

Chipkin™ Chariot Quick Start Guide



Use this guide to quickly configure the Chariot to poll for BACnet IP data points and serve them as Modbus TCP registers, or poll for Modbus TCP registers and serve them as BACnet IP data points. There will be a section dedicated to each.

VERSION: 0.03

PRODUCT NUMBER: Chariot

DOCUMENT REVISION: 2

Table of Contents

1. Configuration – BACnet IP to Modbus TCP	2
1.1 Configuring the BACnet IP Read Tasks	2
1.2 Configuring the Modbus TCP Server Task.....	5
1.3 Saving and Testing the Configuration	8
2. Configuration – Modbus TCP to BACnet IP	10
2.1 Configuring the Modbus TCP Read Tasks.....	10
2.2 Configuring the BACnet IP Server Tasks.....	13
2.3 Saving and Testing the Configuration	16

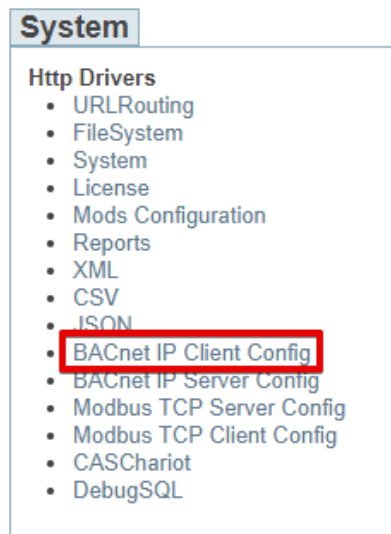
1. Configuration – BACnet IP to Modbus TCP

Follow these steps to setup tasks to read BACnet IP data points and serve them as Modbus TCP registers.

1.1 Configuring the BACnet IP Read Tasks

First, configure tasks to poll for the BACnet IP data.

1. Click on the BACnet IP Client Config link from the Start page.



2. Next, click on the “Insert” link to add BACnet IP tasks.

System

Http Drivers

- URLRouting
- FileSystem
- System
- License
- Mods Configuration
- Reports
- XML
- CSV
- JSON
- BACnet IP Client Config
- BACnet IP Server Config
- Modbus TCP Server Config
- Modbus TCP Client Config
- CASChariot
- DebugSQL

BACnet IP Client

BACnet IP Devices

No BACnet IP devices have been configured yet.

BACnetIP tasks

Actions Insert Download as CSV
 Error: Table is empty

3. Fill out the form to add a BACnet IP data point to poll

BACnet IP Client

Insert new record in to 'BACnetIP_tasks'

Name	Value	Description
Name	<input type="text"/>	The name of the task. This can be anything and is used to help identify the task.
Network	<input type="text"/>	The network that the BACnet device is on
Device ID	<input type="text"/>	The ID of the BACnet device.
IP Address	<input type="text" value="0.0.0.0"/>	The IP Address of the BACnet device
Port	<input type="text" value="47808"/>	The IP Port for the BACnet device. Default: 47808
SADR	<input type="text"/>	Optional: The Source Address of the BACnet device. Must be in Hex. Example: 01E843
Object Type	<input type="text" value="Analog Input"/>	The object type of the BACnet object being polled. Example: Analog Output
Object ID	<input type="text"/>	The ID of the BACnet object being polled.
Property	<input type="text" value="presentValue"/>	The units of the property of the BACnet object being polled. Default: no_units
Service	<input type="text" value="Read Property"/>	Read Property: Sends a ReadProperty message for the configured object and property. Read Property Multiple: Combines other ReadPropertyMultiple tasks into one ReadPropertyMultiple message.
Units	<input type="text" value="noUnits"/>	The property of the BACnet object being polled. Default: present_value
Scan	<input type="text" value="30"/>	How often to poll for this object in seconds.

insert

The form as the following fields:

