ETHERNET/IP FOR SNAP PAC PROTOCOL GUIDE

SNAP-PAC-S1	**SNAP-PAC-EB2**
SNAP-PAC-S2	**G4EB2**
SNAP-PAC-R1	**OPTOEMU-SNR-3V**
SNAP-PAC-R2	**OPTOEMU-SNR-DR1**
SNAP-PAC-EB1	**OPTOEMU-SNR-DR2**

Form 1770-130915—September 2015

OPTO 22
Automation made simple.

43044 Business Park Drive • Temecula • CA 92590-3614
Phone: 800-321-OPTO (6786) or 951-695-3000
Fax: 800-832-OPTO (6786) or 951-695-2712
www.opto22.com

Product Support Services
800-TEK-OPTO (835-6786) or 951-695-3080
Fax: 951-695-3017
Email: support@opto22.com
Web: support.opto22.com

EtherNet/IP for SNAP PAC Protocol Guide
Form 1770-130915—September 2015

The information in this manual has been checked carefully and is believed to be accurate; however, Opto 22 assumes no responsibility for possible inaccuracies or omissions. Specifications are subject to change without notice.

Opto 22 warrants all of its products to be free from defects in material or workmanship for 30 months from the manufacturing date code. This warranty is limited to the original cost of the unit only and does not cover installation, labor, or any other contingent costs. Opto 22 I/O modules and solid-state relays with date codes of 1/96 or newer are guaranteed for life. This lifetime warranty excludes reed relay, SNAP serial communication modules, SNAP PID modules, and modules that contain mechanical contacts or switches. Opto 22 does not warrant any product, components, or parts not manufactured by Opto 22; for these items, the warranty from the original manufacturer applies. These products include, but are not limited to, OptoTerminal-G70, OptoTerminal-G75, and Sony Ericsson GT-48; see the product data sheet for specific warranty information. Refer to Opto 22 form number 1042 for complete warranty information.

Opto 22
Automation Made Simple.

Table of Contents

Chapter 1: Welcome 1

Using this Guide 2
Documents and Online Help 2
Additional Resources for EtherNet/IP 3
Product Support 3

Chapter 2: CIP Object Model for SNAP PAC 5

CIP Object Classes 5
Accessing Object Classes in Implicit and Explicit Messaging 6
For Implicit Messaging 6
For Explicit Messaging 7
Points, Ports, and Channels 7
Column Descriptions 8
0x01 - Identity 9
Class Attributes 9
Instance Attributes 9
Common Services 10
Vendor Specific Services 10
0x02 - Message Router 11
0x04 - Assembly 11
Class Attributes 12
Instance Attributes 12
Common Services 12
0x06 - Connection Manager 12
Object Specific Services 13
0x08 - Discrete Input Point 13
Class Attributes 13
Instance Attributes 13
Features for 0x08 - Discrete Input Point, Attribute 0x80 19
Common Services 25
Vendor Specific Services 26
0x09 - Discrete Output Point 26
Class Attributes 26
Instance Attributes 26

