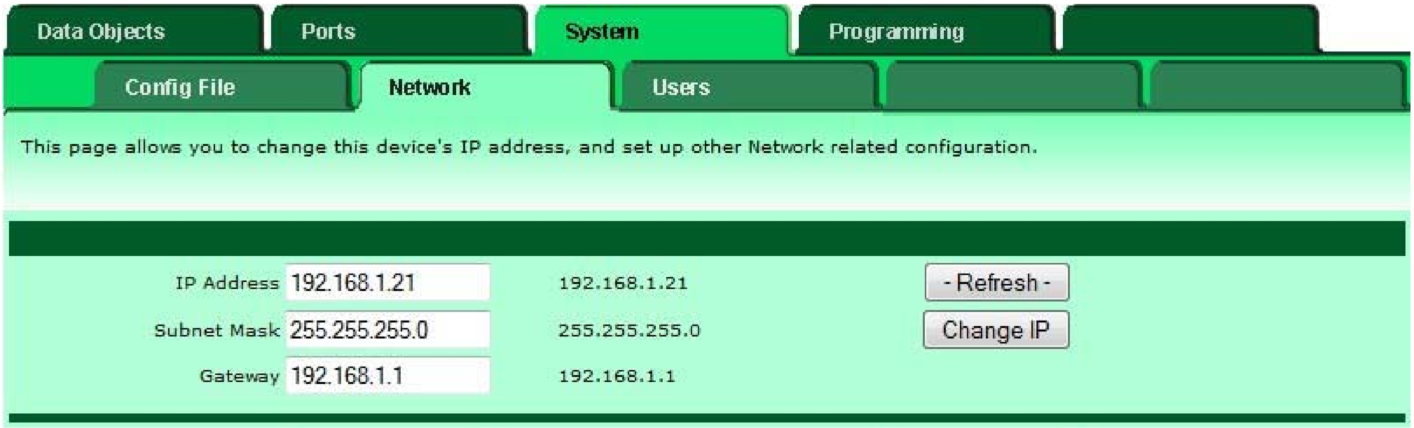
Update Device at Wired Internet Test Dev 21: BB2-6010-Web

General settings		
Device ID:	400021	
Time Zone:	America/Chicago	*
Latitude:	45 degrees	*
Longitude:	-92 degrees	*
Report Timeout:	900 seconds	*
Unit Timeout:	3600 seconds	*
Site Name:	Wired Internet Test Dev 21	*
Config File: (Imports XML Only)	BootConfig.xml	* ***Optional***: *File will update or replace configuration as selected.*
Config File Options:	□ **Check to clear/reset all registers, rules, and device settings (uncheck to modify existing)** □ **Check to queue request to upload to device**	(You must check both boxes to clear/reset all configuration in remote device.)

8.2. Network Settings in BB2-6010-Web

Placing the BB2-6010-Web online requires a few settings to be made on the Setup :: Network page in the BB2-6010-Web. The first section on the page is for setting the IP address of the BB2-6010-Web. You must be logged in as the root user in order to change the IP address.

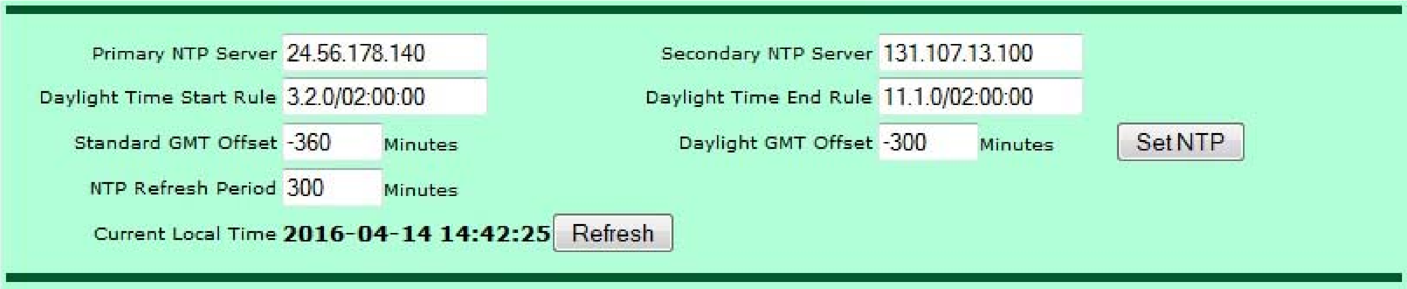
If your device will have a static IP address, enter that IP address, the applicable subnet mask, and the applicable network gateway. If your BB2-6010-Web should use DHCP instead, then enter 255.255.255.255 for IP address. Be sure a DHCP server is available on the network before doing this because the BB2-6010-Web will not boot up if configured for DHCP but no DHCP server is found.



This page allows you to change this device's IP address, and set up other Network related configuration.

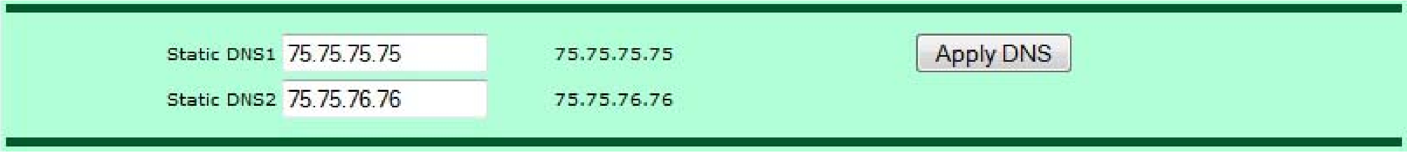
IP Address 192.168.1.21 192.168.1.21
Subnet Mask 255.255.255.0 255.255.255.0
Gateway 192.168.1.1 192.168.1.1

The BB2-6010-Web maintains an internal real time clock, but depends on SNTP to do this. Refer to the directions in the Quick Help section at the bottom of the Network page for setting up SNTP. To find a list of available time servers, just do a web search for "NIST time servers".



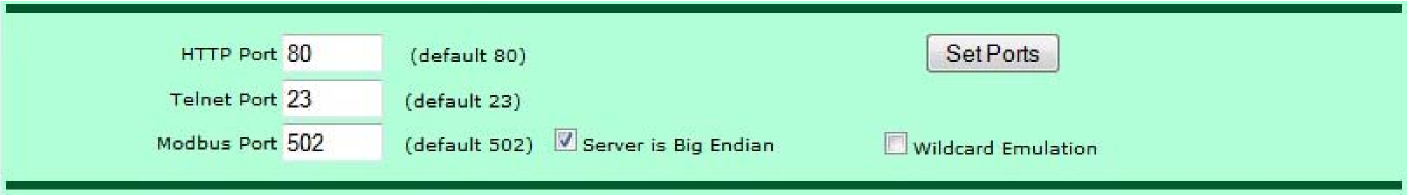
Primary NTP Server 24.56.178.140
Secondary NTP Server 131.107.13.100
Daylight Time Start Rule 3.2.0/02:00:00
Daylight Time End Rule 11.1.0/02:00:00
Standard GMT Offset -360 Minutes
Daylight GMT Offset -300 Minutes
NTP Refresh Period 300 Minutes
Current Local Time **2016-04-14 14:42:25**

Your BB2-6010-Web will not find the web portal without valid DNS servers. If you are using a static IP address, then you must also provide static IP addresses for an accessible DNS server. If you are using DHCP, then DNS servers should be provided by the DHCP server.



Static DNS1 75.75.75.75 75.75.75.75
Static DNS2 75.75.76.76 75.75.76.76

The ports do not affect BB2-6010-Web connecting to the web portal. The HTTP port refers to the port used to access the web pages locally in the BB2-6010-Web. The only use Telnet has in the BB2-6010-Web is recovery of a corrupted MAC address. You can disable Telnet by entering zero here (which it will default to). Modbus TCP generally uses port 502. If you will be using Modbus TCP, enter the port number, otherwise enter zero to disable Modbus TCP (which it will defaul to). Setting for Big Endian applies if the BB2-6010-Web device will be treated as a Modbus TCP slave by some external Modbus TCP master. Refer to section 12 for discussion of Wildcard Emulation. You normally do not want to check this box.



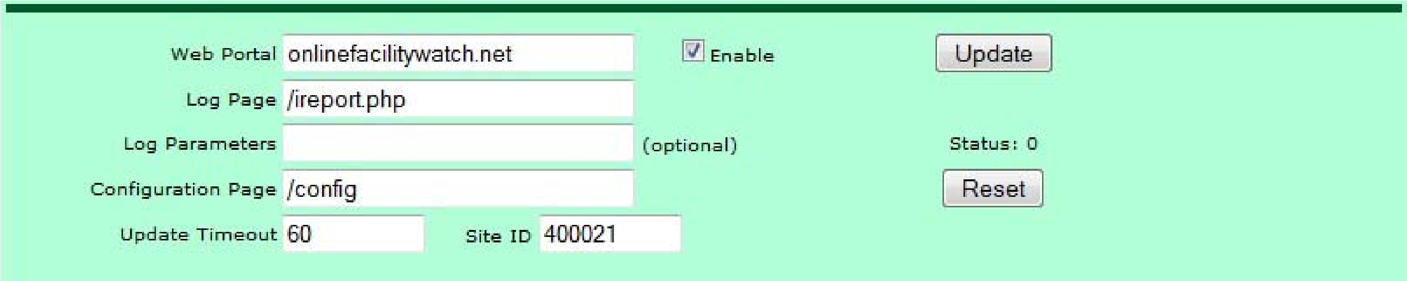
HTTP Port 80 (default 80)

Telnet Port 23 (default 23)

Modbus Port 502 (default 502) ☑ Server is Big Endian ☐ Wildcard Emulation

The last section on the Network page is where you tell the BB2-6010-Web what web portal it should report to. Enter the web portal domain that was provided to you when you signed up for an account on a portal. The log page will be /ireport.php unless instructed otherwise. You do not need to enter any log parameters - these are for special purpose portals only. Configuration page should be /config unless instructed otherwise. Update timeout is relatively unimportant since this only dictates when the BB2-6010-Web will report for the first time - after that, report timing is directed from the web portal.