- **Name:** The name of the task. This can be any value and is only used to help identify what this task is doing.
- **Network:** The BACnet network that the BACnet device is on. If the device is on the local network, use 0.
- **Device ID:** The BACnet ID of the device.
- **IP Address:** The IP Address of the BACnet device.
- **Port:** The IP Port for communicating with the BACnet device. By default, this is 47808.
- **SADR:** The BACnet Source Address of the BACnet Device. The parameter is optional and is only used if the Network is non-zero. This value is a Hex-String.
- **Object Type:** The object type of the BACnet object being polled.
- **Object ID:** The BACnet ID of the object being polled.
- **Property:** The BACnet property of the BACnet object to poll. This is the actual data point that will be polled. By default, this is present_value.
- **Service:** The BACnet service to use to poll the values. Options are: ReadProperty or ReadPropertyMultiple.
- **Units:** The units of the data point being polled.
- **Scan:** How often to poll for the data point in seconds.

4. After filling out the form, click the “insert” button to create the task:

BACnet IP Client

Insert new record in to 'BACnetIP_tasks'

Name	Value	Description
Name	ReadTemp	The name of the task. This can be anything and is used to help identify the task.
Network	0	The network that the BACnet device is on
Device ID	389999	The ID of the BACnet device.
IP Address	192.168.1.26	The IP Address of the BACnet device
Port	47808	The IP Port for the BACnet device. Default: 47808
SADR		Optional: The Source Address of the BACnet device. Must be in Hex. Example: 01E843
Object Type	Analog Input	The object type of the BACnet object being polled. Example: Analog Output
Object ID	0	The ID of the BACnet object being polled.
Property	presentValue	The units of the property of the BACnet object being polled. Default: no_units
Service	Read Property	Read Property: Sends a ReadProperty message for the configured object and property. Read Property Multiple: Combines other ReadPropertyMultiple tasks into one ReadPropertyMultiple message.
Units	degreesCelsius	The property of the BACnet object being polled. Default: present_value
Scan	30	How often to poll for this object in seconds.



5. If successfully added, the task and device will be displayed.

BACnet IP Client

Success, Record #1 has been added

BACnet IP Devices

ID	IP Address	Device Instance	Network	SADR	Port
1	192.168.1.26	389999	0		47808

BACnetIP_tasks

Actions: Insert | Download as CSV
 Displaying 30 records from 0-1 of a total 1

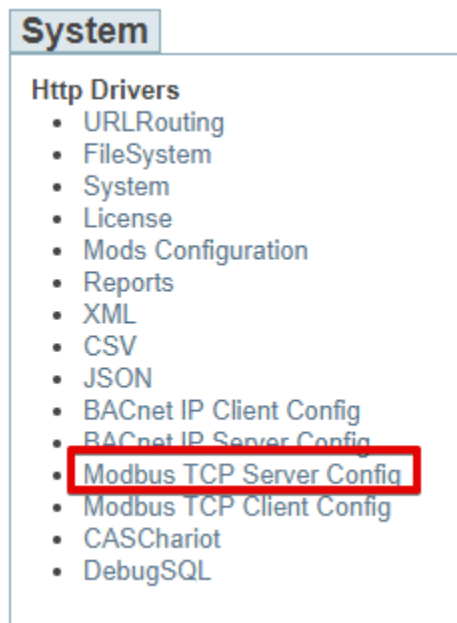
action	id	data_offset	device_id	name	object_id	object_type	object_units	priority	property	scan	service
Edit Delete	1	1	1	ReadTemp	0	analogInput	degreesCelsius		presentValue	30	readProperty

6. Repeat steps 2-4 to add additional points. **NOTE:** The Chariot supports a maximum of 5 data points.

1.2 Configuring the Modbus TCP Server Task

Once all the BACnet IP tasks have been configured, the next step is to configure the Modbus TCP server task that will serve the BACnet data as Modbus registers.

