

Ethernet/IP configuration instructions for VAMP devices

1 Introduction

1.1 Document overview

The purpose of this document is to describe the configuration and testing of the Ethernet/IP protocol in VAMP 50, VAMP 200 and VAMP 300 series devices.

In this document, the word “device” refers to VAMP protection relays, VAMP measuring units and similar types of equipment.

In order to configure a VAMP device, the VAMPSET relay setting and configuration tool is needed. VAMPSET can be downloaded from the website (<http://www.schneider-electric.com/products/ww/en/2300-ied-user-software/2320-vamp-user-software/62050-vamp-software/?xtmc=vamp&xtcr=2>).

The configurator of a device must be logged in with *Configurator* access level in order to be able to make any changes to protocol settings. This is done by entering the appropriate password when connecting to a device with VAMPSET. The default password for configurator access level is “2”.

This document assumes that the reader has some previous knowledge of the Ethernet/IP protocol.

Note: This document applies fully only to firmware versions v.10.134 and newer.

1.2 References

- [1] Application Note DeviceNet and EtherNet/IP Data Model, “DeviceNet and EtherNet/IP data model in VAMP devices”, 30.10.2014, V1.1

2 Ethernet/IP

This section will give a brief overview of Ethernet/IP in general and of its implementation in VAMP devices.

2.1 Ethernet/IP overview

Ethernet/IP is an adaptation of the Common Industrial Protocol (CIP), which is an open standard, designed for vendor independent device interoperability that is strictly object oriented protocol.

CIP can use different transport layers. In the case of Ethernet/IP, Ethernet (IEEE 802.3 and the TCP/IP protocol suite) is used.

The “IP” in “Ethernet/IP” refers to “Industrial Protocol”

2.1.1 Messaging

Ethernet/IP supports two modes of messaging, unconnected and connected messaging.

Unconnected messaging refers to peer-to-peer communication, where opening and closing of connections is allowed via unconnected messaging. This is handled by the Unconnected Message Manager (UCMM). Messages are sent over TCP/IP.

Connected messaging, on the other hand, is dedicated to a particular purpose, such as frequent explicit message transactions or real-time I/O data transfers. Connection resources are reserved and configured using communications services available via the UCMM. Messages are sent over TCP/IP and UDP.

Ethernet/IP specifies a special encapsulation protocol to carry CIP messages over TCP/IP and UDP.

There are two types of connections, explicit and implicit.

Explicit connections refer to request-response connections which are general purpose connections. Explicit connections use TCP/IP and use either unconnected messaging via UCMM (one-time request/response) or Class 3 connections (cyclic request /response).

In *implicit* connections, only application data is contained within the messages. Implicit data may be polled, cyclic or COS (Change of State) messages. Implicit connections are either point-to-point (unicast) or one-to-many (multicast) connections. Implicit connections use UDP/IP.

2.1.2 Devices

There are two classes of devices in a CIP network, adapters and scanners.

Adapters are *targets* of real-time I/O data connection. Adapters cannot send or receive real-time data unless requested to do so by a scanner device. Adapters can exchange data using explicit messages with any class of devices but cannot originate a connection.

Scanners are *originators* of I/O data connection requests and originators or targets of explicit connection requests.

2.1.3 Objects

Objects in CIP (and thereby Ethernet/IP) are defined by:

- A description – a description of an object being specified
- A class code (Class ID) – a hexadecimal identifier assigned to each CIP object.
- Attributes – data elements associated with the object.
- Common services – a list of the common services defined for the object.
- Object-specific Services – the full specifications of any services unique to the object.
- Connections – connections supported by the object.
- Behaviour – the relationship between attribute values and services.

The object model of CIP is shown in Figure 2.1.3-1. A description of the objects is given in Table 2.1.3-1.

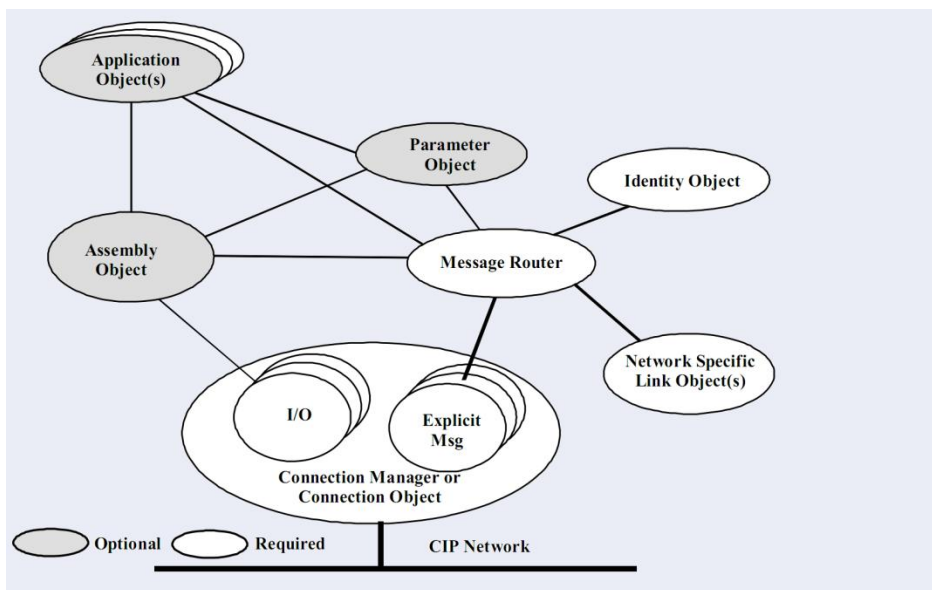


Figure 2.1.3-1: The CIP object model.

The CIP specification also includes an Object Library, which is a set of standardized objects that covers protocol objects such as the Identity Object, Assembly Object, etc. as well as Application Objects. The Object Library covers basic industrial automation building blocks and some more complex devices, including Digital Input, Digital Output, Analog Input, Analog Output, Position Sensor, Position Controller, AC/DC Drive, Overload, Softstart and Motion Axis.

Table 2.1.3-1: CIP object description.