Site ID is very important. The two variables in your instance of this part of the Network setup page are the web portal domain, and site ID. Multiple devices on the same portal will have different Site IDs. The Site ID (or Device ID as sometimes referenced on the web portal) is this device's identity. You will not remotely connect with your BB2-6010-Web, or receive any data or alarms from it, if this Site ID does not match the Site or Device ID on the web portal. This number will be assigned for you and provided along with the domain name when you sign up for a web portal account.



Web Portal onlinefacilitywatch.net ☑ Enable

Log Page /ireport.php

Log Parameters (optional)

Status: 0

Configuration Page /config

Update Timeout 60 Site ID 400021

Finally, when everything else is set correctly, check the Enable box and click Update. The update will take typically 30 seconds as this information is being programmed into Flash memory during this delay. Once the page refreshes, indicating completion, you are now "Live". In about 60 seconds (or whatever timeout was entered), you should see this device's first report to the web portal. If you do not see any traffic within a few minutes, which you can verify by checking the Traffic page for your device, you should take the device offline by unchecking Enable and clicking Update again. Determine why you did not connect before proceeding, and correct any network configuration problems. Do not leave a device with a faulty connection in "Live" mode - you may not know what trouble you are causing on your local network.

9. Online Operation of BB2-6010-Web

9.1. Making Configuration Changes

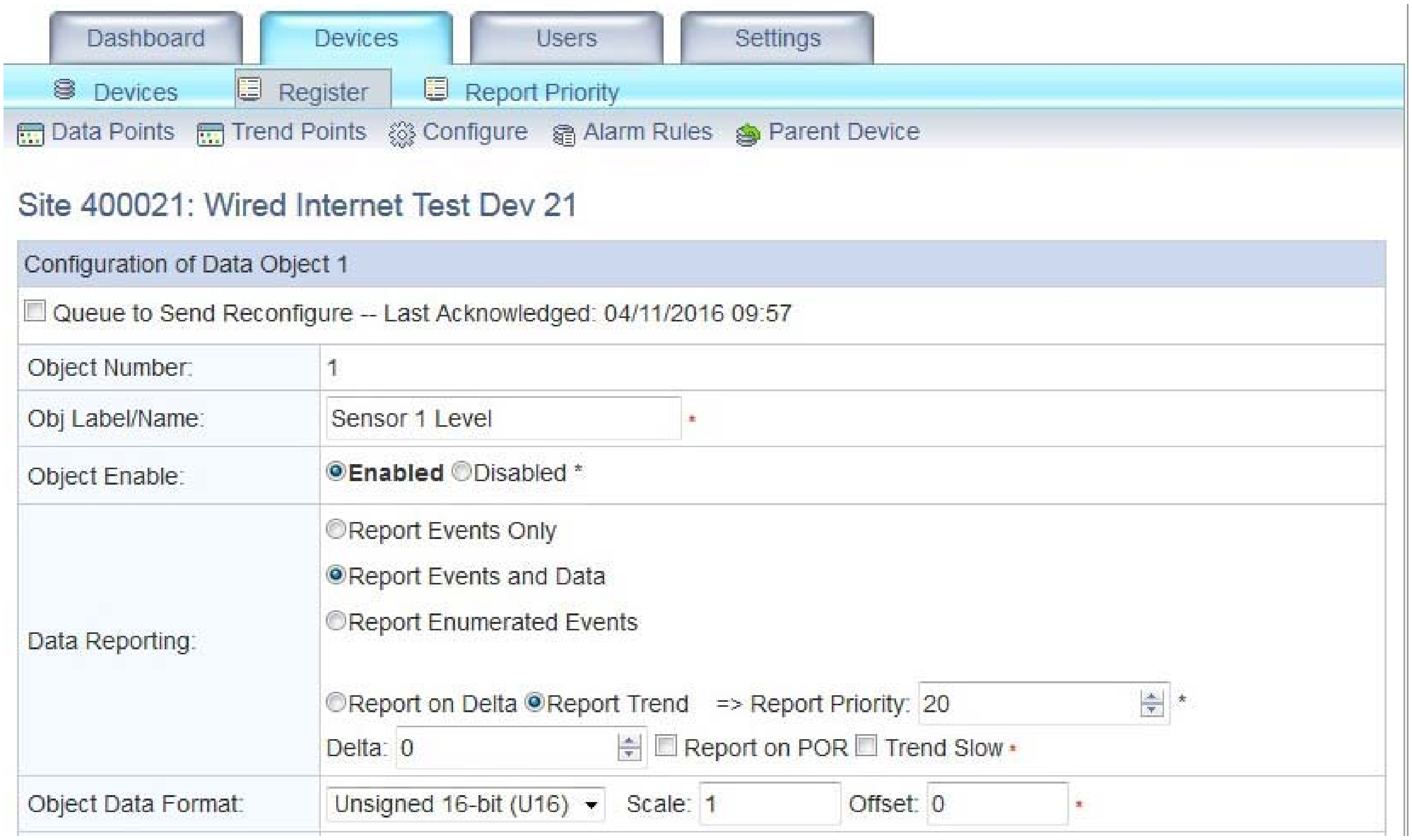
All of the configuration for data objects in the BB2-6010-Web is accessible remotely once online, and furthermore, once online, you should only make changes via the web portal. It is important to keep the device and the web portal in sync. The terms "object" and "register" are used somewhat interchangeably on the web portal, so access to object configuration is under the Register tab.

IMPORTANT: The one main thing you need to know about remote configuration is that changes are not automatically sent to the device. Once you have made all of the changes you wish to make for a given object, check the box at the top of the page marked "Queue to Send Reconfigure". Once this box is checked, the updated configuration will be sent to the device the next time it pings the server. The "Last Acknowledged" timestamp will tell you when this object was most recently reconfigured.

IMPORTANT: Configuration changes sent to the BB2-6010-Web will be auto-saved to its internal XML configuration file so that the same configuration is restored after a power outage. There is a 2-minute timeout on the auto-save to allow for multiple changes to be sent to the device before committed to the Flash file system. If the device should suffer a power outage immediately after receiving new configuration settings, these would be lost. If you have reason to believe this may have happened, simply resend the recent changes.

In addition to all of the configuration settings found in the BB2-6010-Web itself, the configuration page on the web portal has additional settings to define how it will appear on the web. You have the option of creating green-yellow-red bar graphs and assigning different objects to different dashboard pages. Click on the help link in the upper right corner of the web page on the portal for more information about any page you are on in the web portal.

The upper portion of the data object configuration page is illustrated below. This is where you remotely make the same changes that you made directly on the BB2-6010-Web during pre-configuration.



Site 400021: Wired Internet Test Dev 21

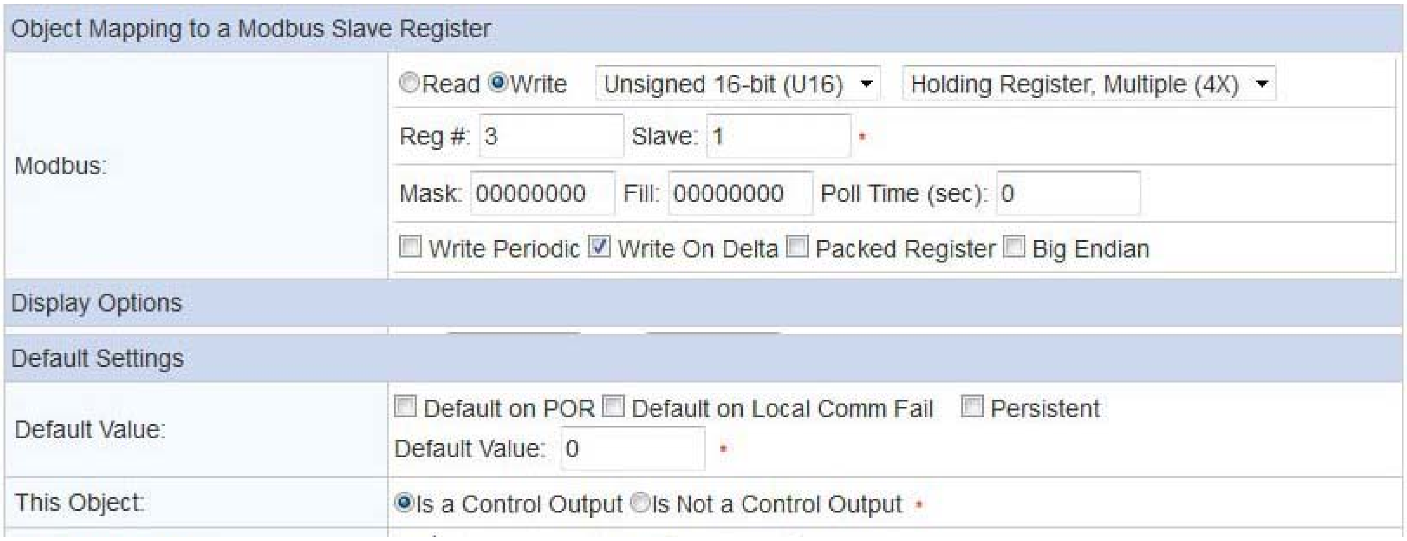
Configuration of Data Object 1

□ Queue to Send Reconfigure -- Last Acknowledged: 04/11/2016 09:57

Object Number:	1
Obj Label/Name:	Sensor 1 Level *
Object Enable:	● **Enabled** ○ Disabled *
Data Reporting:	○ Report Events Only ● Report Events and Data ○ Report Enumerated Events ○ Report on Delta ● Report Trend => Report Priority: 20 * Delta: 0 □ Report on POR □ Trend Slow *
Object Data Format:	Unsigned 16-bit (U16) Scale: 1 Offset: 0 *

9.2. Writing to Modbus Registers from Web Portal

Configure the data object to write to Modbus as discussed under Modbus Mapping in section 6.