1. Click on the Modbus TCP Server Config link



2. Update the TCP Server Configuration settings and Click the Save Device Settings button

Modbus TCP Server Configuration

Setting	Value	
Port	502	The TCP port through which the modbus data is served
Slave ID	1	The Modbus Slave ID of this device

Save Device Settings

1. Update form

2. Click "Save Device Settings" to save

ModbusTCP_tasks

Actions: Insert| Download as CSV
 Error: Table is empty

The fields to configure are:

- **Port:** The Modbus TCP port. By default, this is 502.
- **Slave ID:** The Slave ID that represents the Chariot as a Modbus Server device.

3. Click the "Insert" link to add a Modbus TCP server task

Modbus TCP Server Configuration

Setting	Value	
Port	502	The TCP port through which the modbus data is served
Slave ID	1	The Modbus Slave ID of this device

Save Device Settings

ModbusTCP_tasks

Actions: **Insert**| Download as CSV
 Error: Table is empty

4. Fill out the form to create the Modbus TCP server task
Insert new record in to 'ModbusTCP_tasks'

Name	Value	Description										
Name	<input type="text"/>	The name of the task. This can be anything and is used to help identify the task.										
Modbus Device ID	<input type="text"/>	This is the Modbus Slave device ID. 0-254										
Modbus Address	<input type="text"/>	<table border="1"> <thead> <tr> <th>Function</th> <th>Address</th> </tr> </thead> <tbody> <tr> <td>0x01 Read Coils</td> <td>0xxx1</td> </tr> <tr> <td>0x02 Read Discrete Inputs</td> <td>1xxx1</td> </tr> <tr> <td>0x03 Read Holding Registers</td> <td>4xxx1</td> </tr> <tr> <td>0x04 Read Input Registers</td> <td>3xxx1</td> </tr> </tbody> </table> <p>A value of Zero is invalid.</p>	Function	Address	0x01 Read Coils	0xxx1	0x02 Read Discrete Inputs	1xxx1	0x03 Read Holding Registers	4xxx1	0x04 Read Input Registers	3xxx1
Function	Address											
0x01 Read Coils	0xxx1											
0x02 Read Discrete Inputs	1xxx1											
0x03 Read Holding Registers	4xxx1											
0x04 Read Input Registers	3xxx1											
Number Of Registers	<input type="text"/>	The quantity of registers that this task can serve in one request. Must be smaller then the database length - data offset										
Data Offset	<input type="text"/>	This is a pointer to the data table. Should be non zero and less then the length of the database. You can resize the database from the System actions page.										

insert

The form has the following fields:

- **Name:** The name of the task. This can be any value and is used to identify the task.
- **Modbus Device ID:** The Slave ID configured in the previous step.
- **Modbus Address:** The starting address of the Modbus Registers to serve. This will typically be 40001.
- **Number Of Registers:** The number of data points to serve. If there are 5 BACnet IP data points to serve, then this value should be 5.
- **Data Offset:** The data offset from the physical database of the data to serve. This value should always be 1.

5. After filling out the form, click the “insert” button to add the task:

Insert new record in to 'ModbusTCP_tasks'

Name	Value	Description										
Name	ServeBACnetData	The name of the task. This can be anything and is used to help identify the task.										
Modbus Device ID	1	This is the Modbus Slave device ID. 0-254										
Modbus Address	40001	<table border="1"> <thead> <tr> <th>Function</th> <th>Address</th> </tr> </thead> <tbody> <tr> <td>0x01 Read Coils</td> <td>0xxx1</td> </tr> <tr> <td>0x02 Read Discrete Inputs</td> <td>1xxx1</td> </tr> <tr> <td>0x03 Read Holding Registers</td> <td>4xxx1</td> </tr> <tr> <td>0x04 Read Input Registers</td> <td>3xxx1</td> </tr> </tbody> </table> <p>A value of Zero is invalid.</p>	Function	Address	0x01 Read Coils	0xxx1	0x02 Read Discrete Inputs	1xxx1	0x03 Read Holding Registers	4xxx1	0x04 Read Input Registers	3xxx1
Function	Address											
0x01 Read Coils	0xxx1											
0x02 Read Discrete Inputs	1xxx1											
0x03 Read Holding Registers	4xxx1											
0x04 Read Input Registers	3xxx1											
Number Of Registers	5	The quantity of registers that this task can serve in one request. Must be smaller then the database length - data offset										
Data Offset	1	This is a pointer to the data table. Should be non zero and less then the length of the database. You can resize the database from the System actions page.										

insert

1.3 Saving and Testing the Configuration

Once the configuration is completed, it needs to be saved and the chariot needs to be restarted before we can test to make sure data is being read and served.