Object	Description
Connection Object	The CIP Communication Object manages and provides the run-time exchange of messages
Message Router	The Message Router Object provides a messaging connection point through which a Client may address a service to any object class or instance residing in the physical device. It routes explicit messages over requested paths.
Assembly Object	The Assembly Object binds attributes of multiple objects, which in allows data to or from each of these objects to be sent or received over a single connection (like a data set). Assembly objects can be used to group input data (producing instance of an Assembly Object – information transmitted to the network) or output data (consuming instance of an Assembly Object – information received from the network). I/O connections are established between Assembly Object instances of the devices – between inputs and outputs. Assembly object instances are accessible via explicit messaging.
Identity Object	Provides device identification and general information about the device, such as vendor identifier, product code, name, status, etc. The Identity Object shall be present in all CIP products.

Application Objects should be based on the standard objects from the Object Library if possible, if not, vendors can define their own, private (vendor specific) Application Objects. CIP specifies the Class ID ranges for that purpose, see Table 2.1.3-2.

Table 2.1.3-2: CIP Class ID ranges.

Range (hexadecimal)	Meaning	Quantity
0x00...0x63	Open	100
0x64...0xC7	Vendor specific	100
0xC8...0xEF	Reserved by CIP for future use	40
0xF0...0x2FF	Open	528
0x300...0x4FF	Vendor specific	512
0x500...0xFFFF	Reserved by CIP for future use	64256

2.1.4 Device profile

The series of application objects for a particular device type is known as the device profile. A large number of profiles for many device types have been defined. An example of a device profile is shown in Table 2.2.1-1.

2.1.5 Electronic Data Sheet (EDS)

Ethernet/IP vendors are required to provide some type of documentation specifying how their device is configured. The document may be a set of printed instructions or some electronic file. An electronic listing of the attributes that configure an Ethernet/IP device is usually provided by a vendor.

An Electronic Data Sheet (EDS) is a textual description of a device profile. An EDS specifies the accessible parameters and the content of Assemblies. The format of an EDS (file) is similar to Windows .INI files to ensure machine readability. EDS file mainly contains:

- Device’s identity information – main configuration parameters described in section 3
- Parameters list – all of the attributes available in the data model; see section 5
- Assemblies info – current configuration of producing and consuming assemblies.

2.2 Ethernet/IP in VAMP devices

Ethernet/IP protocol is available on VAMP relay with an optional embedded Ethernet card. The protocol can be used to read / write data from the relay using request / response communication and via cyclic IO messages transporting data assigned to assemblies (sets of data). VAMP relay with the EtherNet/IP protocol selected on the Ethernet port serves as an adapter which means that it is not able to initiate communication with other devices in the network

2.2.1 Objects and Messaging

The Ethernet/IP implementation on VAMP devices supports all required standard objects with their required attributes. There are also total of 10 application objects from which 8 are private. A list of VAMP device’s objects and their classes is shown in Table 2.2.1-1.

Table 2.2.1-1: Device profile of a VAMP 257 protection relay in motor manager mode.

Class	Object	Object Category	
0x01	Identity	Protocol	Standard
0x02	Message Router		
0x04	Assembly		
0x06	Connection Manager		
0xF5	TCP/IP Interface		
0xF6	Ethernet Link		
0x29	Control Supervisor	Application	Private (vendor specific)
0x2C	Overload		
0x64	Digital		
0x65	Analog		
0x66	StgProtCurrent		
0x67	StgProtEF		
0x68	StgProtOther		
0x69	StgGeneral		
0x70	Analog2		
0x71	Special		

The Ethernet/IP implementation in VAMP devices supports three types of communication:

- Unconnected Explicit Requests and Responses – used mainly for establishing explicit and I/O connections, but can also be used for one time requests to attributes of data model objects.
- Explicit Messaging (Class 3) connections – these are strictly point-to-point connections used to cyclically query the adapter for some data or to cyclically write data to the adapter. Transmitted using *TCP*.
- I/O Messaging (Class 1) connections – can be point-to-point or multicast. Used for very frequent exchange of process data. VAMP relay supports only cyclic I/O connections. Transmitted using *UDP*.

Connection limits:

- Class 3 connections – there can be maximum six Class 3 connections at the same time.
- Class 1 connection – there can be only one Class 1 connection at the same time.

2.2.2 Supported services

VAMP device support following services for objects:

- GAA = Get Attribute All
- GAS = Get Attribute Single
- SAS = Set Attribute Single

GAS service is available for all attributes with the GET or GET | SET access type and the SAS service is available for all attributes with the GET | SET or SET access type

A list of VAMP device's services for objects is shown in the Table 2.2.1-1.

Table 2.2.2-1: Supported services for objects in VAMP device

Class	Object	Supported services	
		Get	Set
0x01	Identity	GAA, GAS	-
0x02	Message Router	-	-
0x04	Assembly	GAS	SAS
0x06	Connection Manager	-	-
0xF5	TCP/IP Interface	GAA, GAS	-
0xF6	Ethernet Link	GAA, GAS	-
0x29	Control Supervisor	GAS	SAS
0x2C	Overload	GAS	SAS
0x64	Digital	GAS	SAS
0x65	Analog	GAS	SAS
0x66	StgProtCurrent	GAS	SAS
0x67	StgProtEF	GAS	SAS
0x68	StgProtOther	GAS	SAS
0x69	StgGeneral	GAS	SAS
0x70	Analog2	GAS	SAS
0x71	Special	GAS	SAS

2.2.3 I/O messaging assemblies

EtherNet/IP implementation on VAMP devices includes total of two producing assemblies (Tx, Target → Originator) and two consuming assemblies (Rx, Target ← Originator); see Table 2.2.3-1.

Table 2.2.3-1: Available assemblies in VAMP device

Instance no.	Type	Description
2	Producing	Static Basic Output Image
50	Consuming	Static Basic Input Image
100	Producing	Configurable Output Image (dynamic)
150	Consuming	Configurable Input Image (dynamic)

Additionally, a zero-length configuration assembly with instance number '199' is available.

Note: when updating from firmware versions older than 10.96, value of configuration assembly instance has to be changed manually to '199'.

Assemblies have to be configured during the device setup. Configuring assemblies involve selecting the producing and consuming instances to be used.