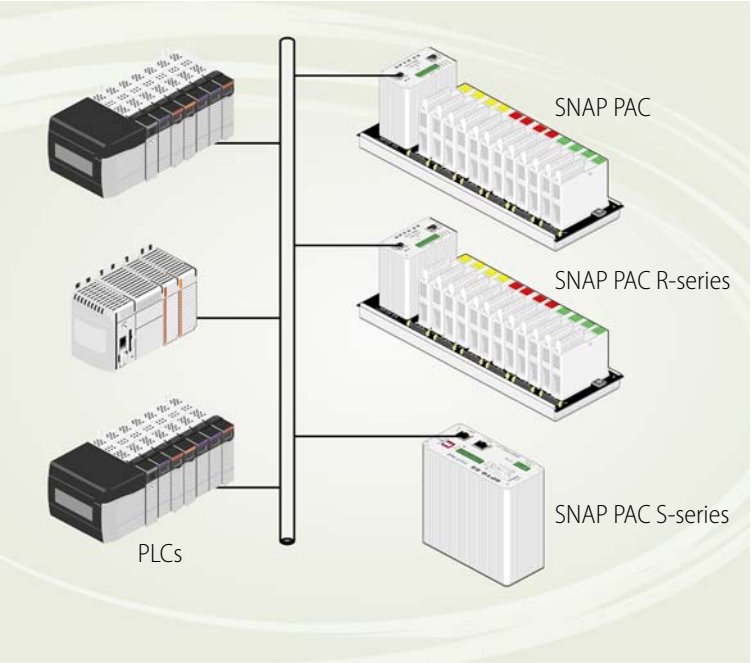
Common Services 30
Vendor Specific Services 30
0x0A - Analog Input Point 31
Class Attributes 31
Instance Attributes 31
Common Services 37
Vendor Specific Services 37
Analog Input: Value Attribute Semantics 38
0x0B - Analog Output Point 40
Class Attributes 40
Instance Attributes 40
Common Services 44
Vendor Specific Services 44
Analog Output: Value Attribute Semantics 44
0x66 - Load Cell Input 45
Class Attributes 45
Instance Attributes 46
Common Services 53
Vendor Specific Services 53
Load Cell Input: Value and Value32 Attribute Semantics 54
0x68 - OptoMMP Request 55
Class Attributes 55
Common Services 55
Object Specific Services 55
0x69 - Scratchpad DINT 57
Class Attributes 57
Instance Attributes 57
Common Services 57
0x70 - Scratchpad REAL 58
Class Attributes 58
Instance Attributes 58
Common Services 58
0x71 - Scratchpad STRING 58
Class Attributes 59
Instance Attributes 59
Common Services 60
0x73 - Pulse and TPO Generator 60
Class Attributes 60
Instance Attributes 61
Common Services 63
0x74 - Ramp Controller 63
Class Attributes 63
Instance Attributes 63
Common Services 64
0x75 - PID Loop Controller 65
Class Attributes 65
Instance Attributes 66

Common Services 71
0x76 - Serial Port 73
Class Attributes 73
Instance Attributes 73
Common Services 77
Object Specific Services 78
0x80 - Communication Watchdog 79
Class Attributes 79
Instance Attributes 79
Common Services 80
0x81 - PPP Link 80
Class Attributes 80
Instance Attributes 80
Common Services 80
0xF5 - TCP/IP Object 81
Class Attributes 81
Instance Attributes 82
Common Services 84
0xF6 - Ethernet Link 85
Class Attributes 85
Instance Attributes 86
Common Services 87

1: Welcome

Welcome to the EtherNet/IP™ for SNAP PAC Protocol Guide. Inside you'll find detailed descriptions of the EtherNet/IP commands that can be accessed when using remote Opto 22 SNAP PAC I/O with an Allen-Bradley® Logix™ controller.

This guide assumes that you are already familiar with how to use Allen-Bradley's RSLogix 5000 software to configure Allen-Bradley devices.



EtherNet/IP automation environment with PLCs and Opto

Using this Guide

Chapter 1: Welcome—Provides information about this guide, where to find additional information, and how to reach Opto 22 Product Support.

Chapter 2: CIP Object Model for SNAP PAC—Describes the SNAP PAC functionality accessible via EtherNet/IP with implicit and explicit messaging.

Documents and Online Help

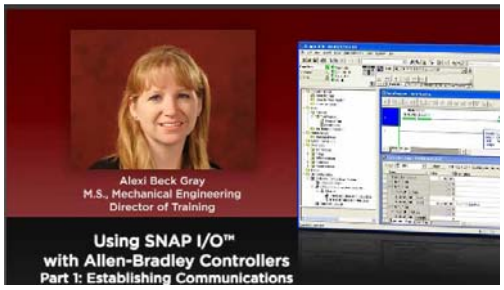
You may also need the following Opto 22 hardware documentation, depending on your application:

For this information	See this guide	Form
Setting up EtherNet/IP messaging between an Allen-Bradley® Logix™ controller and Opto 22 SNAP PAC I/O	*IO4AB User's Guide*	1909
Installing and using SNAP PAC brains	*SNAP PAC Brain User's Guide*	1690
Installing and using SNAP PAC R-series controllers	*SNAP PAC R-Series Controller User's Guide*	1595
Installing and using SNAP PAC S-series controllers	*SNAP PAC S-Series Controller User's Guide*	1592
Using the OptoMMP Communication Toolkit or the IEEE 1394-based OptoMMP memory-mapped protocol for programming	*OptoMMP Protocol Guide*	1465
Wiring, specifications, and installation information for SNAP Analog Input Modules	*SNAP Analog Input Modules Data Sheet*	1065
Wiring, specifications, and installation information for SNAP Analog Output Modules	*SNAP Analog Output Modules Data Sheet*	1066
Wiring, specifications, and installation information for SNAP High-Density Digital Modules	*SNAP High-Density Digital Modules Data Sheet*	1556
Wiring, specifications, and installation information for SNAP Digital Input Modules	*SNAP Digital Input Modules Data Sheet*	0773
Wiring, specifications, and installation information for SNAP Digital Output Modules	*SNAP Digital Output Modules Data Sheet*	1144
Wiring, specifications, and installation information for isolated analog input modules	*SNAP Isolated Analog Input Modules Data Sheet*	1182
Wiring, specifications, and installation information for serial modules	*SNAP Serial Communication Modules Data Sheet*	1184
Latest release, installation, and system requirements information for EtherNet/IP Configurator.	*EtherNet/IP Configurator Release Notes. (See www.opto22.com for the latest form number.)*	

All documents are available on our website, www.opto22.com. The easiest way to find a document is to search on its form number.

Additional resources are also available on the Opto 22 Web site such as a demonstration video that shows how easy it is to configure and program an Allen-Bradley controller to talk to SNAP PAC I/O. If you are viewing this document online and have an internet connection, click the image. Otherwise,

go to the Learn tab on the Opto 22 website, www.opto22.com, and click SNAP I/O with Allen-Bradley Systems.



Additional Resources for EtherNet/IP

For an overview of CIP and EtherNet/IP technology, see
http://www.odva.org/Portals/0/Library/Publications_Numbered/PUB00138R2_CIP_Adv_Tech_Series_EtherNetIP.pdf

For information on the network infrastructure for EtherNet/IP, see
http://www.odva.org/Portals/0/Library/Publications_Numbered/PUB00035R0_Infrastructure_Guide.pdf

For a good starting point for Ethernet/IP research on the ODVA web site, see
http://www.odva.org/default.aspx?tabid=67

Product Support

If you have any questions about using EtherNet/IP to communicate with Opto 22 devices, you can call, fax, or email Opto 22 Product Support.

Phone:	800-TEK-OPTO (800-835-6786) 951-695-3080 (Hours are Monday through Friday, 7 a.m. to 5 p.m. Pacific Time)	*NOTE: Email messages and phone calls to Opto 22 Product Support are grouped together and answered in the order received.*
Fax:	951-695-3017	
Email:	support@opto22.com	
Opto 22 website:	www.opto22.com	

When calling for technical support, be prepared to provide the following information about your system to the Product Support engineer:

- Opto 22 software and version being used
- Opto 22 firmware versions
- PC configuration (type of processor, speed, memory, and operating system)
- PLC software description and version
- PLC model
- PLC firmware version

- A complete description of your hardware and operating systems, including:
  - type of power supply
  - types of remote I/O installed
  - third-party devices installed (for example, barcode readers)
- Description and symptoms of the issue
- Specific error messages seen.

2: CIP Object Model for SNAP PAC

CIP Object Classes

This chapter details the available CIP objects that can be accessed via EtherNet/IP for implicit and explicit messaging.

The following table shows the CIP object classes that are supported on Opto 22 devices. While all of the classes are available for explicit messaging, only the classes marked with an asterisk (*) include *some* attributes that are available for implicit messaging.