1. Save the Configuration.

First, return to the main system page by typing in the follow URL into a web browser:

- <http://{ipAddress}/bin/system/> where {ipAddress} is the IP Address of the gateway.

From this page, under the System actions, first click the “Save Database” link to save all changes.

System Actions

This page is for system wide actions that effect all the drivers.

System

- [Reboot System](#) - Use this link to send a reboot REST request to the system.
- [Save Database](#) - Use this link to send a save database REST request to the system.
- [Delete Database](#) - Use this link to send a delete database REST request to the system.
- [Generate Configuration File](#) - Use this link to generate a configuration file.

Figure 1.3-1 - Save Database Link

Click “Ok” when prompted and check that the response status is OK.

2. Reboot the Application.

Click the back button in your web browser and then click the “Reboot System” link:

System Actions

This page is for system wide actions that effect all the drivers.

System

- **Reboot System** - Use this link to send a reboot REST request to the system.
- Save Database - Use this link to send a save database REST request to the system.
- Delete Database - Use this link to send a delete database REST request to the system.
- Generate Configuration File - Use this link to generate a configuration file.

Figure 1.3-2 - Reboot System Link

Click “Ok” when prompted and you will see the following screen with a timer counting up:

Restarting device

The device is restarting, this may take a few minutes.

Status: Checking ... 2

Figure 1.3-3 - Reboot System Count

The system page will refresh once the application has been properly rebooted.

3. Use the CAS Modbus Scanner to test the Modbus registers. Refer to the [Modbus for Field Technicians](#) booklet for more information about Modbus. For instructions on using the CAS Modbus Scanner, review section 19 on page 61 of the Modbus for Field Technicians.

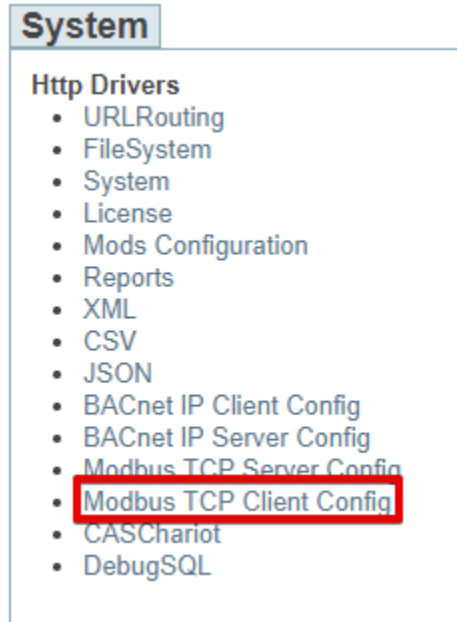
2. Configuration – Modbus TCP to BACnet IP

Follow these steps to setup tasks to read Modbus TCP registers and serve them as BACnet IP data points and serve them.

2.1 Configuring the Modbus TCP Read Tasks

First, configure tasks to poll for the Modbus TCP Data.

1. Click on the Modbus TCP Client Config link from the Start Page



2. Next, click on the “insert” link to begin adding a Modbus TCP client task

Modbus TCP Devices

No Modbus TCP devices have been configured yet.

