

AcquiSuite[®]

Modbus Framework

Date Jun 1, 2010

Copyright Information

Copyright © 2007 - 2010 by Obvius

Obvius and **AcquiSuite** are trademarks of Obvius Holdings LLC

Other brand and product names are trademarks or registered trademarks of their respective holders.

U.S. Government Restricted Rights: Use, duplication or disclosure by the Government is subject to restrictions set fourth in subparagraph (a) through (d) of the Commercial Computer Restricted Rights clause at FAR 52.227-19 when applicable, or subparagraph (c) (1) (ii) of the Rights in Technical Data and Computer Software clause at DFARS 252.227-7013, and in similar clauses in the NASA FAR Supplement.

The Linux operating system, busybox, tinylogin, wuftp, sysvinit and other software products are covered by the GPL (Gnu Public License) A copy of the GPL and all software components covered by the GPL are available at www.obvius.com/gnu.

Limited Warranty

OBVIUS IS PROVIDING THIS WARRANTY IN LIEU OF ALL OTHER EXPRESS OR IMPLIED WARRANTIES, INCLUDING ANY WARRANTY OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. THIS WARRANTY IS BUYER'S EXCLUSIVE REMEDY FOR ALL CLAIMS AGAINST OBVIUS. OBVIUS SHALL NOT BE LIABLE FOR ANY CONSEQUENTIAL OR INCIDENTAL DAMAGES. OBVIUS'S TOTAL LIABILITY FOR ALL CLAIMS SHALL BE LIMITED TO THE PRICE PAID FOR ITS PRODUCT.

Obvius promises buyer that any standard product manufactured by Obvius shall be free from all material defects in design, material, or manufacturing for a period of 2 years from the manufacture date; provided, however, that the warranty shall not extend to ordinary wear and tear or to normally replaceable components (e.g., batteries). During the warranty period, Obvius may repair or replace (in its sole discretion) any product suffering from a warranty defect and returned freight prepaid by buyer, with no charge to buyer for any warranty repair or replacement. The warranty shall remain in full force and effect for such 2 year period, provided that the product: (1) was installed, operated, and maintained properly; (2) has not been abused or misused; (3) has not been repaired, altered, or modified outside of Obvius's authorized facilities; (4) has not been sold subject to other warranty terms specified at the time of sale; and (5) is still owned by the original purchaser. This warranty provides specific legal rights that may be varied by state law. Obvius's products are not designed for life or safety applications.

Product Application Limitation

Obvius products are not intended for use in critical applications such as nuclear facilities, human implantable devices or life support. Obvius is not liable, in whole or in part, for any claims or damages arising from such uses.

Obvius strongly believes in continuous improvement, therefore we must reserve the right to change specifications and product offerings without notice. Where possible, we will substitute products with equivalent functionality when necessary.

Obvius
3300 NW 211th Terrace
Hillsboro, OR 97124
ph: 503-601-2099
www.obvius.com

Overview

This document covers the Modbus Framework feature in the AcquiSuite firmware. The Modbus Framework provides the ability to communicate with any Modbus RS485/RTU device. The Modbus Framework supports multiple Modbus commands including read coils, input and holding registers. Registers may be formatted with a number of translations including integer, floating point, 16 or 32bit, as well as fixed multipliers or slope/offset scaling.

This document assumes the reader is already familiar with Modbus devices and the Modbus protocol. For more information on the Modbus protocol, please visit <http://www.modbus.org>

Before you begin, the reader must obtain proper Modbus register documentation from the hardware vendor for each Modbus device that will be used. Without this documentation, it will not be possible to read data from the Modbus device.

Communications:

Before the Modbus Framework can be used to access a Modbus device, the Modbus device must be connected to the AcquiSuite and it must be configured for proper communication parameters.

Basic configuration parameters include the following:

- Modbus RTU
- RS485 two wire
- No Parity, 8 data bits, 1 stop bit (N81)
- 9600 baud rate. (19200 is also supported)
- Unique Modbus Address

It is important to configure the new Modbus device to match all the other devices on the RS485 network. For example, the baud rate for all devices should be set to the same rate. In addition, it is important to make sure that each device has a unique address. If multiple Modbus devices have the same address, neither will be able to communicate with the AcquiSuite.