If dynamic assemblies (instance numbers 100 & 150) are used it is also needed to configure the contents of both assembly. By default both assemblies are configured with one byte of data each. By default producing assembly is configured with 'Control Supervisor Object' / 'Faulted attribute' and consuming assembly with 'Control Supervisor Object' / 'FaultRst'.

I/O connections with VAMP device should be opened with the RPI (Requested Packet Interval) no less than 50ms and not greater than 5s. Default Value is 100ms.

2.2.4 Electronic Data Sheet (EDS)

Every change to main configuration parameters or assemblies configuration requires a new EDS file to be generated (once all changes are made and the device is about to be used in the network).

Some of the configuration tools are capable of simplifying device configuration based on the EDS file. In the current implementation the EDS file can only be generated from VAMPSET tool – EDS file extraction over the EtherNet/IP network is not supported in VAMP device.

EDS file cannot be extracted from VAMP devices over the Ethernet/IP network, rather, it must be generated with VAMPSET. This operation is explained in section 0 of this document.

2.2.5 Events

VAMP device events are available under the following attributes of the Digital Object (0x64):

- Attribute 147 – Event Code (bits 0-5: code, bits 6-15: channel)
- Attribute 148 – Event Milliseconds And Seconds (bits 0-5: seconds, bits 6-15: milliseconds)
- Attribute 149 – Event Min And Hour (bits 0-7: hour, bits 8-15: minutes)
- Attribute 150 – Event Day And Month (bits 0-7: month, bits 8-15: day)
- Attribute 151 – Event Year.

Events are read starting from the oldest one in the Event Buffer on the VAMP device. Events are read sequentially, the next event is read when the previous one is acknowledged.

Acknowledgement is done by setting attribute 152 of the Digital Object (0x64) – Event Ack. When all events have been read and the event buffer thus is empty, the attributes will contain zero-data (zeroes). This zero-data will automatically be replaced with the data of a new event when one is registered.

Reading of events is the same for all communication types. The relay sends the oldest available event. Next oldest event will be read only after setting Event Ack parameter (the one previously read is not available any more) or after event buffer has been cleared and a new event was generated later on.

2.2.6 Fault codes

Table 2.2.6-1 below contains a translation of VAMP protection stages to Ethernet/IP Fault Codes.

Table 2.2.6-1: Ethernet/UIP fault code to VAMP protection stage translation.

Ethernet/IP fault code	Description	VAMP protection stage	
20	CURRENT TRIP	Overcurrent Stage I>	50/51
		Overcurrent Stage I>>	50/51
		Overcurrent Stage I>>>	50/51
21	THERMAL OVERLOAD	Thermal Overload Stage T>	49
26	PHASE IMBALANCE	Unbalance Stage I2>	46
27	GROUND FAULT	Earth Fault Stage Io>	50N/51N
		Earth Fault Stage Io>>	50N/51N
		Earth Fault Stage Io>>>	50N/51N
		Earth Fault Stage Io>>>>	50N/51N
29	UNDERLOAD	Under Current Stage I<	37
31	STALL	Stall Protection Stage Ist>	48
51	UNDERVOLTAGE	U<	27
		U<<	27
		U<<<	27
52	OVERVOLTAGE	U>	59
		U>>	59
		U>>>	59
54	PHASE REVERSAL	Unbalance Stage I2>>	47
55	FREQUENCY	f<	81L
		f<<	81L
		f><	81
		f>><<	81
73	START/HOURS EXCEEDED	Frequent Start Protection N>	66

3 Configuration

This section will explain how to configure a VAMP device to use Ethernet/IP.

Note: The configuration and features available may vary slightly between different VAMP devices and between different firmware versions.

3.1 General

To begin with, VAMPSET must be connected to the device. This is typically done by connecting a “USB to RS-232” converter to the PC and then connecting a VX003 cable from the converter to the front port of the device. Alternatively, an Ethernet connection can be used. See the device manual for more information on the different ports on the device.

The Ethernet/IP protocol is activated by first configuring the Ethernet port settings. These settings can be found by navigating to the *PROTOCOL CONFIGURATION* menu in the VAMPSET Group List (the list to the left in the VAMPSET window), see Figure 3.1-1. If DHCP (Dynamic Host Configuration Protocol) is not enabled the *IP address*, *Subnet mask* and *Gateway* must be set. This must be considered before connecting a relay to an existing network, so that no conflicts emerge.

Note: There are two TCP port instances “TCP PORT 1st INST” and “TCP PORT 2nd INST”, that is, two independent sockets for two different protocols. The default TCP IP port for Ethernet/IP is 44818.

The Ethernet/IP protocol is activated by setting it as the *Ethernet port protocol* on one of the *TCP port instances*. The protocol is set by clicking on one of the highlighted parts in Figure 3.1-1 and selecting “EtherNetIP”.

Note: Setting a protocol on any port will require a reboot of the device before the changes will take effect. VAMPSET will prompt for a device reboot when a change has been made.

```

COLD LOAD/INRUSH
+ OVERCURRENT STAGE...
+ DIRECTIONAL O/C STAGE...
+ REVERSE POWER STAGE...
UNDER CURRENT STAGE I< 37
UNBALANCE STAGE I2> 46
THERMAL OVERLOAD STAGE I> 49
+ EARTH-FAULT STAGE...
+ DIRECTIONAL E/F...
TRANSIENT INTERMITTENT E/F 67NI
+ OVERVOLTAGE STAGE...
+ UNDERVOLTAGE STAGE...
+ ZERO SEQ. VOLTAGE STAGE...
+ FREQUENCY STAGE...
+ UNDERFREQUENCY STAGE...
FREQUENCY STAGE df/dt 81R
+ PROGRAMMABLE STAGE...
CB FAILURE PROTECTION 50BF
+ ARC...
2nd HARMONIC O/C STAGE If2> 51F2
5th HARMONIC O/C STAGE If5> 51F5
O B J E C T S
AUTO RECLOSING 79
AR Shot settings 79
AR COUNTER TEXTS
CBWEAR
+ SUPERVISOR...
RELEASE OUTPUT MATRIX LATCHES
+ MATRIX...
L O G I C
M I M I C
+ LOCAL...
S C A L I N G
STAGE EVENT ENABLING
+ EVENT MASKS for AR...
DI EVENT TEXTS
+ EVENT ENABLING...
SYSTEM CLOCK
CLOCK SYNC
P R O T O C O L C O N F I G U R A T I O N
EtherNet/IP main configuration
EtherNet/IP I/O 2+50 (static)
EtherNet/IP I/O 100+150 (dyn.)
EtherNet/IP I/O 101+151 (static)
                
```