Object Mapping to a Modbus Slave Register

Modbus:	○ Read ● Write Unsigned 16-bit (U16) Holding Register, Multiple (4X) Reg #: 3 Slave: 1 * Mask: 00000000 Fill: 00000000 Poll Time (sec): 0 □ Write Periodic ☑ Write On Delta □ Packed Register □ Big Endian

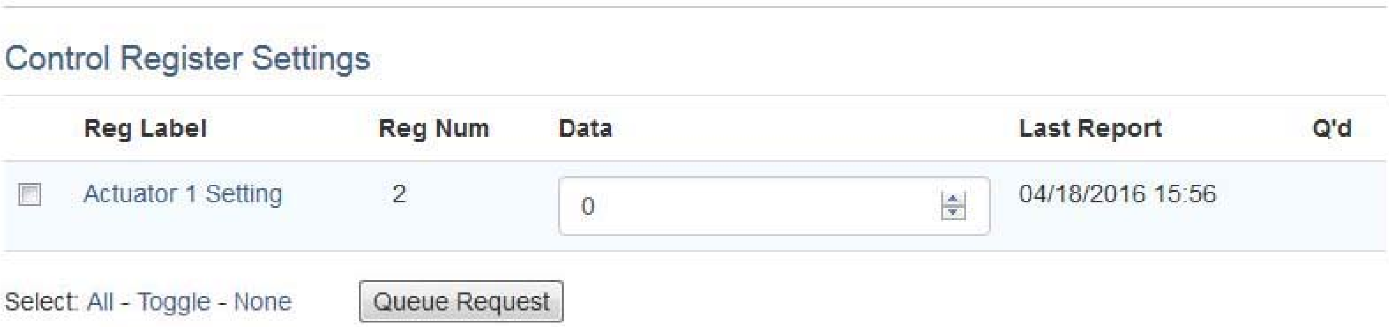
Display Options

Default Settings

Default Value:	□ Default on POR □ Default on Local Comm Fail □ Persistent Default Value: 0 *
This Object:	● Is a Control Output ○ Is Not a Control Output *

The one setting you need to make after uploading your BB2-6010-Web configuration to the web portal is to select "Is a Control Output" at the bottom of the object configuration page. Once you do this, the object will appear in the Control Register Settings section of the Diagnostics page on the portal.

To write a new value to the Modbus register, enter a value in the Data window, check the box in front of the object name (Reg Label), and click Queue Request. The next time the BB2-6010-Web pings the server, it will receive your new data value, and the BB2-6010-Web will then write that value to the Modbus register (assuming the object had been configured to write to Modbus).



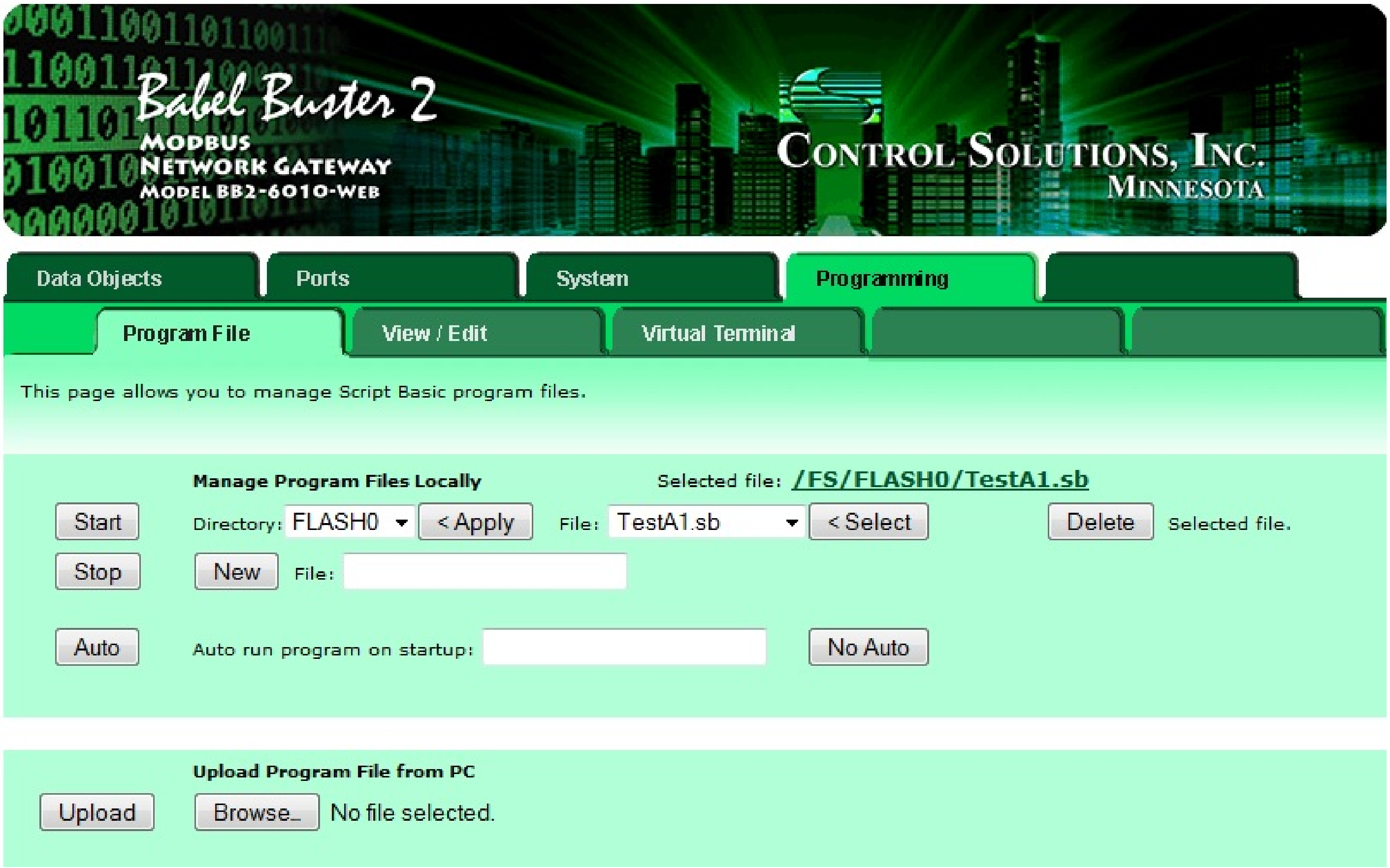
Control Register Settings

	Reg Label	Reg Num	Data	Last Report	Q'd
□	Actuator 1 Setting	2	0	04/18/2016 15:56	

10. Programming with Script Basic

10.1. Creating a Program

Start by selecting the FLASH0 directory (unless your program will be very temporary). Enter a new file name, and click the New button. You now have a new blank slate to fill. Your program name should always have a *.sb suffix. If you are going to edit an existing program, find it in the File list, and click Select.



This page allows you to manage Script Basic program files.

Manage Program Files Locally

Selected file: /FS/FLASH0/TestA1.sb

Directory: FLASH0

File: TestA1.sb

Selected file.

File:

Auto run program on startup:

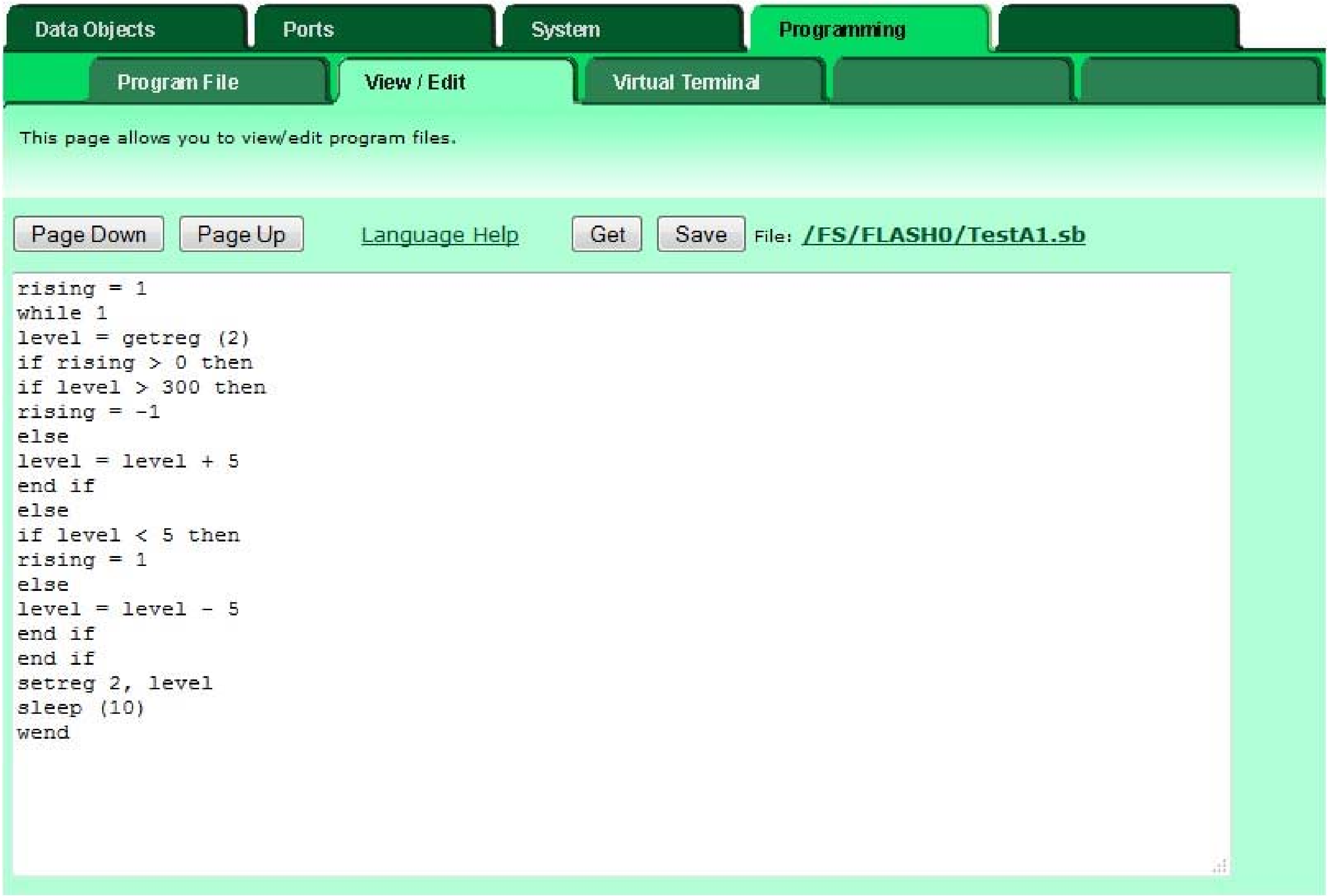
Upload Program File from PC

No file selected.

There is a local, very simple text editor available via the web View/Edit page. To edit an existing file, start by clicking Get. After entering a new program, or editing an existing program, click Save.

It is recommended that you use an external text editor for large programs, and simply upload it on the Program File page.

The Language Help link provides a summary of Script Basic. For a complete reference, use the external compiled help file available for Script Basic.



This page allows you to view/edit program files.

```
rising = 1
while 1
level = getreg (2)
if rising > 0 then
if level > 300 then
rising = -1
else
level = level + 5
end if
else
if level < 5 then
rising = 1
else
level = level - 5
end if
end if
setreg 2, level
sleep (10)
wend
```

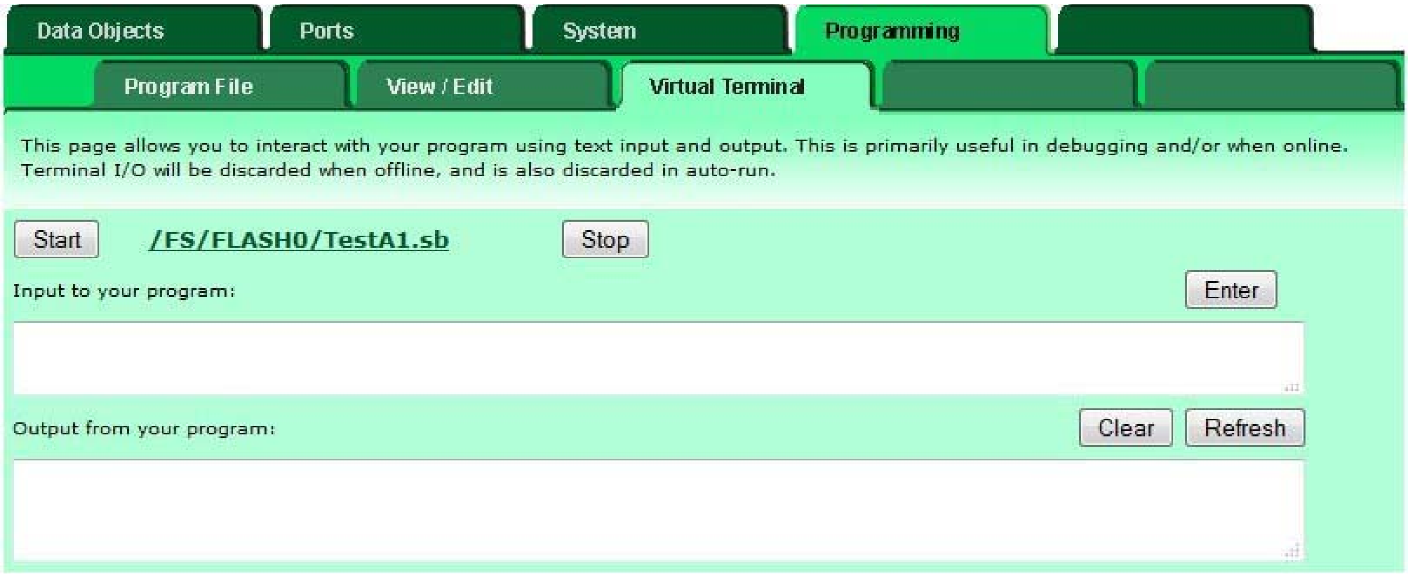
10.2. Testing the Program

The virtual terminal can be used while testing programs. Any "print" statement will send its output to the Output window, and any "line input" statement will take input from the Input window. The print and line input statements will have no effect when running in the background (i.e. running as Auto run from startup).

The Virtual Terminal page does not auto refresh, therefore any output produced by a print statement will not appear until you click the Refresh button. Some initial output may appear immediately since the program began running faster than the initial page refresh that occurs after clicking Start, but you will need to click Refresh to see additional output.

WARNING: If you are testing a program, you should select No Auto (see below), then restart the BB2-6010-Web. You will have two programs running at the same time if the auto run program is running and you are also running a program here. The results can be unpredictable if you are running two copies of the same or similar program at the same time.

An abbreviated screen shot of the Virtual Terminal is illustrated below.



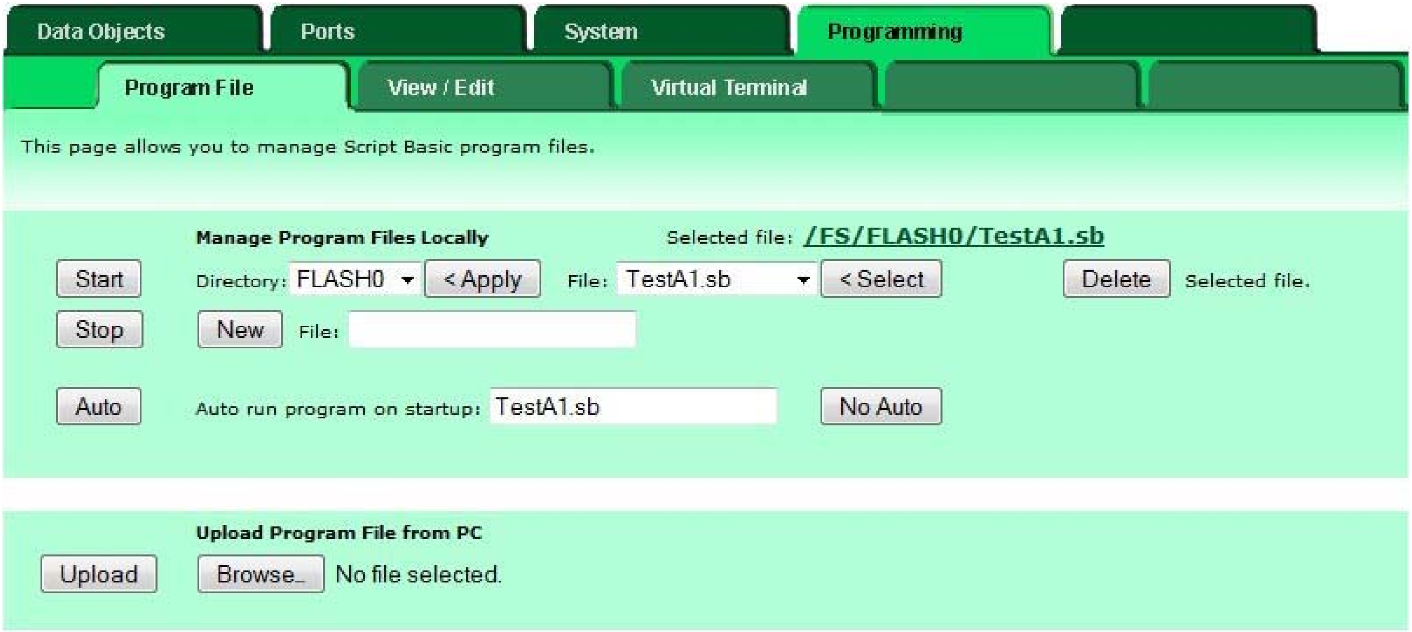
This page allows you to interact with your program using text input and output. This is primarily useful in debugging and/or when online. Terminal I/O will be discarded when offline, and is also discarded in auto-run.

/FS/FLASH0/TestA1.sb

10.3. Setting the Program to Auto-Run on Startup

Your program will not be very useful if it does not start up when the BB2-6010-Web boots up. You cause that to happen by selecting your file from the list, and then clicking the Auto button. Setting the auto run program will take several seconds as Flash memory is being reprogrammed at this point.

From this point on, your program named in the "Auto run program on startup" window will be automatically started upon bootup. Of course, if the program has errors, it might not keep on running. Be sure to test your program ahead of time, and also consider use of the "On Error" feature of Basic. What exactly you might do upon error is entirely up to you, but one potentially useful option is to set an error number of your own making in a specific data object that will be reported to the web portal using "Report on Delta".



This page allows you to manage Script Basic program files.

Selected file: /FS/FLASH0/TestA1.sb

Directory: FLASH0

File: TestA1.sb

File:

Auto run program on startup: TestA1.sb

No file selected.