ModbusTCPClient_tasks

Actions [Insert](#) Download as CSV

Error: Table is empty

3. Fill out the form to create a Modbus TCP Client Task

Insert new record in to 'ModbusTCPClient_tasks'

Name	Value	Description										
Name	<input type="text"/>	The name of the task. This can be anything and is used to help identify the task.										
IP address	<input type="text"/>	The IP address of the slave device.										
Port	502	The TCP Port of the slave device. Default: 502										
Slave ID	<input type="text"/>	The address of the Modbus slave device. Valid addresses are in the range 1..247.										
Modbus Address	<input type="text"/>	<table border="1"> <thead> <tr> <th>Function</th> <th>Address</th> </tr> </thead> <tbody> <tr> <td>(0x01) Read Coils</td> <td>0-10000</td> </tr> <tr> <td>(0x02) Read Discrete Inputs</td> <td>10001-30000</td> </tr> <tr> <td>(0x03) Read Holding Registers</td> <td>30001-40000</td> </tr> <tr> <td>(0x04) Read Input Registers</td> <td>40001+</td> </tr> </tbody> </table>	Function	Address	(0x01) Read Coils	0-10000	(0x02) Read Discrete Inputs	10001-30000	(0x03) Read Holding Registers	30001-40000	(0x04) Read Input Registers	40001+
Function	Address											
(0x01) Read Coils	0-10000											
(0x02) Read Discrete Inputs	10001-30000											
(0x03) Read Holding Registers	30001-40000											
(0x04) Read Input Registers	40001+											
Number Of Registers	1	The number of elements that will be requested in this task.										
Scan	30	How often in seconds to run this task. (Seconds)										

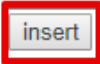
The form has the following fields:

- **Name:** The name of the task. This can be any value and is only used to identify the task
- **IP Address:** The IP Address of the Modbus slave device.
- **Port:** The TCP port of the Modbus slave device for Modbus TCP communication. By default, this is 502.
- **Slave ID:** The ID of the Modbus Slave device.
- **Modbus Address:** The starting address of the Modbus registers to read.
- **Number Of Registers:** The number of Modbus registers to read.
- **Scan:** How often to poll for the data in seconds.

4. After filling out the form, click the “insert” button to add the task

Insert new record in to 'ModbusTCPClient_tasks'

Name	Value	Description										
Name	<input type="text" value="Read40001"/>	The name of the task. This can be anything and is used to help identify the task.										
IP address	<input type="text" value="192.168.1.26"/>	The IP address of the slave device.										
Port	<input type="text" value="502"/>	The TCP Port of the slave device. Default: 502										
Slave ID	<input type="text" value="1"/>	The address of the Modbus slave device. Valid addresses are in the range 1..247.										
Modbus Address	<input type="text" value="40001"/>	<table border="1"> <thead> <tr> <th>Function</th> <th>Address</th> </tr> </thead> <tbody> <tr> <td>(0x01) Read Coils</td> <td>0-10000</td> </tr> <tr> <td>(0x02) Read Discrete Inputs</td> <td>10001-30000</td> </tr> <tr> <td>(0x03) Read Holding Registers</td> <td>30001-40000</td> </tr> <tr> <td>(0x04) Read Input Registers</td> <td>40001+</td> </tr> </tbody> </table>	Function	Address	(0x01) Read Coils	0-10000	(0x02) Read Discrete Inputs	10001-30000	(0x03) Read Holding Registers	30001-40000	(0x04) Read Input Registers	40001+
Function	Address											
(0x01) Read Coils	0-10000											
(0x02) Read Discrete Inputs	10001-30000											
(0x03) Read Holding Registers	30001-40000											
(0x04) Read Input Registers	40001+											
Number Of Registers	<input type="text" value="1"/>	The number of elements that will be requested in this task.										
Scan	<input type="text" value="30"/>	How often in seconds to run this task. (Seconds)										



5. If successfully added, the task will be displayed:

Success, Record #2 has been deleted

Modbus TCP Devices

ID	Slave ID	IP address	Port
1	1	192.168.1.26	502

ModbusTCPClient_tasks

Actions: Insert | Download as CSV

Displaying 30 records from 0-1 of a total 1

Action	ID	Number Of Registers	Device ID	Function	Name	Modbus Offset	Scan
Edit Delete	1	1	1	03 Read Holding Registers (4xxxx)	Read40001	0	30