ETHERNET PORT	
MAC address	001AD3005742
Enable DHCP service	<input type="checkbox"/>
Enable IP verification service	<input type="checkbox"/>
IP Address	10.4.128.72
NetMask	255.255.255.0
Gateway	10.4.128.254
NTP server	10.4.128.250
IP port for setting tool	23
TCP keepalive interval	0 s
Eth Port status	10M HD

FTP SERVER	
Enable FTP server	<input type="checkbox"/>
FTP password	config
FTP max speed	4 kB/s

TCP PORT 1st INST	
Ethernet port protocol	EtherNet/IP
IP port for protocol	44818
Message counter	0
Error counter	0
Timeout counter	0

TCP PORT 2nd INST	
Ethernet port protocol 2nd inst	None
IP port for protocol 2nd inst	502
Message counter	0
Error counter	0
Timeout counter	0

The IP address of the relay

The Subnet mask of the relay

The gateway to which the relay is connected

The SNTP server IP address (if used)

Ethernet port protocol selection

IP port selection

Ethernet port protocol selection

IP port selection

Figure 3.1-1: The protocol configuration menu in VAMPSET.

3.2 Ethernet/IP main configuration

The configuration of the Ethernet/IP settings is done in the *EtherNet/IP MAIN CONFIGURATION* menu in VAMPSET. The available settings are shown in Figure 3.2.1 and explained in Table 3.2-1.

The screenshot shows the VAMPSET configuration interface. On the left, a tree view lists various system settings, with 'EtherNet/IP main configuration' highlighted. On the right, a configuration panel is displayed with several sections:

- Multicast Parameters:**
 - Multicast IP Address: 239.0.0.1
 - Multicast TTL: 1
 - Vendor ID: 1194
 - Device Type: 43
 - Product Code: 1
 - Major revision: 1
 - Minor revision: 97
 - Serial Number: 00005F70
 - Product Name: VAMP
 - Auto/Manual Header Detection: Auto
 - I/O assembly instances in use: 2+50
- Target => Originator Conn Params:**
 - Producing Instance: 2
 - Include Run/Idle Header:
 - Producing Instance Size: 1
- Target <= Originator Conn Params:**
 - Consuming Instance: 50
 - Include Run/Idle Header:
 - Consuming Instance Size: 1
- Configuration Assembly Params:**
 - Configuration Instance: 199
 - Configuration Instance Size: 0

Figure 3.2.1: The Ethernet/IP main configuration menu in VAMPSET.

Table 3.2-1: Description of the Ethernet/IP main configuration parameters.

Parameter	Range	Description
Multicast IP Address	239.0.0.0...239.255.255.255	Multicast IP address used for sending I/O messages
Multicast TTL	1...100	Time-to-live of the I/O messages sent as multicast
Vendor ID	1...65535	Identification of the vendor by number
Device Type	0...65535	Identification of the general type of product
Product Code	1...65535	Identification of the particular product of an individual vendor
Major revision	1...127	Major revision of the item that the Identity Object represents
Minor revision	1...255	Minor revision of the item that the Identity Object represents

Parameter	Range	Description
Serial Number	0...4294967295	Serial number of the device
Product Name	32 characters	Human readable identification
Auto/Manual Header Detection	Manual	Include Run/Idle Header parameters are used to define whether Run/Idle Header should be used
	Auto (default)	Run/Idle Headers are detected automatically
I/O assembly instances in use	2+50, (default) 100+150, 101+151	Instance numbers of producing and consuming assemblies being used
Producing Instance	[producing instance number]	Instance number of the producing assembly (not directly editable) *
Include Run/Idle Header	On	Include Run/Idle Header in outgoing I/O messages
	Off (default)	Exclude Run/Idle Header in outgoing I/O messages
Producing Instance Size	[bytes]	The size of the producing assembly (not directly editable) **
Consuming Instance	[consuming instance number]	Instance number of the consuming assembly (not directly editable) *
Include Run/Idle Header	On (default)	Expect presence of Run/Idle Header in incoming I/O messages
	Off	Expect absence of Run/Idle Header in incoming I/O messages
Consuming Instance Size	[bytes]	The size of the consuming assembly (not directly editable) **
Configuration Instance	1...1279	Instance number of configuration instance
Configuration Instance Size	[bytes]	The size of the consuming assembly (not directly editable) **

* Automatically updated according to the value of 'I/O assembly instances in use' parameter.

** Automatically updated as the assemblies are configured.

3.2.1 IP address

Each device in each subnetwork must have a unique IP address. Valid Class A, B or C IP address needs to be assigned to the device to allow it to serve as a TCP/IP node. Assigned IP address will be used by the EtherNet/IP scanner to find the device in the network and start a communication with it.

3.2.2 Multicast IP

Multicast IP address is a parameter used by the device to send EtherNet/IP multicast packets, if requested to do so by the scanner. Multicast IP should be a valid Class D IP address. In VAMP device there is only one possible I/O connection at a time and therefore only single Multicast IP is used.

Multicast IP parameter is ignored when scanner requested T→O (target-to-originator, i.e. adapter to scanner) communication to be point-to-point.

The default value of this parameter is 239.0.0.1.

3.2.3 Multicast TTL

Multicast TTL value should be used for the IP header Time-to-live field when sending EtherNet/IP multicast packets. This value is ignored for the unicast packets and TTL as configured for the TCP/IP stack is used instead.

The default value of this parameter is 1 (the number of network hops over which the multicast packet is propagated – datagrams limited to the local subnet).

3.2.4 Auto/Manual Header Detection

When this parameter is set to 'Manual', user needs to define manually the presence of Run/Idle header in the I/O messaging. This is done by configuring parameters 'Include Run/Idle Header' for more information, see sections 3.2.6 Include Run/Idle Header (Producing) and 3.2.7 Include Run/Idle Header (Consuming).