Additionally, the Framework can support Modbus devices behind a remote Modbus/TCP gateway, or devices that have native Modbus/TCP support.

Once the device is configured and is operational, the device should appear in the AcquiSuite Modbus Device List. The device list is organized by Modbus address. The new device will be listed with “unknown device” in the device type column. This indicates the device is communicating properly with the AcquiSuite, however the AcquiSuite does not recognize it as a device with a built-in driver.

If the new device does not show up in the device list, there is a communications problem. This must be corrected before you can proceed with setting up any further options in the Modbus Framework.

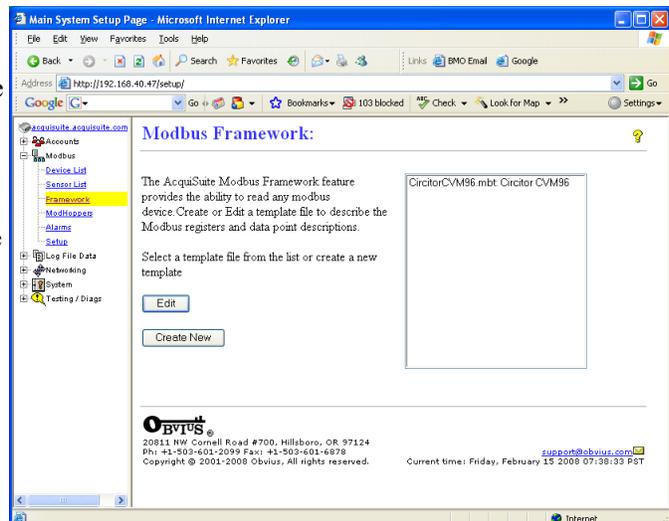
Create a new template:

To create a new template for a Modbus device, first select the Framework option from the Modbus Menu on the left side of the screen. The following page will be presented.

From this screen you may create a new template or edit an existing one. To edit an existing template, click the template name on the list, and click the edit button.