6. Repeat steps 2-4 to add additional tasks if required.

Note: If all the data points being read are in consecutive registers, then you will only need to configure one task. For example: to read registers 40001 to 40005 you can fill out the form as follows:

Insert new record in to 'ModbusTCPClient_tasks'

Name	Value	Description										
Name	Read40001-40005	The name of the task. This can be anything and is used to help identify the task.										
IP address	192.168.1.26	The IP address of the slave device.										
Port	502	The TCP Port of the slave device. Default: 502										
Slave ID	1	The address of the Modbus slave device. Valid addresses are in the range 1..247.										
Modbus Address	40001	<table border="1"> <thead> <tr> <th>Function</th> <th>Address</th> </tr> </thead> <tbody> <tr> <td>(0x01) Read Coils</td> <td>0-10000</td> </tr> <tr> <td>(0x02) Read Discrete Inputs</td> <td>10001-30000</td> </tr> <tr> <td>(0x03) Read Holding Registers</td> <td>30001-40000</td> </tr> <tr> <td>(0x04) Read Input Registers</td> <td>40001+</td> </tr> </tbody> </table>	Function	Address	(0x01) Read Coils	0-10000	(0x02) Read Discrete Inputs	10001-30000	(0x03) Read Holding Registers	30001-40000	(0x04) Read Input Registers	40001+
Function	Address											
(0x01) Read Coils	0-10000											
(0x02) Read Discrete Inputs	10001-30000											
(0x03) Read Holding Registers	30001-40000											
(0x04) Read Input Registers	40001+											
Number Of Registers	5	The number of elements that will be requested in this task.										
Scan	30	How often in seconds to run this task. (Seconds)										

insert

2.2 Configuring the BACnet IP Server Tasks

Once Modbus TCP client tasks have been configured, the next step is to configure the BACnet IP server tasks that will serve the Modbus data as BACnet objects.

1. Click on the BACnet IP Server Config link from the Start page

System

Http Drivers

- URLRouting
- FileSystem
- System
- License
- Mods Configuration
- Reports
- XML
- CSV
- JSON
- BACnet IP Client Config
- **BACnet IP Server Config**
- Modbus TCP Server Config
- Modbus TCP Client Config
- CASChariot
- DebugSQL

2. Fill out the BACnet IP Device Configuration and Protocol Services Supported Settings

BACnet IP Server

BACnet IP Device Configuration

Setting	Value	Description
Port	47808 <input type="text"/>	The port for the BACnet IP connection. Default: 47808
Device Instance	389999 <input type="text"/>	The Device instance for this BACnet server object
Device Name	Device (389001) <input type="text"/>	The Device name this BACnet server object

Protocol Services Supported

Show

The BACnet IP Device Configuration has the following fields:

- **Port:** The BACnet IP port to use. By default, this is 47808
- **Device Instance:** The BACnet device instance number to represent this device. This must be unique on the BACnet intra-network.
- **Device Name:** The name for this BACnet device.

The Protocol Services Supported is a set of options that enable different BACnet functionality. At a minimum, readProperty and who_Is should be selected.

3. After filling out the form, click the Save Device Settings button to save the settings

BACnet IP Server

BACnet IP Device Configuration

Setting	Value	Description
Port	47808 <input type="text"/>	The port for the BACnet IP connection. Default: 47808
Device Instance	12345 <input type="text"/>	The Device instance for this BACnet server object
Device Name	Test Device <input type="text"/>	The Device name this BACnet server object

Protocol Services Supported

Show

4. Next add a BACnet IP Server task by clicking on the “Insert” link:

BACnetIP tasks

Actions: insert Download as CSV
 Displaying 30 records from 0-1 of a total 1
 No records

5. Fill out the form to create the BACnet IP Server Task

Insert new record in to 'BACnetIP_tasks'

Name	Value	Description
Name	<input type="text"/>	The name of the BACnet object. This name will be displayed to the BACnet clients polling this device
Object Type	Analog Input ▾	The type of the BACnet object. Default: Analog Input
Object ID	<input type="text"/>	The instance number of the BACnet object. The instance number should be unique amount all other objects of the same type.
Property	presentValue ▾	The property that this data point is serving. Normally this will be present_value, Default: present_value
Units	noUnits ▾	This is the units of the present_value property. Only used when Property is present_value. Default: no_units
Data Point	1 ▾	This is a pointer to the data table. Should be non zero and less then the length of the database. You can resize the database from the System actions page.