When 'Auto' is selected VAMP device automatically detects whether Run/Idle header is used in the I/O messaging.

The default value of this parameter is 'Auto'.

3.2.5 I/O assembly instances in use

Selection of producing and consuming instances to be used. The values of producing and consuming assemblies are available in the EDS file and are used by the configuration tool as a reference path during I/O connection opening.

Every change to this parameter requires restarting the device and generation of the new EDS file.

3.2.6 Include Run/Idle Header (Producing)

An I/O connection can be established with or without the Run/Idle Header in the T→O direction (adapter to scanner). Including Run/Idle Header in the producing assembly adds additional 4 bytes to the beginning of the data part of an I/O message. Run bit is always set in the outgoing messages if VAMP device is configured to send I/O messages with the Run/Idle Header. Information about whether the Run/Idle Header is included in the outgoing messages is available in the EDS file and can be used by the configuration tool to properly establish I/O communication.

Every change to this parameter requires generation of the new EDS file.

The default value of this parameter is 'Off'.

3.2.7 Include Run/Idle Header (Consuming)

An I/O connection can be established with or without the Run/Idle Header in the T←O direction (scanner to adapter). Setting this value to 'On' inform the configuration tool that VAMP device expects the consuming assembly to contain additional 4 bytes of data. If the Run/Idle Header is included and the Run bit is set in the incoming I/O messages then VAMP device processes received data, and if the Run bit is cleared then VAMP device ignores received data. If the Run/Idle Header is not included in the incoming I/O messages then the received data is always processed. Information about whether the Run/Idle Header is expected in the incoming I/O messages is available in the EDS file and can be used by the configuration tool to properly establish I/O communication.

Every change to this parameter requires generation of the new EDS file.

The default value of this parameter is 'On'.

NOTE: VAMP device does not allow changing of both Include Run/Idle Header parameters while the I/O connection is running.

3.3 Data point configuration

Available data items, that is, the contents of the Producing Assembly and the Consuming Assembly can be viewed / configured in the following VAMPSET menus:

- EtherNet/IP I/O 2+50 (static)
- EtherNet/IP I/O 100+150 (dyn.)

Assembly 2+50 is static, meaning user cannot make changes to the contents of assembly.

Assembly 100+150 is dynamic, meaning user can select data items to the assembly by clicking on a row and selecting a desired data point. An example of this is shown in Figure 3.3-1 below. The columns of the assembly table are described in Table 3.3-1.

Producing Assembly (Tx, T => O)			
Name	Length	Scaling	Offset
Control Supervisor#010 / Faulted	1	1 = 1	000
Digital#170 / Device status	1	MotorRun=0,Warning=1,Fault=2	001
Digital#108 / Virtual input 1	1	0,1	002
Overload#008 / CurrentL1	2	1.0 A = 10	003
			004
			005
			006
			007
			008
			009
			010
			011
			012
			013
			014
			015
			016

Editing the Producing Assembly

- Digital#174 / Virtual output 4
- Digital#175 / Virtual output 5
- Digital#176 / Virtual output 6
- Analog#001 / Phase current IL1
- Analog#002 / Phase current IL2
- Analog#003 / Phase current IL3

Figure 3.3-1: Configuration of the Ethernet/IP Producing Assembly 100 in VAMPSET.

Table 3.3-1: Description of Assembly configuration table contents.

Attribute	Description
Name	Type and name of the data item
Length	Length of the data item in bytes
Scaling	The scaling used for the data item
Offset	The offset of the data item in the assembly

Note:

1. Making changes to the assemblies will require a device reboot for the changes to take effect.
2. A list of the set of available data items in Ethernet/IP, see section 5 of this document.

3.4 Generating an EDS file with VAMPSET

An EDS file can be generated with VAMPSET. This is done by navigating to “Communication” → “Get EtherNet/IP EDS...”, see Figure 3.4-1. Selecting this option will generate the EDS file and bring up a file browser window asking where to save the generated file, see Figure 3.4-2. After clicking Save the generated EDS file will be stored at the selected location.

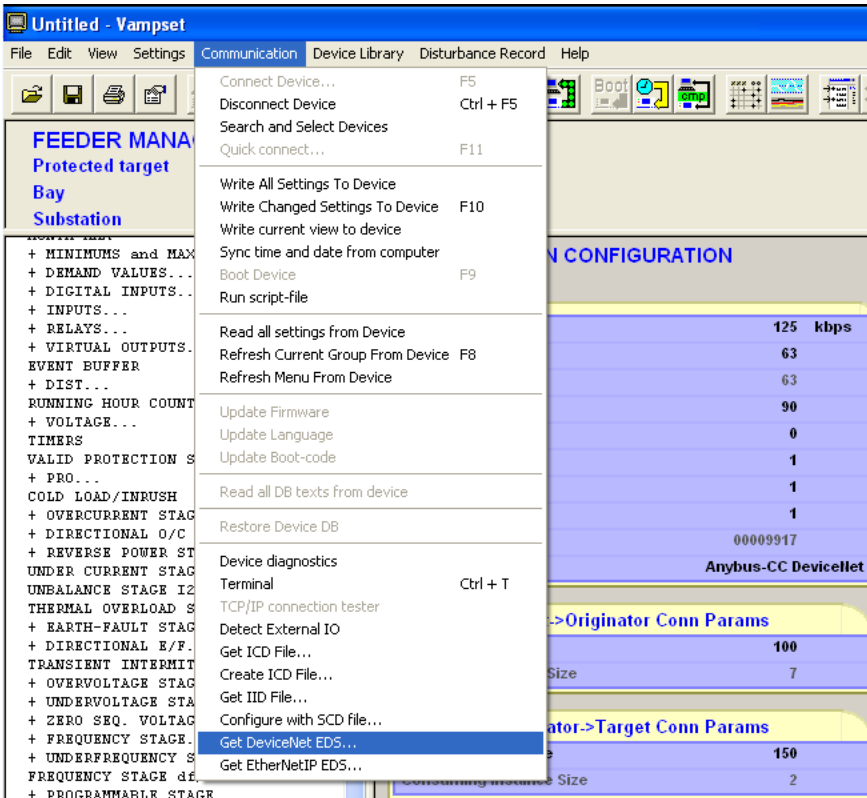


Figure 3.4-1: Creating an EDS file with VAMPSET.