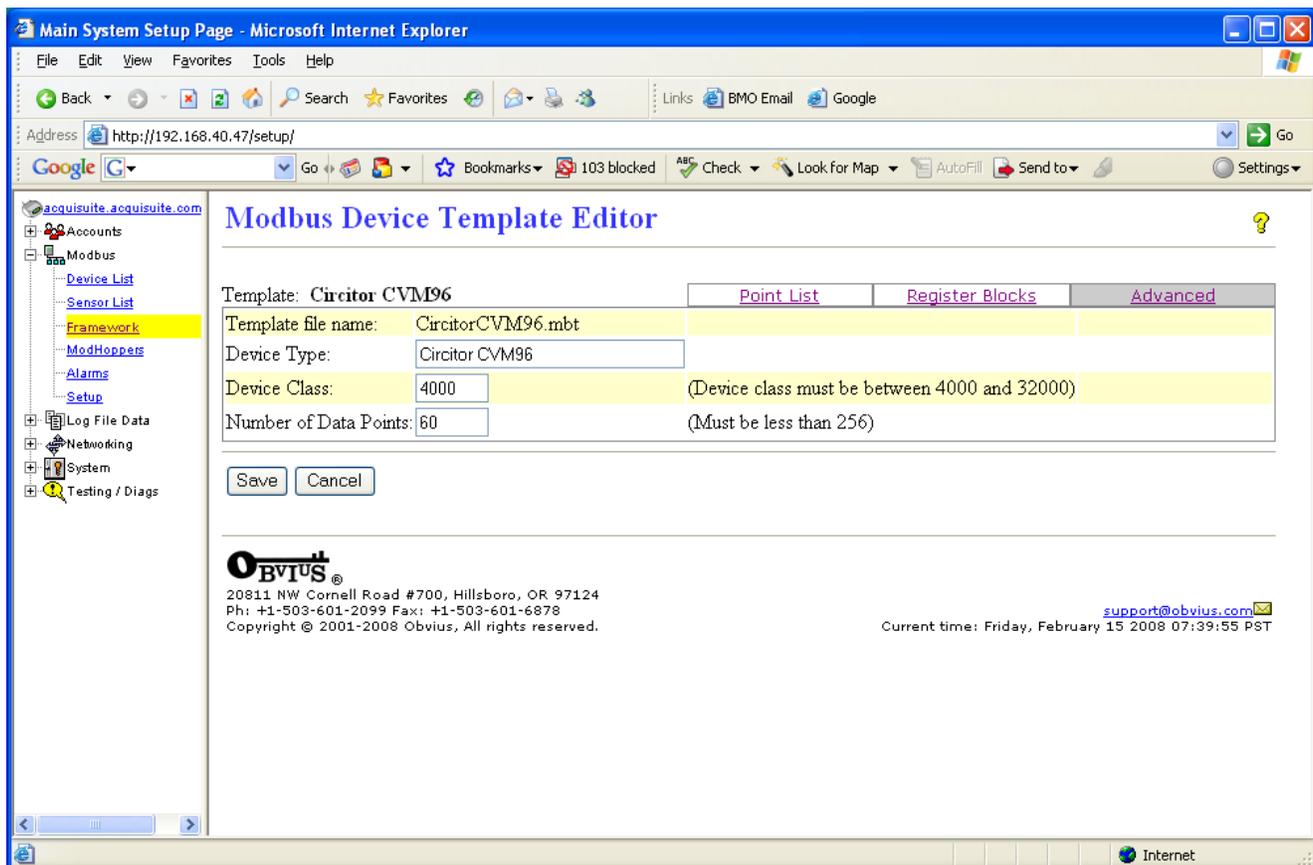
To create a new template, click the “create new” option and you will be prompted for a name. Template names should use standard letters and numbers. Avoid characters such as “!,:/” etc.

Once a template has been created it will appear in the list as shown in the example to the right.



Template Advanced Settings:

The first option to configure for a template is the advanced and template wide features. These include the template name and the number of data points you wish to configure.



Fill in the field for the **Template Name**. This is the name that will be presented to the user when accessing the device. This field may have a full range of characters including spaces, slashes, and punctuation, however it may not contain quotations.

The **Device Class** field is the number that is used by the AcquiSuite to identify the modbus device when data files are uploaded to the database/webserver. Device class numbers below 1000 are reserved for built-in device drivers in the AcquiSuite firmware. Device class numbers for Modbus templates must be 4000 and above. It is important to not use the same device class on more than one template.

The **Number of Data Points** field controls the number of data columns that the template will log in the log file. Note this is not the number of registers to read from the Modbus device. Fill in the total number of data points required. More may be added later, if needed. Note: the data point list menu will not update with new data point rows until you click the save button on this page. It is a good idea to immediately save the advanced page if the number of data points is changed for any reason.

Template Register Blocks:

The Modbus Template requires you to describe what Modbus data queries you wish to perform. This allows you greater control of the way registers are read from the Modbus device. The Modbus **Register Block** tab will display the following screen.

Point#	Modbus Register Address	Offset (hex)	Count	Function
Block 0:	40001	0 0x0000	40	Read Holding Registers
Block 1:	40097	96 0x0060	40	Read Holding Registers
Block 2:	40193	192 0x00C0	40	Read Holding Registers
Block 3:	0	-1 0xFFFFFFFF	0	[not used]
Block 4:	0	-1 0xFFFFFFFF	0	[not used]
Block 5:	0	-1 0xFFFFFFFF	0	[not used]
Block 6:	0	-1 0xFFFFFFFF	0	[not used]
Block 7:	0	-1 0xFFFFFFFF	0	[not used]
Block 8:	0	-1 0xFFFFFFFF	0	[not used]
Block 9:	0	-1 0xFFFFFFFF	0	[not used]

The Template will allow up to 10 Modbus queries to read all the data from the Modbus device. The Modbus protocol supports many different ways to read the data registers from a device. As a practical matter however, there are three common query methods that can be used to read almost all Modbus devices. These include “read coils”, “read input registers” and “read holding registers.”. These three functions can be used to read large blocks, typically 125 to 127 registers, from the Modbus device.