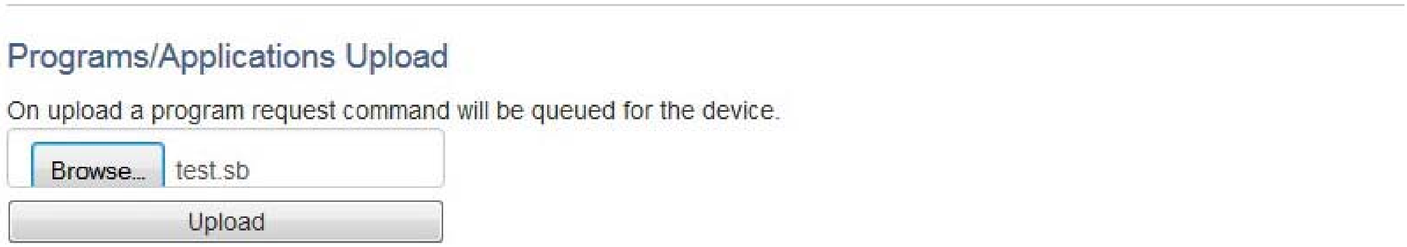
To eliminate the auto run program, simply click the No Auto button. Clicking the

No Auto button will only make certain no program is started upon bootup from this point forward. You will need to restart the BB2-6010-Web to halt any program that was already running from the last bootup.

10.4. Uploading Program via Web Portal

You have the ability to make program changes, or even create a new program, and upload that to the remote BB2-6010-Web. Find the "Programs/Applications Upload" section on the Diagnostics page for your device on the web portal. Click Browse to find the program file which should have a *.sb suffix. Click Upload. The program will be uploaded the next time the device pings the server. Whatever file is named here will also be selected as the new "Auto run" program in the device.

Be aware that you have little recourse for testing your program remotely. It is recommended that you have a second (local) BB2-6010-Web for test purposes. If you need to simulate the entire remote system, you can do that with yet another local BB2-6010-Web set to operate in Wildcard Emulation mode.



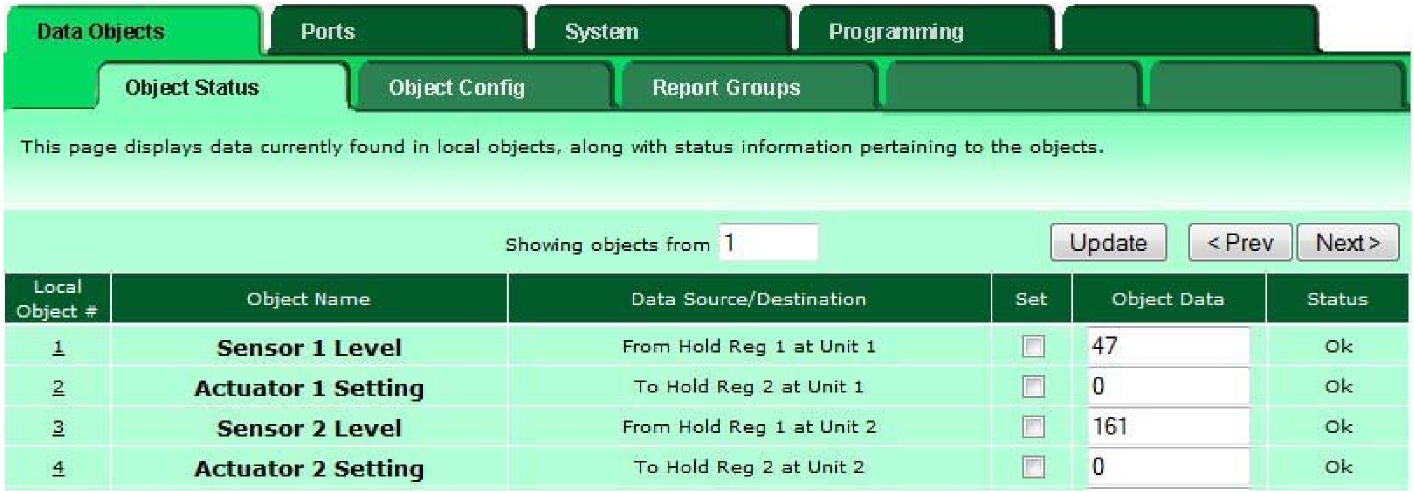
Programs/Applications Upload

On upload a program request command will be queued for the device.

11. Trouble Shooting

11.1. Object Status Indications

Status by object is displayed on the Object Status page. The status column should normally just show "Ok".

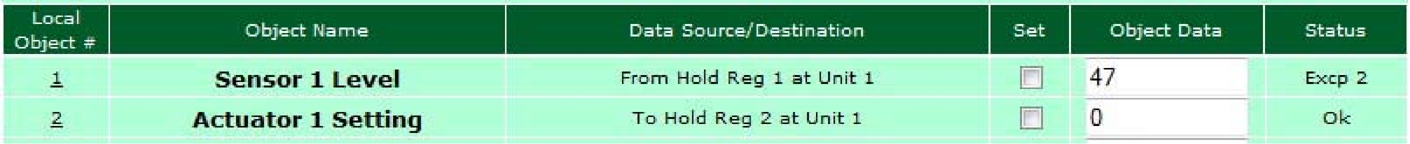


This page displays data currently found in local objects, along with status information pertaining to the objects.

Showing objects from 1

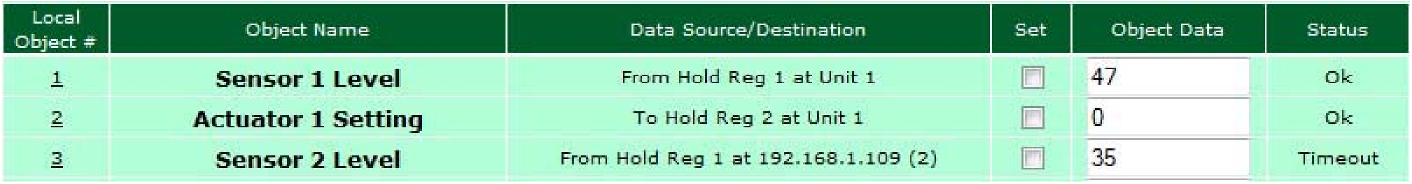
Local Object #	Object Name	Data Source/Destination	Set	Object Data	Status
1	Sensor 1 Level	From Hold Reg 1 at Unit 1	☐	47	Ok
2	Actuator 1 Setting	To Hold Reg 2 at Unit 1	☐	0	Ok
3	Sensor 2 Level	From Hold Reg 1 at Unit 2	☐	161	Ok
4	Actuator 2 Setting	To Hold Reg 2 at Unit 2	☐	0	Ok

If the data object is configured to poll a Modbus slave, and that slave returned an exception error, that exception is shown in the Status column. Exception errors suggest problems with the configuration. The definitions of the exception codes are listed in the Quick Help section at the bottom of the RTU Errors page.



Local Object #	Object Name	Data Source/Destination	Set	Object Data	Status
1	Sensor 1 Level	From Hold Reg 1 at Unit 1	☐	47	Excp 2
2	Actuator 1 Setting	To Hold Reg 2 at Unit 1	☐	0	Ok

If a Modbus slave device simply does not respond, it will be indicated by "Timeout" in the Status column. This can indicate any of several possible problems, including wiring problems, wrong slave address, wrong baud rate, etc. If you see CRC errors indicated either here or on the RTU errors page, that is almost always a wiring issue. CRC errors are more infrequently due to wrong parity, but most frequently a wiring problem, especially missing ground.



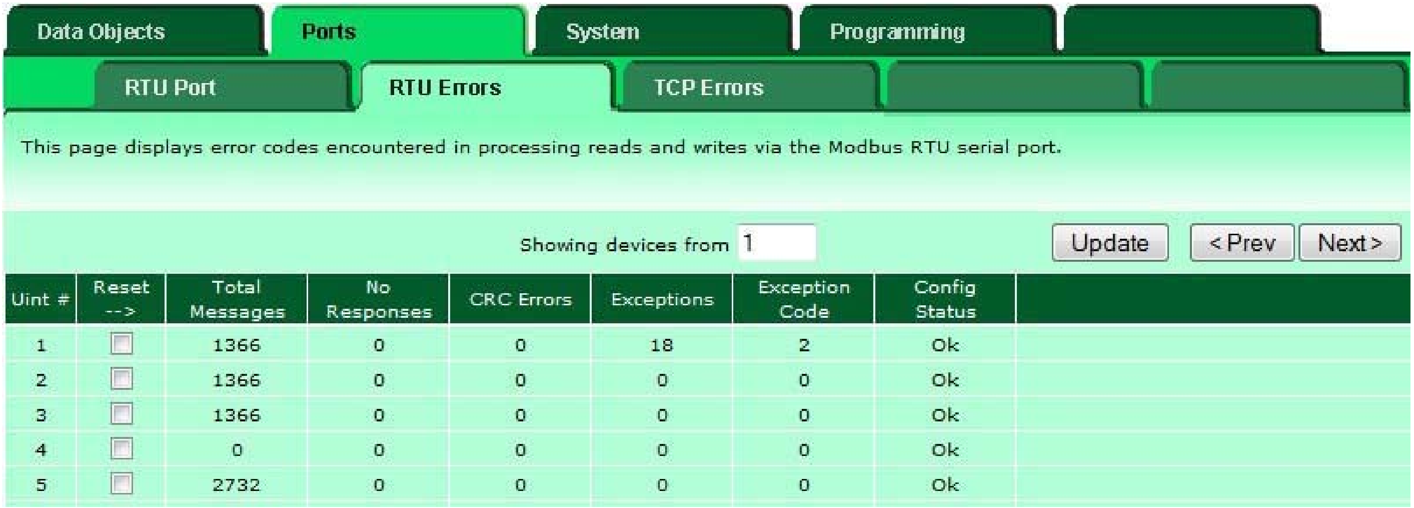
Local Object #	Object Name	Data Source/Destination	Set	Object Data	Status
1	**Sensor 1 Level**	From Hold Reg 1 at Unit 1	□	47	Ok
2	**Actuator 1 Setting**	To Hold Reg 2 at Unit 1	□	0	Ok
3	**Sensor 2 Level**	From Hold Reg 1 at 192.168.1.109 (2)	□	35	Timeout

The Status column can also indicate "Fault" if there was a problem that prevents BB2-6010-Web from attempting to process the polling configuration for this object. The absence of "Fault" does not mean the configuration is fault free. Attempting to poll a slave that does not exist, for example, is certainly a fault but is not a fault that the BB2-6010-Web can detect other than by indicating "Timeout" when it attempted to poll that slave that it does not know is non-existent.