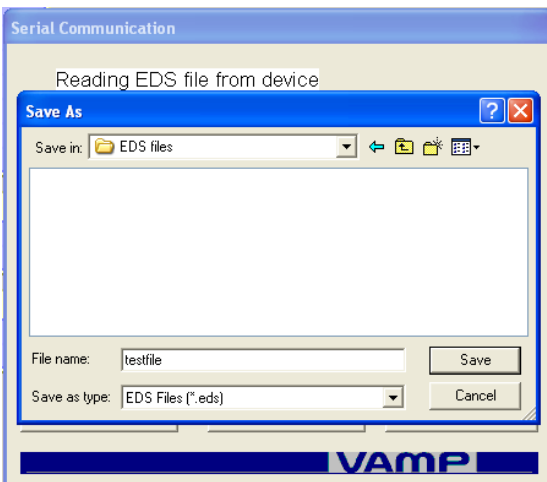


Figure 3.4-2: Saving the EDS file.

4 Testing with Simple Tester

The purpose of this section is to give a description of how to test that the Ethernet/IP protocol has been successfully configured through a few examples with the *Simple Tester* testing tool.

“DUT” will be used in this section to refer to the Device Under Test, that is, the VAMP device to which Simple Tester is connected.

4.1 Simple Tester overview

Simple Tester is a basic protocol testing tool (for PC) that supports several different protocols, including Ethernet/IP. Simple Tester can be used to test reading and writing data as well as viewing events and performing clock synchronisation from the PC. The user interface of Simple Tester with Ethernet/IP selected as current protocol (but not connected to a device) is shown and explained in Figure 4.1-1.

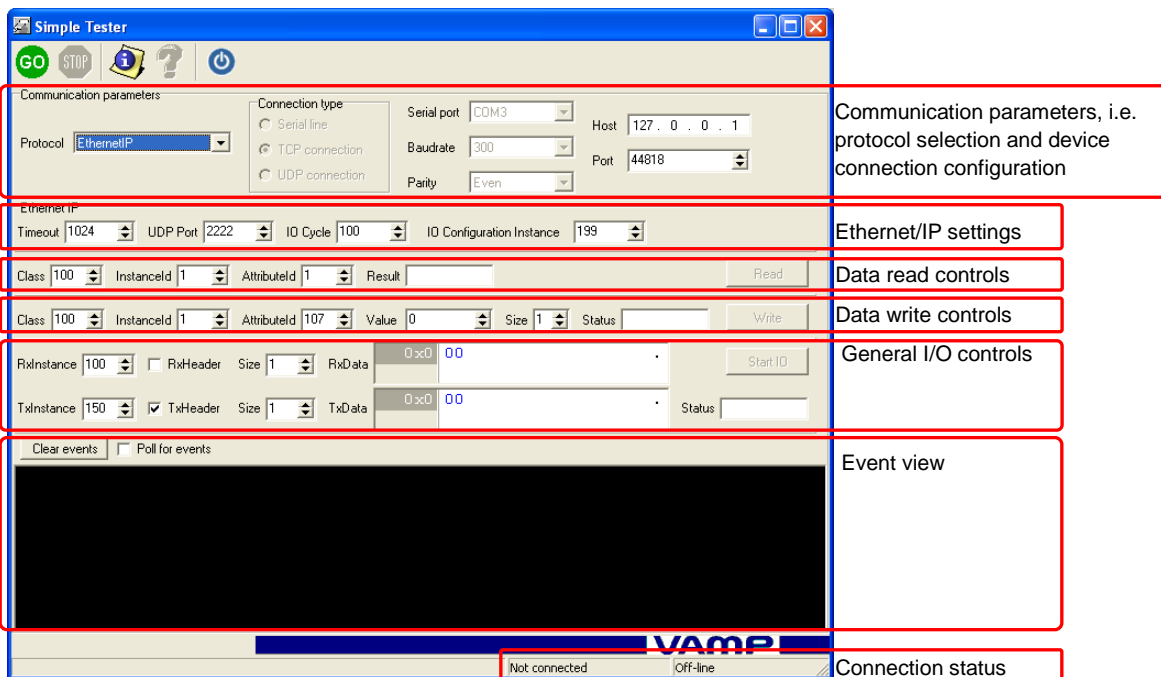


Figure 4.1-1: The user interface of Simple Tester with Ethernet/IP selected. Not connected to a device.

4.2 Connecting Simple Tester

In order to connect to a device over Ethernet/IP with Simple Tester, the device must first be configured as described in section 3 of this document, and a physical Ethernet connection be made to the device.

Connecting Simple Tester is done as follows (please refer to Figure 4.2-1):

1. Select “EthernetIP” in the *Protocol* field in the *Communication Parameters* section.
2. Enter the IP address of the device under test in the *Host* field of the *Communication parameters* section.
3. Set the IP port in the *Port* field of the *Communication Parameter* section. This refers to the IP port as set in the *IP port for protocol* setting in the protocol configuration menu in VAMPSET (see Figure 3.1-1).
4. Click “GO”.
5. A successful connection will be indicated by an “OK” down in the rightmost corner of the Simple Tester window. See Figure 4.2-1 for an example of a successful connection.