Object Class	Devices Supported					
	SNAP-PAC-EB1	SNAP-PAC-EB2	SNAP-PAC-R1	SNAP-PAC-R2	SNAP-PAC-S1	SNAP-PAC-S2
0x01 - Identity	●	●	●	●	●	●
0x02 - Message Router	●	●	●	●	●	●
0x04 - Assembly	●	●	●	●	●	●
0x06 - Connection Manager	●	●	●	●	●	●
0x08 - Discrete Input Point*	●	●	●	●		
0x09 - Discrete Output Point*	●	●	●	●		
0x0A - Analog Input Point*	●	●	●	●		
0x0B - Analog Output Point*	●	●	●	●		
0x66 - Load Cell Input*	●	●	●	●		
0x68 - OptoMMP Request	●	●	●	●	●	●
0x69 - Scratchpad DINT*	●	●	●	●	●	●
0x70 - Scratchpad REAL*	●	●	●	●	●	●
0x71 - Scratchpad STRING*	●	●	●	●	●	●
0x73 - Pulse and TPO Generator*	●	●	●	●		
0x74 - Ramp Controller*	●	●	●	●		
0x75 - PID Loop Controller*	●	●	●	●		
0x76 - Serial Port*	●	●	●	●		
0x80 - Communication Watchdog	●	●	●	●		
0x81 - PPP Link			●	●	●	●
0xF5 - TCP/IP Object	●	●	●	●	●	●
0xF6 - Ethernet Link	●	●	●	●	●	●

Accessing Object Classes in Implicit and Explicit Messaging

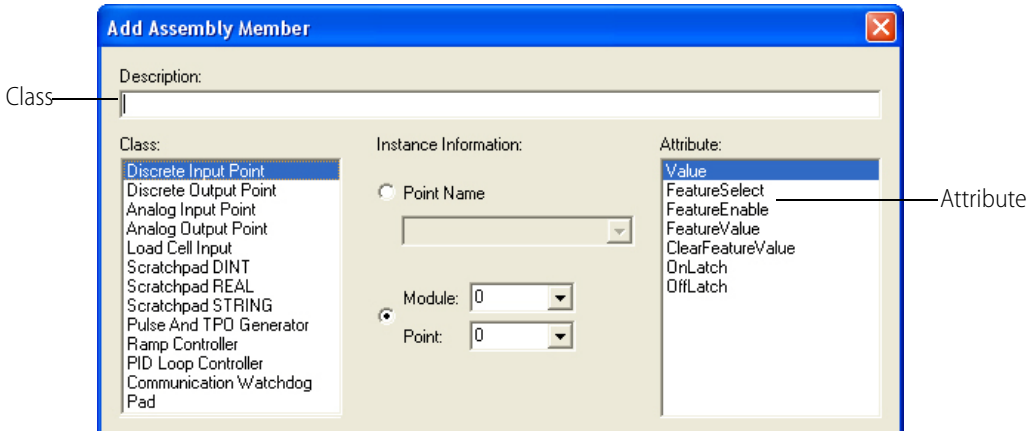
All of the attributes in the object model can be accessed by explicit messaging. Only a subset of the attributes can be accessed by implicit messaging.

Attributes with this background color can only be accessed by explicit messaging.

Attributes with this background color can be accessed by explicit messaging and implicit messaging (through a properly configured assembly instance.)

For Implicit Messaging

Implicit messaging is configured in EtherNet/IP Configurator in the Add Assembly Member dialog box. In the following example, the class called DiscreteInputPoint in EtherNet/IP Configurator is the same as 0x08 - Discrete Input Point in the CIP object model. For more information on using EtherNet/IP Configurator to set up an implicit messaging connection, see form 1909, the *IO4AB User's Guide*.



The same Class and Attribute that are selected above in EtherNet/IP Configurator can be found in the object model tables.