When configuring the register block queries, you must create a set of register queries that will obtain all of the registers required to process the data points you have configured. Data block queries should be constructed to read as many of the needed registers per query. Registers that are read but not used in a data point configuration will be discarded.

When the AcquiSuite makes a Modbus query to a device, the query takes approximately 100ms to 200ms. However, increasing the number of register to a query adds approximately 2ms per register. Therefore it is usually advantageous to read large blocks of registers to gather all the important parts of the data you want, even if many of the registers are unneeded and will be discarded.

To configure a register block query, you must first specify the register starting address. The template editor requires the register address to use the standard Modbus register address notation. For example, the first holding register on most Modbus devices is 40001. Sometimes this will be described in the documentation as an offset. Typically, offset of 0 is equivalent to Modbus register 40001.

Second, you must specify the number of register to read in the query. Typically this must be 125 or less.

Last, you must specify the Modbus query function. This is usually set to “read holding registers”. Note: holding registers are typically 40001+, input registers are 30000+, and coils are at Modbus register address 00001+

Template Data Points:

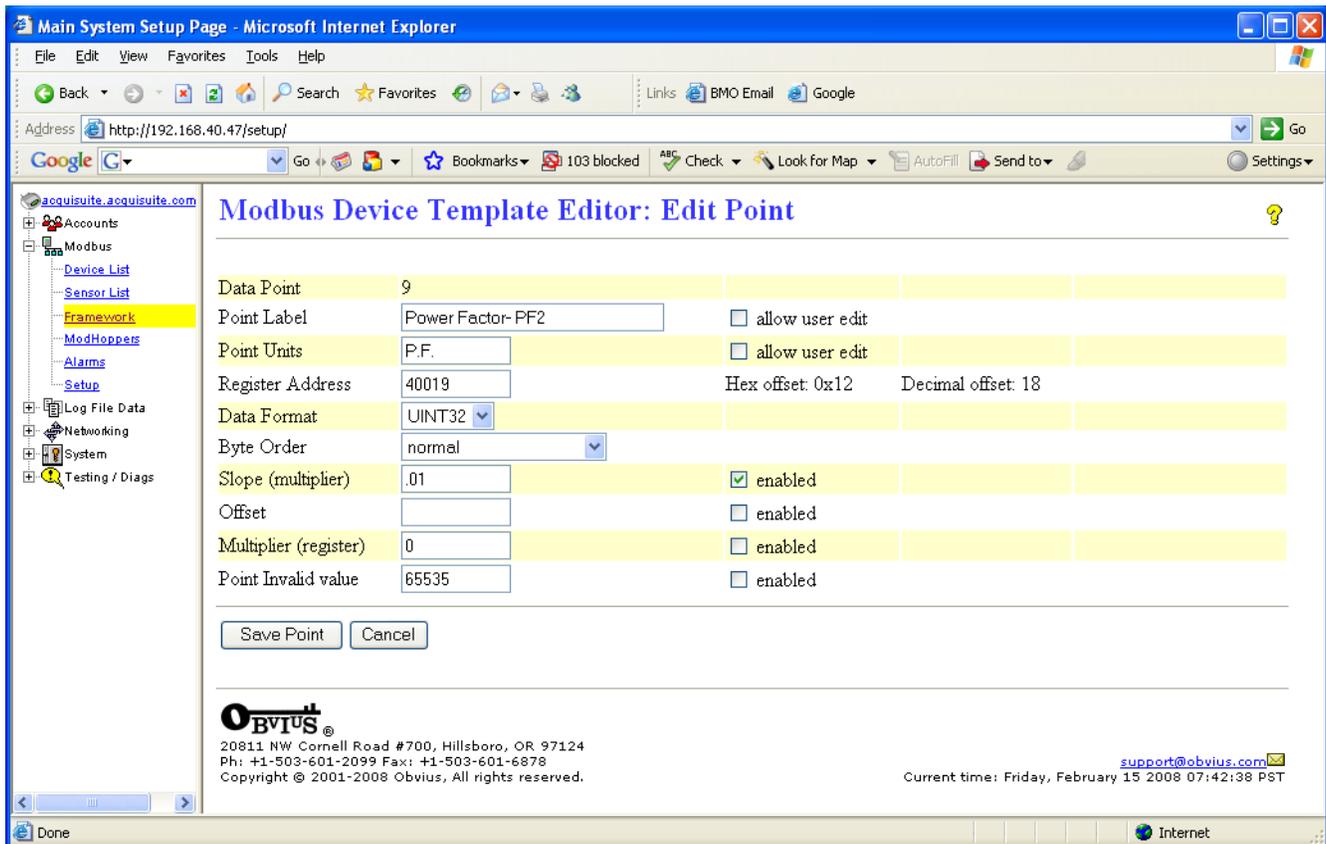
The template point list will show a list of all the data points that have been configured, including a summary of the configuration options for each point. The following screen shows the register list for an example template.