11.2. Modbus RTU Errors

Modbus RTU errors are listed by device (slave address or unit number) on the RTU Errors page when the BB2-6010-Web is Modbus master. The Total Messages should be an ever increasing count. If total message count is zero, or it is not increasing from time to time when the page is refreshed, then the BB2-6010-Web is probably not set to be Modbus master, or there are no objects configured to poll Modbus.

If there are errors, they will be indicated by counts in the respective columns. The Exception Code is not a count, but rather the actual exception code returned by the slave device.



This page displays error codes encountered in processing reads and writes via the Modbus RTU serial port.

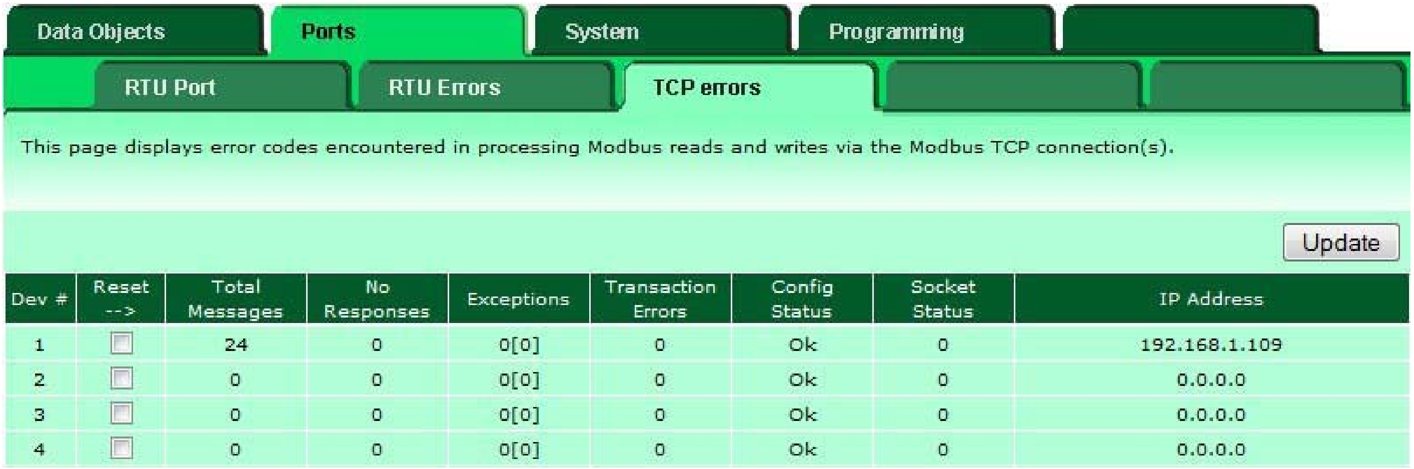
Showing devices from 1

Uint #	Reset -->	Total Messages	No Responses	CRC Errors	Exceptions	Exception Code	Config Status
1	□	1366	0	0	18	2	Ok
2	□	1366	0	0	0	0	Ok
3	□	1366	0	0	0	0	Ok
4	□	0	0	0	0	0	Ok
5	□	2732	0	0	0	0	Ok

The Config Status column will indicate "Ok" if no configuration errors were detected by the BB2-6010-Web. It will indicate "Fault" if there was a problem that prevents BB2-6010-Web from attempting to process the polling configuration for an object.

11.3. Modbus TCP Errors

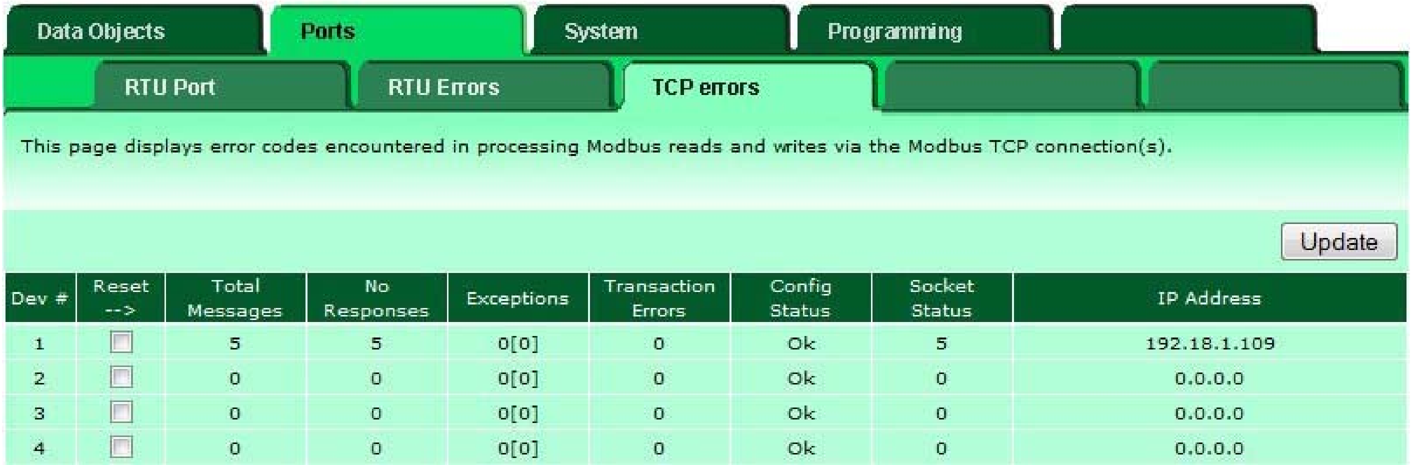
Modbus TCP connection problems and errors related to polling Modbus devices via TCP are indicated on the TCP Errors page. A good connection with successful polling should show an increasing Total Messages count, and zero for everything else.



This page displays error codes encountered in processing Modbus reads and writes via the Modbus TCP connection(s).

Dev #	Reset -->	Total Messages	No Responses	Exceptions	Transaction Errors	Config Status	Socket Status	IP Address
1	□	24	0	0[0]	0	Ok	0	192.168.1.109
2	□	0	0	0[0]	0	Ok	0	0.0.0.0
3	□	0	0	0[0]	0	Ok	0	0.0.0.0
4	□	0	0	0[0]	0	Ok	0	0.0.0.0

A non-zero count in the No Responses column usually indicates a connection problem. If the TCP slave returned an exception error in response to a request, this would indicate a configuration problem. You would need to view object status to see which particular object is causing problems.



This page displays error codes encountered in processing Modbus reads and writes via the Modbus TCP connection(s).

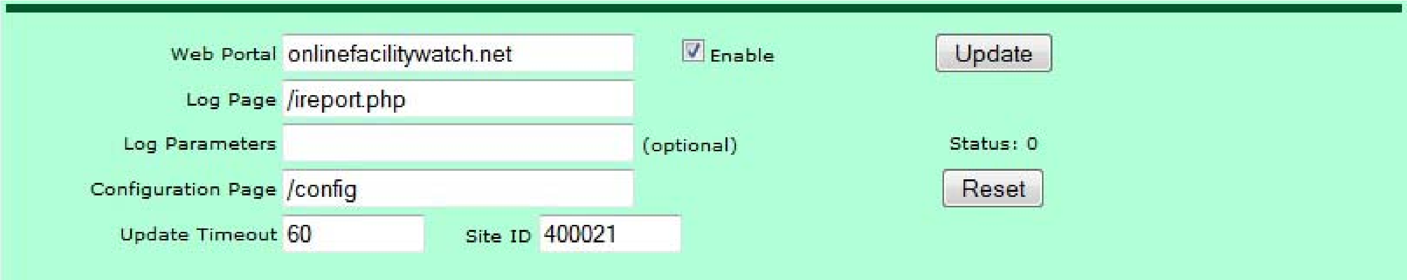
Dev #	Reset -->	Total Messages	No Responses	Exceptions	Transaction Errors	Config Status	Socket Status	IP Address
1	□	5	5	0[0]	0	Ok	5	192.18.1.109
2	□	0	0	0[0]	0	Ok	0	0.0.0.0
3	□	0	0	0[0]	0	Ok	0	0.0.0.0
4	□	0	0	0[0]	0	Ok	0	0.0.0.0

A non-zero value for Socket Status means there is a problem. The codes likely to happen are listed in the Quick Help section at the bottom of the TCP errors page. A socket status of 5 is common - it means the IP address shown cannot be reached.

The Config Status column will indicate "Ok" if no configuration errors were detected by the BB2-6010-Web. It will indicate "Fault" if there was a problem that prevents BB2-6010-Web from attempting to process the polling configuration for an object.

11.4. Portal Connection Errors

If the BB2-6010-Web is unable to connect to the web portal server for some reason, the cause is likely to be displayed as a non-zero value for Status in the web portal section of the Network setup page. The codes most likely to appear are listed in the Quick Help section at the bottom of the Network page.