Class

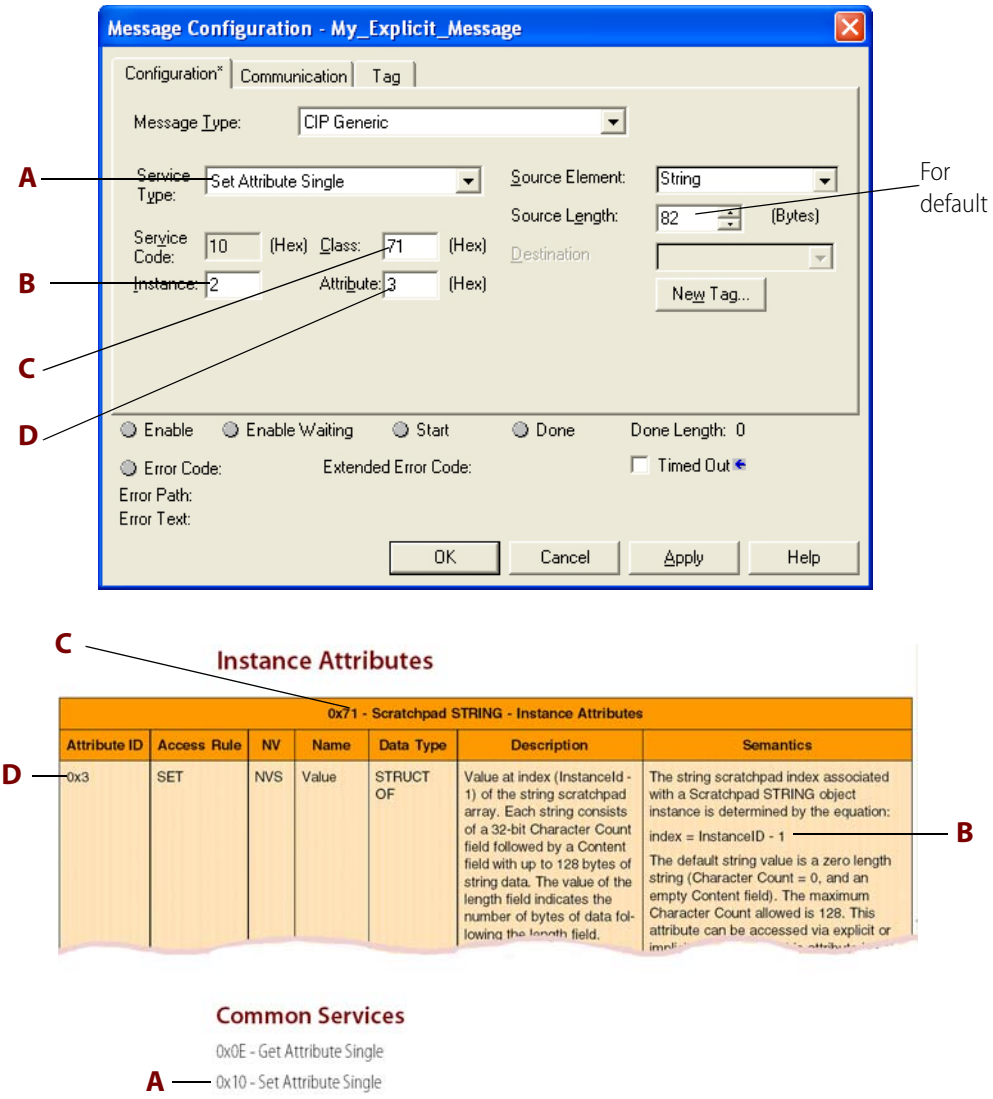
0x08 - Discrete Input Point - Instance Attributes						
Attribute ID	**Access Rule**	**NV**	**Name**	**Data Type**	**Description**	
0x1	GET		Number Of Attributes	USINT	Number supported in this product	
0x2	GET		Attribute List	ARRAY OF USINT	List of attributes supported in this product	
0x3	GET		Value	BOOL	Input Point Value	0 = Off 1 = On

Attribute ID

Attribute name

For Explicit Messaging

Explicit messaging is configured in RSLogix 5000 as described in form 1909, the *IO4AB User's Guide*. In the following example, fields A-D on the Message Configuration dialog box correspond to items in the object model tables.