The screenshot shows a web browser window titled "Main System Setup Page - Microsoft Internet Explorer" with the address "http://192.168.40.47/setup/". The page content is titled "Modbus Device Template Editor" and displays a table of data points for the "Template: Circitor CVM96". The table has columns for Point, Name, Units, Register, Type, Byte Order, Slope, Offset, Mult reg, and Invalid val. Each row includes an "Edit Point" button on the right. The data points listed are:

Point	Name	Units	Register	Type	Byte Order	Slope	Offset	Mult reg	Invalid val	Edit Point
0	Phas Point V-1 instantaneous	V	40001	UINT32	none	.1				Edit Point
1	Current- A1	A	40003	UINT32	none	.001				Edit Point
2	Active power- kW1	kW	40005	UINT32	none					Edit Point
3	Reactive power- kvar 1	kvar	40007	UINT32	none	.001				Edit Point
4	Power Factor- PF1	P.F.	40009	UINT32	none	.01				Edit Point
5	Phase Power	V	40011	UINT32	none	.1				Edit Point
6	Current - A2	A	40013	UINT32	none	.001				Edit Point
7	Active Power- kW2	kW	40015	UINT32	none	.001				Edit Point
8	Reactive power- kvar 2	kvar	40017	UINT32	none	.001				Edit Point
9	Power Factor- PF2	P.F.	40019	UINT32	none	.01				Edit Point
10	Phase Voltage- V3	V	40021	UINT32	none	.1				Edit Point
11	Current- A3	A	40023	UINT32	none	.001				Edit Point
12	Active Power- kW3	kW	40025	UINT32	none	.001				Edit Point
13	Reactive Power- kvar3	kvar	40027	UINT32	none	.001				Edit Point
14	Power Factor- PF3	P.F.	40029	UINT32	none	.01				Edit Point

The Data Point list shows the point number (starting at zero), the name, units, register, and several conversion values for each data point that has been configured.

The edit point button on the right side of the screen allows you to make changes to any one data point.



The edit data point page contains all the options for configuring the register conversion for the data value.

Data Point: This is the numerical value of the position of the data point in the log file. The first data point is zero. This can not be edited.

Point Name: Each data point should have a logical name assigned to it. This name should describe the purpose or function of the data point value. Often, it is best to use the same name as the Modbus device manufacturers manual to avoid confusion reading data points at a later time.

Allow Edit: The point name field is followed by a checkbox to allow editing. If this is checked, the name for the data point may be changed later when a device is assigned a template. This allows you to have several Modbus devices with the same template, but rename the specific data point for each Modbus device. This is useful for templates that describe generic inputs such as 0-10v sensors.

Point Units: This is the field label that describes the engineering unit of measure for the data point. e.g. Amps, Volts, etc.