Web Portal onlinefacilitywatch.net ☑ Enable

Log Page /ireport.php

Log Parameters (optional)

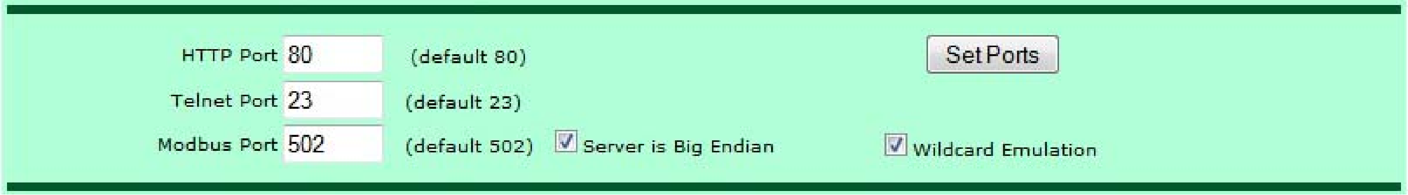
Configuration Page /config

Update Timeout 60 Site ID 400021

Status: 0

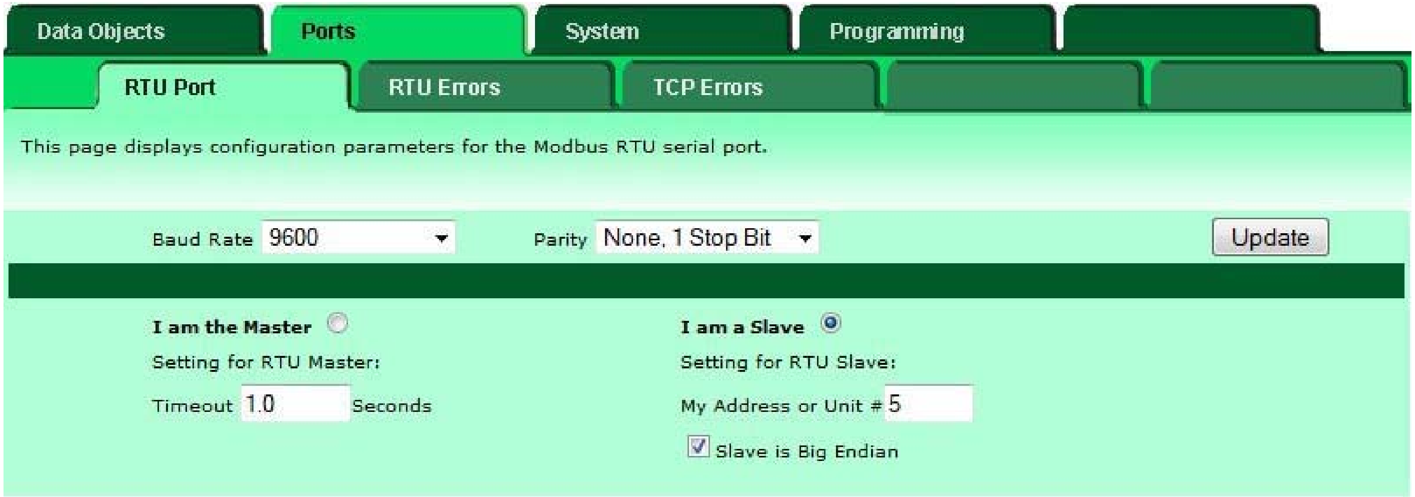
12. Testing with Wildcard Simulator

The BB2-6010-Web includes a feature making it useful as a test simulator for another BB2-6010-Web device. Once you have configured a BB2-6010-Web to report to the web portal, polling any number of Modbus devices, you can make the simulator pretend to be all of those devices at the same time. To turn a BB2-6010-Web device into a simulator, go to the Network setup page, and check Wildcard Emulation, then click Set Ports.


HTTP Port 80 (default 80)
Telnet Port 23 (default 23)
Modbus Port 502 (default 502) ☑ Server is Big Endian ☑ Wildcard Emulation

To set up a second BB2-6010-Web device as the simulator, export the XML configuration file from the device to be tested. Obtain this XML file the same way you would for uploading it to the web portal. In addition to uploading that XML file to the web portal, you will also upload it to the second BB2-6010-Web acting as the simulator (via the simulator's Config File page).

Once you have uploaded the configuration to the simulator, there is one change you will need to make manually. Assuming the device to be tested is acting as Modbus RTU master, you need to change the simulator device to be Modbus slave. Once you make this change, go to the Config File page and click Save so that your simulator boots up still simulating correctly should you power cycle the device.



This page displays configuration parameters for the Modbus RTU serial port.

Baud Rate 9600 Parity None, 1 Stop Bit

I am the Master ○
Setting for RTU Master:
Timeout 1.0 Seconds

I am a Slave ●
Setting for RTU Slave:
My Address or Unit # 5
☑ Slave is Big Endian

Now, instead of connecting the BB2-6010-Web to be tested to its normal Modbus network, connect it instead to only the simulator. Do not have any other Modbus devices connected to the Modbus RTU network as the simulator will reply to any and all slave addresses known to it in its configuration.

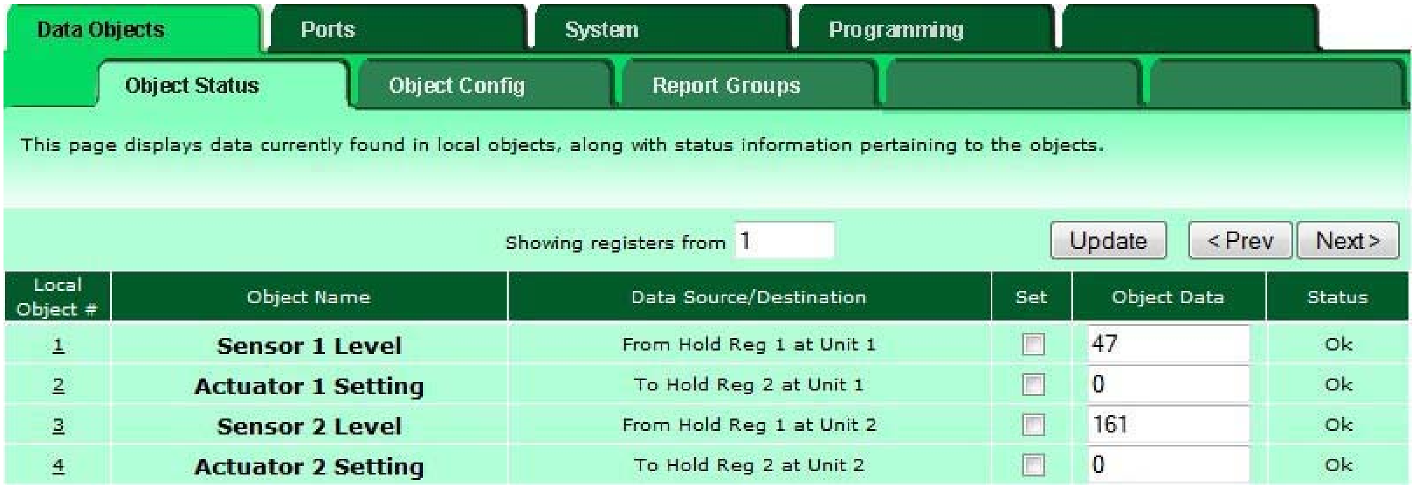
Testing Modbus TCP is a little tricker because the simulator cannot have a wildcard IP address. You will need to temporarily change slave IP addresses in the device being tested so that they point to the simulator.

Of course, you can also use ModSim and ModScan to do all necessary Modbus testing. The BB2-6010-Web wildcard emulation is simply one option available. The wildcard emulation is most useful for longer term testing where you do not want to tie up your PC with ModSim.

The Wildcard Emulation is only intended to simulate multiple RTU slaves to test a BB2-6010-Web that is going to function as a Modbus master in its final application. If the BB2-6010-Web device reporting to the web portal is going to function as a Modbus slave in its final application, then it will have a single slave address, and any Modbus master can be used to exercise it, including a second BB2-6010-Web not running in Wildcard Emulation mode. Because there was no Modbus mapping in the BB2-6010-Web intended to be a slave, the configuration from that device will not result in any polling by the simulation device, and therefore Wildcard Emulation does not work in reverse.

You may go to the Data Objects, Object Status page and change values for any of the objects. The device being tested should now receive these values as its data. You can even write a Script Basic program on the simulator to cause an object's value to increase and decrease over time. This is actually what the example program shown in section 10.1 is doing.

To make changes to data values in the simulator, go to the Object Status page, enter something for Object Data, check the Set box, and click Update. Please note that the data source/destination displayed is as it would be in the device being tested. The indication displayed here in the simulator will be opposite what it actually is doing in the simulator. Also, disregard any Status indications. Any errors reported by the simulator are generally going to be invalid, especially timeout errors. The error messages that matter are on the device being tested.



This page displays data currently found in local objects, along with status information pertaining to the objects.