Points, Ports, and Channels

The terms *point*, *port*, and *channel* are used in this document. Each channel of a SNAP-PAC I/O module provides connectivity to one point of I/O. Each channel of a SNAP-PAC communication module provides connectivity to one communication port.

Column Descriptions

Attribute ID

An integer identification value assigned to an attribute. Use the Attribute ID in the Get_Attributes and Set_Attributes services list. The Attribute ID identifies the particular attribute being accessed.

Access Rule

Specifies how a requestor can access an attribute. The definitions for access rules are:

- Settable (SET) - The attribute can be accessed by one of the Set_Attribute services. *Important*: Settable attributes can also be accessed by Get_Attribute services.
- Gettable (GET) - The attribute can be accessed by one of the Get_Attribute services.

NV

NV indicates whether an attribute value is maintained through power cycles. This column is used in object definitions where non-volatile storage of attributes is required. An entry of 'NV' indicates the value shall be saved, 'V' means not saved and 'NVS' indicates the value shall be saved only when the "Store Configuration In Non-Volatile Memory" service (0x32) for the Identity object is invoked.

Name

Name refers to the attribute.

Data Type

Specifies the data type of the attribute value.

Description

Provides general information about the attribute.

Semantics

Specifies the meaning of the attribute value.

0x01 - Identity

This object class provides identification and general information about the device. Vendor specific services have been implemented that store or erase configuration data in non-volatile memory. There is only one instance of the Identity object class: instance ID 1.

Class Attributes

0x01 - Identity - Class Attributes						
Attribute ID	**Access Rule**	**NV**	**Name**	**Data Type**	**Description**	**Semantics**
0x1	GET		Revision	UINT	Revision of this open object	The revision of the Identity Object class specified by the CIP Common Specification. The current value assigned to this attribute is 1.
0x2	GET		Max Instance	UINT	Maximum instance number for objects of this class.	There is only one instance of the Identity Object: instance ID 1.

Instance Attributes

Only one instance (instance ID = 1) is implemented in SNAP-PAC devices.

0x01 - Identity - Instance Attributes						
Attribute ID	**Access Rule**	**NV**	**Name**	**Data Type**	**Description**	**Semantics**
0x1	GET		Vendor ID	UINT	Identification of vendor by number.	This attribute is set to the vendor number for Opto 22: 83.
0x2	GET		Device Type	UINT	Identification of general type of product.	This attribute is set to the 0. This device conforms to the Generic Device profile.
0x3	GET		Product Code	UINT	Vendor specific product identification code.	The value of this attribute is set to one of the product codes listed in the following table: **Product Name** / **Product Code** SNAP-PAC-EB1: 118 SNAP-PAC-EB2: 116 SNAP-PAC-R1: 122 SNAP-PAC-R2: 120 SNAP-PAC-S1: 124 SNAP-PAC-S2: 110
0x4	GET		Revision	STRUCT OF	Revision of the product this identity object represents.	This attribute is set to the major/minor revision of the firmware installed on the device.
			Major Revision	USINT	Major revision of this product.	

0x01 - Identity - Instance Attributes (Continued)						
Attribute ID	**Access Rule**	**NV**	**Name**	**Data Type**	**Description**	**Semantics**
			Minor Revision	USINT	Minor revision of this product.	
0x5	GET		Status	WORD	Summary status of this device.	
0x6	GET		Serial Number	UDINT	Serial number of device.	
0x7	GET		Product Name	SHORT_STRING	Human readable identification	This attribute is set to the product name.

Common Services

- 0x0E - Get Attribute Single
- 0x05 - Reset
  - Description
    This instance-level service performs the type of requested.
  - Request Service Data Field Parameters

Name	Data Type	Description Of Parameter	
Type	USINT	Type of reset:	
		Value	**Type of Reset**
		0	Emulate power-cycle on SNAP-PAC device.
		1	Return to factory-default configuration, then emulate power-cycle on SNAP PAC device. This operation does not erase IP configuration.