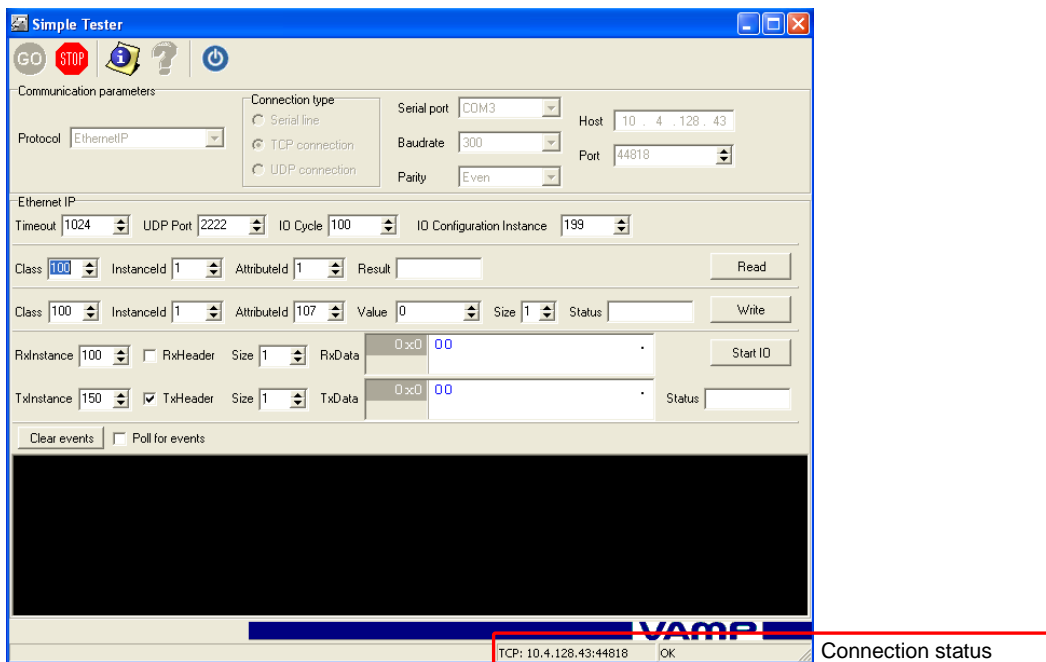


Figure 4.2-1: Simple Tester connected.

4.3 Reading data

Data can be read in two different ways: by specific data item, or by the Producing Assembly of the device. In the former case, the *Data read controls* are used, in the latter case the *general I/O controls* are used (see Figure 4.1-1).

This will be shown through a few examples:

Some virtual measurements are sent to the device with VAMPSET. These, as well as the Assembly configurations are shown in Figure 4.3-1.

The screenshot displays two tables for assembly configurations and a panel for virtual measurements.

Producing Assembly (Tx, T->O)

Name	Length	Scaling	Offset
Control Supervisor#010 / Faulted	1	1 = 1	000
Digital#169 / DevStatus	1	1 = 1	001
Digital#107 / Virtual input 1	1	0,1	002
Overload#008 / CurrentL1	2	1.0 A = 10	003
			004
Analog#004 / Frequency	2	50.000 Hz = 5000	005
			006
			007
			008
			009
			010
			011
			012
			013
			014
			015
			016

Consuming Assembly (Rx, O->T)

Name	Length	Scaling	Offset
Control Supervisor#012 / FaultRst	1	1 = 1	000
Digital#107 / Virtual input 1	1	0,1	001
			002
			003
			004
			005
			006
			007
			008
			009
			010
			011
			012
			013

Virtual Measurements Panel:

- Enable virtual measurement:
- IL1: 536 A
- IL2: 507 A
- IL3: 521 A
- U12: 11450 V
- U23: 10850 V
- Uo: 0.0 %
- Io1: 0.00 A
- Io2: 0.000 A

Cycles: 100000, Frequency: 50.000 Hz

Figure 4.3-1: The Assembly configurations and virtual measurements in VAMPSET.

The simulated (virtual measurement) of Phase 1 current, IL1, is 536 A. This value can be found in the Ethernet/IP configuration as: **Class 0x2C, Instanceld 0x01, AttributeID 0x08**. These values are entered into the *Read controls* of Simple Tester as **decimal values**. The result is shown in Figure 4.3-2. To read the value, click the *Read* button.

The screenshot shows the Simple Tester interface with the following fields:

- Class: 44
- Instanceld: 1
- AttributeID: 8
- Result: 5370
- Class: 100
- Instanceld: 1
- AttributeID: 107
- Value: 0
- Size: 1
- Status: [Empty]
- RxInstance: 100
- RxHeader:
- Size: 7
- RxData: 0x0 0000 00FA 1453 C3
- TxInstance: 150
- TxHeader:
- Size: 2
- TxData: 0x0 0000
- Status: Success

Figure 4.3-2: Result of the first data read.

The result, 5370, corresponds to the virtual value, 536 A fed to the device by VAMPSET. To be noted here is the scaling, which applies to objects of type Overload. The scaling is indicated in the *Scaling* column of the assembly configuration as “1.0A = 10”, which means that the observed value would represent 537.0 A.

The value of IL1 is also available as an attribute of an object of class Analog (0x65) with the parameters: **Class 0x65, Instanceld 0x01, AttributeID 0x01**. The result of a read of these values is shown in Figure 4.3-3. Please note that the values have been entered as decimal values. The result in this case, 537 again corresponds to the virtual value, 536 A, but has not been scaled, since values represented with objects of class Analog are not scaled.

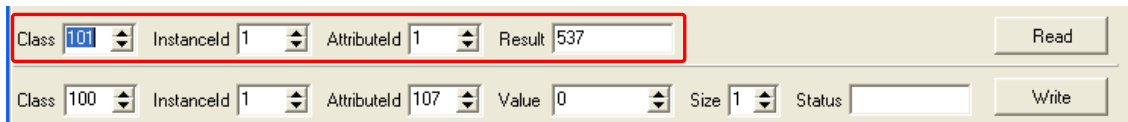


Figure 4.3-3: Result of the second data read.

When using the general I/O controls, the {RxInstance, RxHeader, Size} and {TxInstance, TxHeader, Size} parameters need to be configured to correspond to the definition of the consuming and producing assemblies on the DUT. These settings are seen from the point of view of the test application, not the DUT. Thus, the values of the parameters {RxInstance, RxHeader, Size} should correspond to those of the Producing Assembly on the DUT, and the values of the parameters {TxInstance, TxHeader, Size} should correspond to those of the Consuming Assembly on the DUT. The sizes of the assemblies can be checked from the the *EtherNet/IP MAIN CONFIGURATION* menu in VAMPSET. In this example, this leads to the configuration as seen in Figure 4.3-2.

When the data is to be read, click the *Start IO* button. A successful action will be indicated in the *Status* field of the General I/O controls.

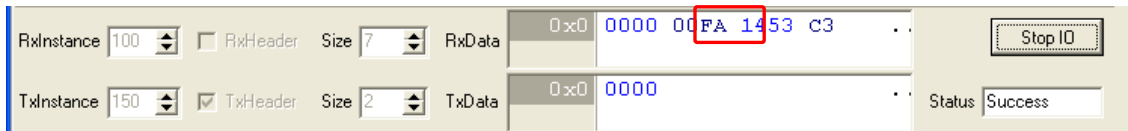


Figure 4.3-4: Result of general I/O.