Allow Edit: The point unit field is followed by a checkbox to allow editing. If this is checked, the unit field for the data point may be changed later when a device is assigned a template. This allows you to have several Modbus devices with the same template, but change the unit of measure for the specific data point for each Modbus device. This is useful for templates that describe generic inputs such as 0-10v sensors.

Register Address: The starting Modbus register for the data point. For example, register 40001.

Register Offset [hex/decimal]: this field is not editable. It will be updated after the point edit page has been saved. The value shown will be the decimal and hex offset of the Modbus register configured for this data point. Register 40001 will be at offset at 0. (hex 0x00)

Data Format: Registers may be provided with different types. The following types may be selected.

UINT16:	unsigned 16 bit values (0-65535)
INT16:	signed 16 bit values (-32767 – 32768)
UINT32:	unsigned 32 bit values.
INT32:	signed 32 bit values.
Float:	IEEE 32 bit floating point.
Mod10k (4 registers):	64bit conversion. $A + (B * 10,000) + (C * 100,000,000) + (D * 1,000,000,000,000)$
Mod10k (3 registers):	48bit conversion. $A + (B * 10,000) + (C * 100,000,000)$
Mod10k (2 registers):	32bit conversion. $A + (B * 10,000)$
Mod1k (4 registers):	64bit conversion. $A + (B * 1,000) + (C * 1,000,000) + (D * 1,000,000,000)$
Mod1k (3 registers):	48bit conversion. $A + (B * 1,000) + (C * 1,000,000)$
Mod1k (2 registers):	32bit conversion. $A + (B * 1,000)$
Hex16:	Print 16 bit register in hex format. e.g. 0x1B34. This option is useful for reporting status registers that are bitmap encoded. Note, Hex16 format skips slope/offset scaling. Hex numbers in the log files will have 0x preceding the value.

Byte Order: Byte swapping may be selected. Consult your Modbus documentation to verify the byte order. Typical order is Most significant to least significant. Use “normal” unless otherwise required. Note, select byte ordering that has the appropriate number of byte swaps based on your data format.

Slope (multiplier): This value will be multiplied against the register value. For example, if the register provides AMPS x 10, the slope should be set to 0.1 to convert the register to the final value. In this example, a register value of 1234 will become 123.4 in the log file data.

Offset: This value will be added to the register.

Multiplier (register): If the meter provides the multiplier as another register in the meter's register list, you may specify it here. The value of this register will be multiplied against the data point before storing in the log file.

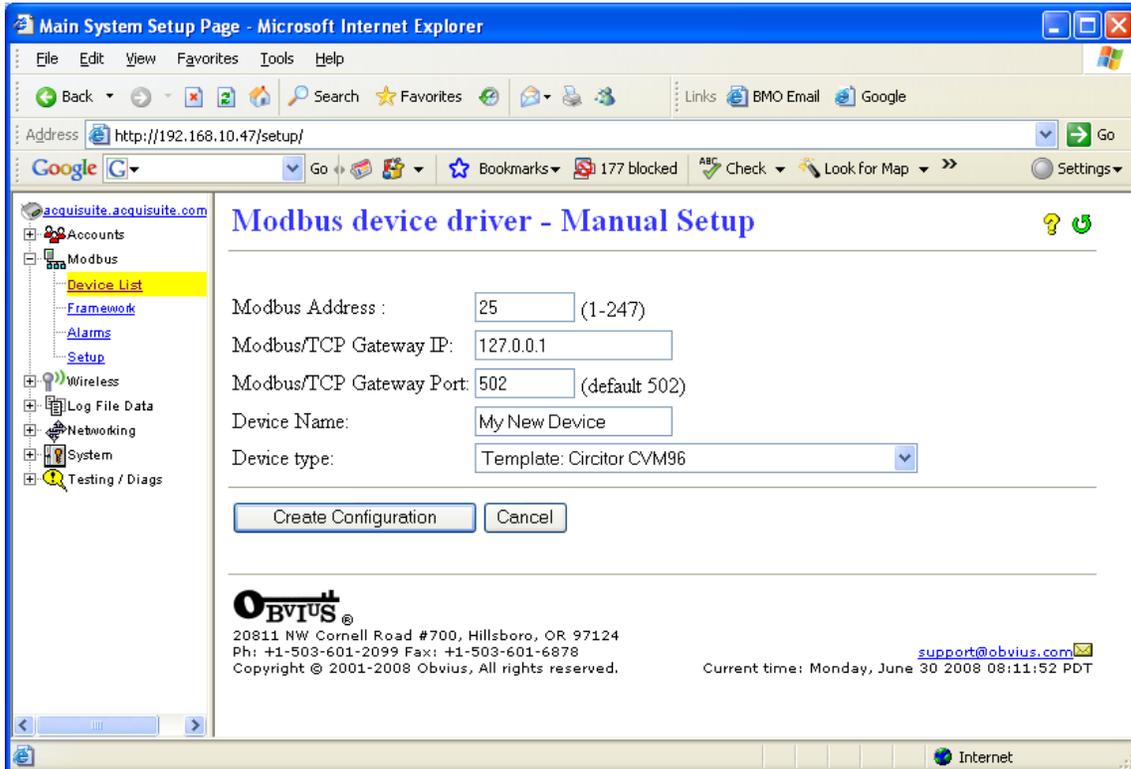
Invalid value: If the meter can report invalid data values (off scale high) the register value may be specified here. For example, many power meters use 65535 to indicate off-scale-high when reporting power values.