  - Success Response Service Data Field Parameters
    None
- 0x01 - Get Attributes All (instance level only)

Vendor Specific Services

- 0x32 - Store Configuration in Non-volatile Memory
  - Description
    This instance level service stores the current device configuration to non-volatile memory.
  - Request Service Data Field Parameters
    None
  - Success Response Service Data Field Parameters
    None
- 0x33 - Erase Configuration from Non-volatile Memory

- Description
  This instance level service erases configuration information from non-volatile memory.
- Request Service Data Field Parameters
  None
- Success Response Service Data Field Parameter
  None

0x02 - Message Router

This object has no supported attributes.

0x04 - Assembly

The assembly object class binds attributes from multiple objects into the Data attribute. The Data attribute value is a block of data composed of the values of all bound attributes. By accessing the Data attribute of an assembly instance, all bound attributes are accessed as a single block of data. Input or output data may be bound by an assembly object. The terms *input* and *output* are defined from the network's point of view. An input will produce data on the network and an output will consume data from the network. 16 assembly instances are provided (instance IDs 100-115). By default, no attributes are bound (the length of the data attribute value is 0 for all assembly instances). Using the EtherNet/IP Configurator software, the format of the each assembly Data attribute may be configured by specifying the object attributes to be bound and their locations with in the data block.

Changes made to the configuration of an assembly instance at the same an implicit connection is using the assembly instance will not take effect until the implicit connection is closed.

Instance 254 is provided to serve as the output connection point for Input Only connections. The modeless real-time data format with a data length of 0 is used for this instance.

Instance 255 is provided to serve as the output connection point for Listen Only connections. The modeless real-time data format with a data length of 0 is used for this instance.

Class Attributes

0x04 - Assembly - Class Attributes						
Attribute ID	**Access Rule**	**NV**	**Name**	**Data Type**	**Description**	**Semantics**
0x1	GET		Revision	UINT	Revision of this open object	The revision of the Assembly Object class specified by the CIP Common Specification. The current value assigned to this attribute is 2.
0x2	GET		Max Instance	UINT	The maximum instance number for objects of this class.	Default value is 255.

Instance Attributes

0x04 - Assembly - Instance Attributes						
Attribute ID	**Access Rule**	**NV**	**Name**	**Data Type**	**Description**	**Semantics**
0x3	SET	V	Data	ARRAY OF BYTE	All attribute data bound by this assembly instance.	The format of this data can be specified using the EtherNet/IP configurator utility.

Common Services

- 0x0E - Get Attribute Single
- 0x10 - Set Attribute Single

0x06 - Connection Manager

- One instance of this object is implemented, instance ID = 1. This object has no supported attributes.
- Explicit (class 3 transport type) and implicit (or I/O) (class 1 transport type) connections are supported.

Type	Maximum Number of Connections	Maximum Message Size		
		Type	**Read/ Input**	**Write/ Output**
Explicit (Class 3 transport)	32 (16 EtherNet/IP encapsulation sessions)	connected	498	494
		unconnected	500	496
Implicit (Class 1 transport)	16	Read/Input = 500, Write/Output = 496		

- The following implicit (or I/O) connection types are supported:
  - Exclusive Owner
  - Input Only

  - Listen Only
- Implicit (or I/O) messaging (class 1 transport) real-time formats are:

Messaging Direction	Real-Time Format
input (target to originator)	Modeless
output (originator to target)	32-bit Run/Idle Header

Object Specific Services

- 0x4E - Forward Close (instance only)

  Description
  Closes a connection. See The CIP Networks Library, Volume 1, Common Industrial Protocol, section 3-5 for more information.
- 0x54 - Forward Open (instance only)

  Description
  Opens a connection, maximum data size is 511 bytes. See The CIP Networks Library, Volume 1, Common Industrial Protocol, section 3-5 for more information.

0x08 - Discrete Input Point

The Discrete Input Point object class provides access to the input channels of SNAP-PAC digital I/O modules.