Showing registers from 1

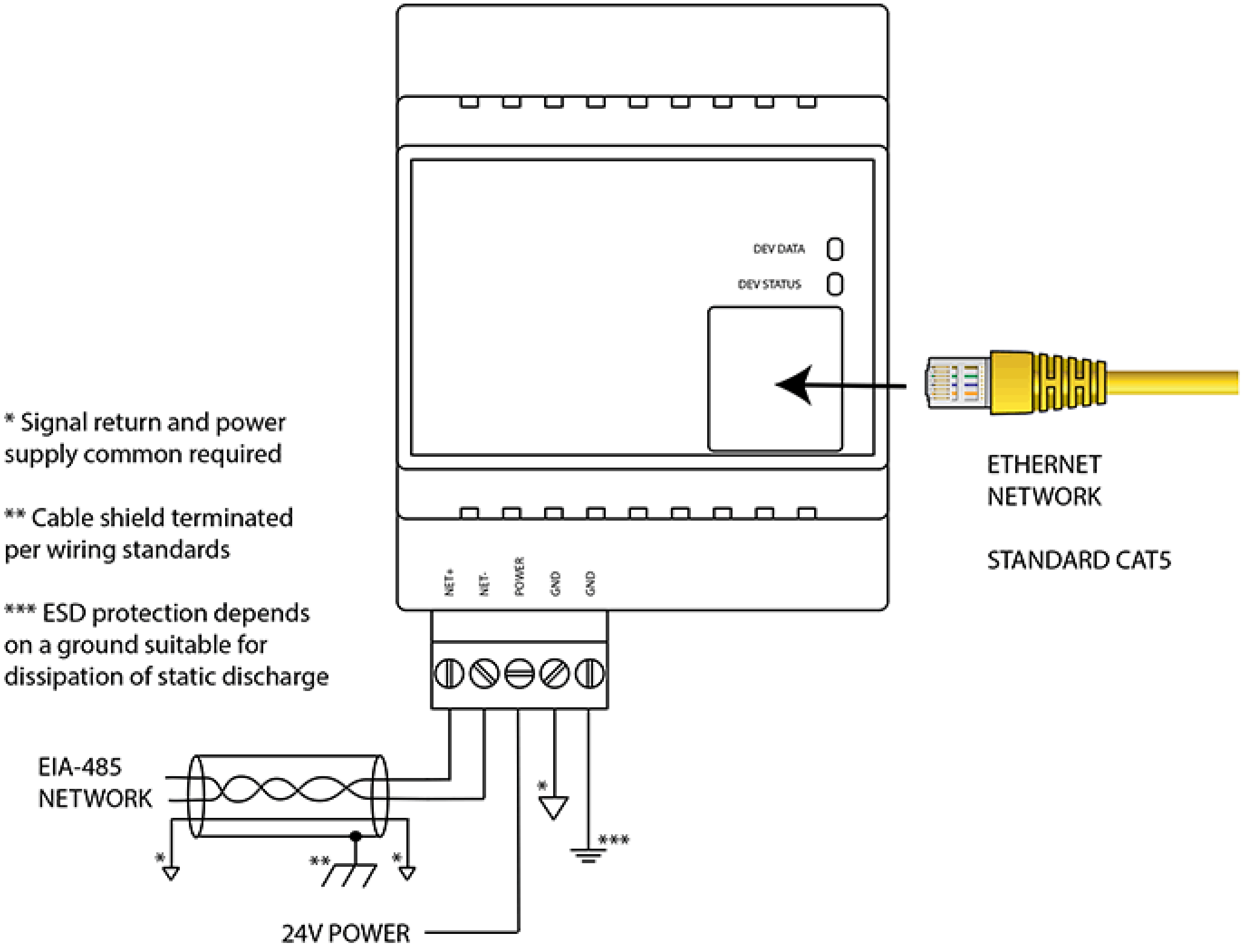
Local Object #	Object Name	Data Source/Destination	Set	Object Data	Status
1	**Sensor 1 Level**	From Hold Reg 1 at Unit 1	□	47	Ok
2	**Actuator 1 Setting**	To Hold Reg 2 at Unit 1	□	0	Ok
3	**Sensor 2 Level**	From Hold Reg 1 at Unit 2	□	161	Ok
4	**Actuator 2 Setting**	To Hold Reg 2 at Unit 2	□	0	Ok

The XML file loaded into a BB2-6010-Web device is in the same format as used in VP3-2290 cellular devices, except for the tracking section found in a VP3-2290 file. You can also use the BB2-6010-Web as a simulator for cellular devices. Obtain the VP3-2290's XML file by connecting to it via USB. Use its "xsave" command to save its XML file. Next, power down the VP3-2290, remove its flash card, insert into a card reader on a PC, and retrieve the XML file saved. Now edit this XML file using a generic text editor (Windows Notepad works), and delete the tracking section (the <tracking> and </tracking> tags and everything between them). Save this minimally modified file and upload this file to the BB2-6010-Web simulator.

Appendix A Hardware Details

A.1 Wiring

Wiring for the Babel Buster BB2-6010-Web is illustrated below.


DEV DATA
DEV STATUS
ETHERNET
NETWORK
STANDARD CAT5
* Signal return and power supply common required
** Cable shield terminated per wiring standards
*** ESD protection depends on a ground suitable for dissipation of static discharge
NET+
NET-
POWER
GND
GND
EIA-485
NETWORK
*
**
*
*
***
24V POWER

Wire the gateway as illustrated. Follow all conventional standards for wiring of EIA-485 networks when connecting the Modbus RTU EIA-485 (RS485) network. This includes use and termination of shield, termination of the network, and grounding.

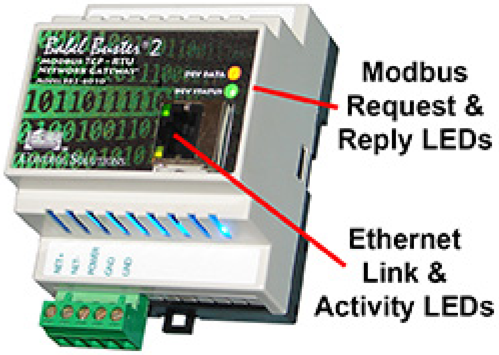
IMPORTANT: Although EIA-485 (RS485) is thought of as a 2-wire network, you MUST include a third conductor connected to GND or common at each device so that all devices are operating at close to the same ground potential. Proper grounding of equipment should ensure proper operation without the third conductor; however, proper grounding often cannot be relied upon. If large common mode voltages are present, you may even need to insert optically isolated repeaters between EIA-485 devices.

Use standard CAT5 cables for Ethernet connections. Use control wire as applicable for local electrical codes for connecting the 24V (AC or DC) power supply.

Note that in addition to connecting power supply common to a GND terminal, you must also connect a GND terminal to earth ground in order to ensure proper ESD protection.

A.2 Front Panel LED Indicators

Power-up LED behavior: Following a server boot-up delay, all LEDs on front panel will turn on yellow or red for half a second, then all will turn on green for half a second. Then they will proceed to indicate as normally defined for the indicators.



Babel Buster BB2-6010-Web request/reply LEDs reflect RTU traffic while the Ethernet activity LED will indicate TCP traffic. To see TCP errors, one needs to look at the Errors page in the web UI. (Note: If Modbus RTU is not configured for any use, then the request/reply LEDs will reflect TCP traffic; however, normally the only indication of TCP traffic is the Ethernet activity LEDs.)

Babel Buster BB2-6010-Web LEDs indicate as follows (LEDs are bi-color):

DEV DATA	Flashes yellow each time a request is sent when operating as Modbus Master, or each time a request is received when operating as Modbus Slave.
DEV STATUS	Operating as Modbus Master, flashes green each time a good response is received, or red when an error code is received, the request times out, or there is a flaw in the response such as CRC error. Operating as Modbus Slave, flashes green each time a good

	response is sent, or red if an exception code is sent (meaning the received request resulted in an error).

Ethernet link LED is the yellow LED integrated into the CAT5 connector. Ethernet activity LED is the green LED integrated into the CAT5 connector.

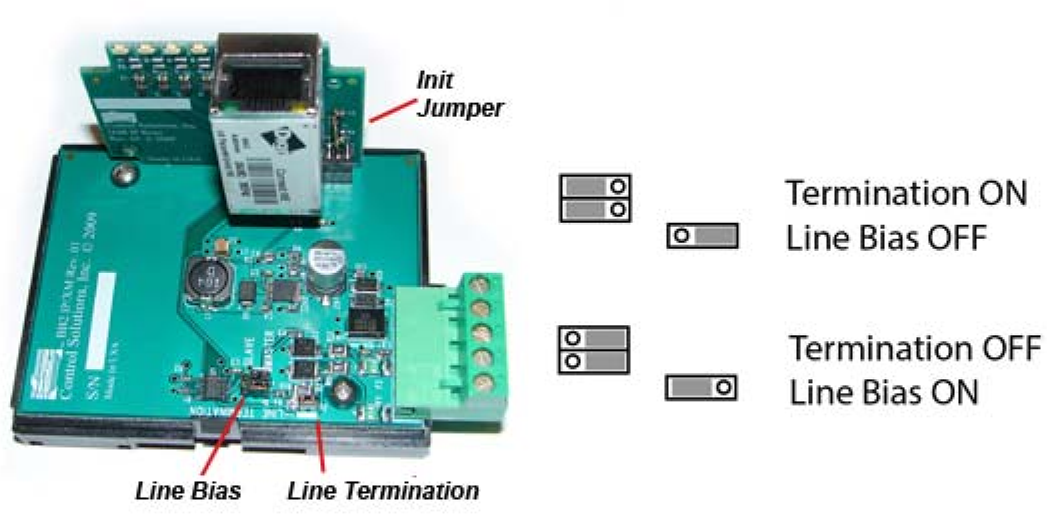
A.3 RS-485 Line Termination & Bias

Enable line termination only when this device is placed at the end of the network. Termination should only be enabled at two points on the network, and these two points must be specifically the end points.

Enable line bias when needed. Line bias should only be enabled at one point on the network, and does not have to be the end point. Line bias holds the line in a known neutral state when no devices are transmitting. Without bias, the transition from offline to online by a transmitter can look like a false start bit and cause loss of communication.

The line conditioning options are enabled when the respective shunt is moved to the position indicated by the white block next to the 3-pin header. Putting the shunt on the opposite 2 pins disables the option, and is simply a place to store the shunt.

Jumper locations for Babel Buster BB2-6010-Web:



The "Init" jumper on the server module should only be used when advised by tech support. Installing this jumper prior to power-up causes the server to go into firmware update mode.