Enable: This check box enables or disables the data value conversion feature to the left of the checkbox. Enable only the functions that you wish to use in the conversion.

Assigning Templates to devices:

From the device list, select the Modbus address number for the device you wish to configure. If the device is not yet present, you may also click the “add” button at the bottom of the list. You will be presented with a list of all templates on the system. Select the correct template, and click save.

When configuring a new Modbus device you must assign a name for the device in order for the AcquiSuite to begin logging data.



Configure the following options:

Modbus Address: should be the Modbus address of the device that you wish to configure. If you click on the specific device number from the device list, this field will not be editable.

Modbus/TCP Gateway IP, Port: You may enter the IP address and port number to connect to a Modbus/TCP gateway or direct connect to a remote device via the LAN. For Modbus devices that are direct connected to the AcquiSuite RS485 port, use the local host ip address “127.0.0.1” and port “502” as shown in the example above. Note: up to 32 remote LAN based Modbus/TCP devices may be configured.

Device Name: Enter a logical name for the device to identify it in the device list. This name should include the location and purpose of the Modbus device.

Device Type: Use this picklist option to select either a built in driver, or a template that you have created.

Click the “Create Configuration” button to finalize the options. Return to the device list to view the status of the device you have assigned to a template.

Copy Templates from one AcquiSuite to another:

Template files are stored in a subdirectory of the AcquiSuite configuration area. Template files may be transferred between AcquiSuite systems using FTP to move the files from one AcquiSuite to a desktop or laptop computer, and then out to another AcquiSuite system.

First, FTP to the AcquiSuite using a command line FTP tool or a windows based tool such as Internet Explorer or SmartFTP.

When prompted to login, supply “root” for the login name, and the admin password.

Change to binary mode before transferring any files.

Change directories to `/etc/sysconfig/modbus/`

locate all files ending in the “.mbt” file extension. The first part of the file name indicates the Modbus device type that was configured.

You may use “get” and “put” to download or upload the mbt files.

Using Microsoft Internet Explorer:

Microsoft Internet Explorer can be used to FTP files to and from the AcquiSuite. To use MSIE, enter the following URL into the address bar:

```
ftp://root@192.168.10.50/./mnt/main/sysconfig/modbus
```

Enter your administrator password when prompted.

In the window provided you can copy and paste the “.mbt” files to transfer templates to the AcquiSuite.

For more details on using MSIE to transfer files to the AcquiSuite, please read [TN20-AcquiSuite-FTP-MSIE.pdf](#) available on the obvius.com website in the documentation/technotes section.