Now the received data must be interpreted. As can be seen in the Producing Assembly configuration (Figure 4.1-1), the offset of IL1 (CurrentL1) is 003 and 004. The two bytes are in the order “low byte, high byte”, the two bytes are highlighted in Figure 4.3-4 above. The value of the data is thus:

0x14FA = 5370, which again corresponds to the virtual value, 536 A, and because the value is contained within an object of class Overload, it is scaled as indicated by the *Scaling* column of the Assembly configuration.

The Analog class representation of this data item is not available via the Assemblies, as it has not been set. This can, however, be done by following the instructions in subsection 3.3 of this document.

4.4 Writing data

Data can also be written by either writing to a particular data item, or by writing to an item defined in the consuming assembly of the DUT.

This will be described with two examples:

In order to write directly to a data item, the *Write controls* in Simple Tester are used. For this example a “1” is written to Virtual Input 1 (VI1), which can be found as an object of the **Digital Class (0x64), Instanceld 0x01 and Attributeld 0x6B**. This translates to the decimal values 100, 1 and 107.

These values are entered into their respective fields in the *Write controls* and a “1” is entered in the *Value* field. The write is performed by clicking the *Write* button. A screenshot of Simple Tester after the write to VI1 is shown in Figure 4.4-1. A successful write will be indicated by the message “Success” in the *Status* field.

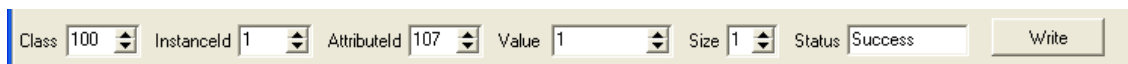


Figure 4.4-1: Simple Tester after writing to Virtual Input 1.

In order to write to VI1 via the *general I/O controls*, the item must first be defined in the consuming assembly of the DUT. Then I/O must be active, which is done by clicking the *Start I/O* button (if not already clicked).

In Figure 4.3-1 VI1 can be seen in the configuration of the consuming assembly at offset 1. To perform the actual write, click on the lower half-byte of the byte at offset 1 and enter a “1”. This is shown in Figure 4.4-2 and Figure 4.4-3.

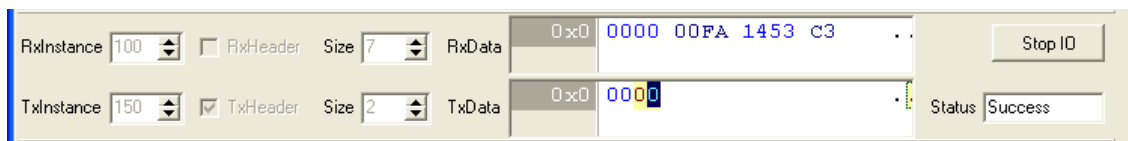


Figure 4.4-2: Selecting the half-byte to write to.

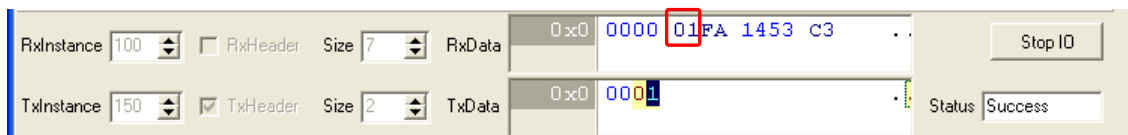


Figure 4.4-3: Result after writing to VI1 in the General I/O section.

In this example, VI1 has also been defined in the producing assembly at offset 1, see Figure 4.3-1. Thus, its value can also be seen at offset 2 in the producing assembly of the DUT (RxData in Simple Tester), highlighted in Figure 4.4-3.

4.5 Events

Events can be viewed by checking the *Poll for events* checkbox. This will poll the DUT for any events in its event buffer. Events are in the format “<timestamp> <event code>”, where the event codes are the same as used by the SPA-bus protocol. See for an example of this.

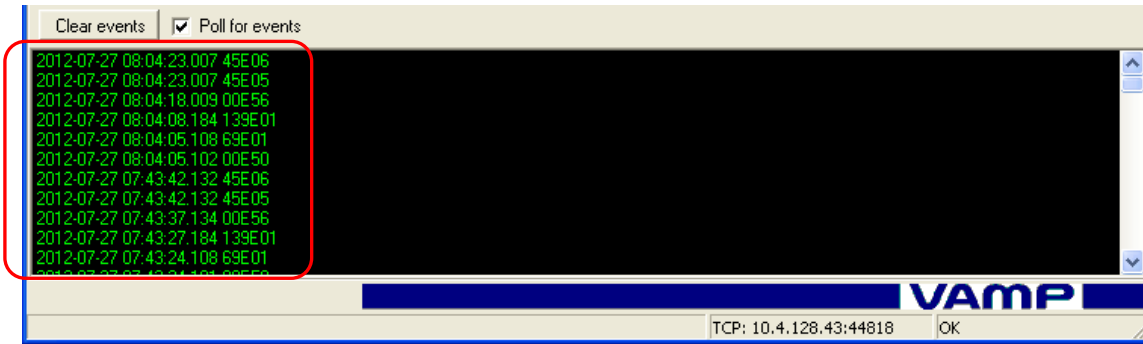


Figure 4.5-1: Example of events listed in the event view when “Poll for events” checkbox is checked.

The meaning of the SPA-Bus event codes can be found in a separate document available at the website.

5 Data model

Please refer to [1]

Schneider Electric

35 rue Joseph Monier
92506 Rueil-Malmaison

FRANCE

Phone: +33 (0) 1 41 29 70 00

Fax: +33 (0) 1 41 29 71 00

www.schneider-electric.com

Publishing: 11/2014

© 2014 Schneider Electric Industries SAS - All rights reserved.

Electrical equipment should be installed, operated, serviced, and maintained only by qualified personnel. No responsibility is assumed by Schneider Electric for any consequences arising out of the use of this material.