The form has the following fields:

- **Name:** The name of the BACnet Object. This name will be displayed to the BACnet clients polling this device.
- **Object Type:** The type of BACnet object to represent the data point. For example: Temperature would be Analog Input, while Occupancy could be a Binary Input.
- **Object ID:** The instance number of the BACnet object. This number should be unique among all other objects of the same type.
- **Property:** The property that this data point is serving. This should usually be present_value.
- **Units:** The units of the data point.
- **Data Point:** The offset in the physical data base that stores the value to serve. For the Chariot, this will have a value from 1 to 5. For example, if registers 40001-40005 were configured on the Modbus TCP Client configuration, then data point 1 has the value from register 40001, etc.

6. After filling out the form, click the “insert” button

Insert new record in to 'BACnetIP_tasks'

Name	Value	Description
Name	<input type="text" value="Temperature(40001)"/>	The name of the BACnet object. This name will be displayed to the BACnet clients polling this device
Object Type	<input type="text" value="Analog Input"/>	The type of the BACnet object. Default: Analog Input
Object ID	<input type="text" value="1"/>	The instance number of the BACnet object. The instance number should be unique amount all other objects of the same type.
Property	<input type="text" value="presentValue"/>	The property that this data point is serving. Normally this will be present_value, Default: present_value
Units	<input type="text" value="degreesCelsius"/>	This is the units of the present_value property. Only used when Property is present_value. Default: no_units
Data Point	<input type="text" value="1"/>	This is a pointer to the data table. Should be non zero and less then the length of the database. You can resize the database from the System actions page.

7. Repeat steps 4-6 to add additional BACnet IP server tasks.

2.3 Saving and Testing the Configuration

Once the configuration is completed, it needs to be saved and the Chariot needs to be restarted before we can test to make sure data is being read and served.

1. Save the Configuration.

First, return to the main system page by typing in the follow URL into a web browser:

- <http://{ipAddress}/bin/system/> where {ipAddress} is the IP Address of the gateway.

From this page, under the System actions, first click the “Save Database” link to save all changes.

System Actions

This page is for system wide actions that effect all the drivers.

System

- [Reboot System](#) - Use this link to send a reboot REST request to the system.
- [Save Database](#) - Use this link to send a save database REST request to the system.
- [Delete Database](#) - Use this link to send a delete database REST request to the system.
- [Generate Configuration File](#) - Use this link to generate a configuration file.

Figure 2.3-1 - Save Database Link

Click “Ok” when prompted and check that the response status is OK.

2. Reboot the Application.

Click the back button in your web browser and then click the “Reboot System” link:

System Actions

This page is for system wide actions that effect all the drivers.

System

- **Reboot System** - Use this link to send a reboot REST request to the system.
- Save Database - Use this link to send a save database REST request to the system.
- Delete Database - Use this link to send a delete database REST request to the system.
- Generate Configuration File - Use this link to generate a configuration file.

Figure 2.3-2 - Reboot System Link

Click “Ok” when prompted and you will see the following screen with a timer counting up:

Restarting device

The device is restarting, this may take a few minutes.

Status: Checking ... 2

Figure 2.3-3 - Reboot System Count

The system page will refresh once the application has been properly rebooted.

3. Use the CAS BACnet Explorer to test the BACnet Objects. Refer to the [BACnet For Field Technicians](#) booklet for more information about BACnet. For instructions on using the CAS BACnet Explorer, review <https://store.chipkin.com/articles/cas-bacnet-explorer-manual>.