Class Attributes

0x08 - Discrete Input Point - Class Attributes						
Attribute ID	**Access Rule**	**NV**	**Name**	**Data Type**	**Description**	**Semantics**
0x1	GET		Revision	UINT	Revision of this open object	The revision of the Discrete Input Point specified by the CIP Common Specification. The current value assigned to this attribute is 2.
0x64	GET		Vendor Specific Revision	UINT	Revision of vendor specific extension to this open object	The revision of our vendor specific extensions to the open Discrete Input Point specified by the CIP Common Specification. The current value assigned to this attribute is 1.

Instance Attributes

Each channel of I/O can be uniquely identified by channel number and slot number. The channel number can range from 0 through 31, and is unique for each channel on a particular module. The module slot number can range from 0 through 15 and is printed next to each module slot on the I/O

rack. The instance id of the Discrete Input Point object associated with a given channel number and slot is determined through the following equations.

The following equations provide a means to determine the Instance Id for the Discrete Input object that represents a digital input channel in a given slot (or vice versa). The slot number can range from 0 - 15. The channel number can range from 0 - 31.

Instance Id = 1 + (Module Slot Number)*64 + (Module Channel Number)

Module Slot Number = RoundDown((Instance Id - 1)/64)

Channel Number = (Instance Id - 1) % 64

0x08 - Discrete Input Point - Instance Attributes						
Attribute ID	**Access Rule**	**NV**	**Name**	**Data Type**	**Description**	**Semantics**
0x1	GET		Number Of Attributes	USINT	Number supported in this product	Default value is 16.
0x2	GET		Attribute List	ARRAY OF USINT	List of attributes supported in this product	
0x3	GET		Value	BOOL	Input Point Value	0 = Off 1 = On
0x64	GET		Module (Slot) Number	UINT	The module slot associated with this I/O point.	The Module Slot Number identifies the slot into which the module associated with this I/O point is installed. The first slot (nearest the brain) is assigned Module Slot Number 0. The module slot number is also related to the instance id for this object as follows: Module Slot Number = RoundDown((Instance Id-1)/64)
0x65	GET		Module Channel Number	UINT	The module channel associated with this I/O point.	The first channel on a module is channel number 0. The module channel number is related to the instance id for this object as follows: Module Channel Number = (Instance Id - 1) % 64
0x66	GET		Module Path	STRUCT OF	EPATH for the Module Configuration object associated with this I/O point. This path contains one logical class segment and one logical instance segment.	
			Path Size	UINT	Length (in octets) of the Packed EPATH contained in the Path.	

<table>
<tr><th colspan="7">0x08 - Discrete Input Point - Instance Attributes (Continued)</th></tr>
<tr><th>Attribute ID</th><th>Access Rule</th><th>NV</th><th>Name</th><th>Data Type</th><th>Description</th><th>Semantics</th></tr>
<tr><td></td><td></td><td></td><td>Path</td><td>PACKED EPATH</td><td>EPATH for the Module Configuration object associated with this I/O point. This path contains one logical class segment and one logical instance segment.</td><td></td></tr>
<tr><td>0x67</td><td>SET</td><td>NVS</td><td>Point Name</td><td>STRING</td><td>Name for this point</td><td>Maximum length of string is 50 characters. The response to attempts to write a name length longer than 50 characters will contain general status code 0x15 (Too much data.)</td></tr>
<tr><td>0x68</td><td>GET</td><td></td><td>Module Type</td><td>UINT</td><td>Read only access to the Module Type attribute for the module associated with this point.</td><td>Possible module types for discrete input points are:<table><tr><th>Module Type</th><th>Description</th></tr><tr><td>0x00</td><td>Standard digital module.</td></tr><tr><td>0xE0</td><td>SNAP-IDC-32</td></tr><tr><td>0xE3</td><td>SNAP-IAC-A-16</td></tr><tr><td>0xE4</td><td>SNAP-IAC-16</td></tr><tr><td>0xE5</td><td>SNAP-IDC-16</td></tr></table></td></tr>
<tr><td>0x69</td><td>SET</td><td>NVS</td><td>Point Type</td><td>UINT</td><td>The point type and associated module type determine the range and the engineering units used by a point.</td><td>The point type is 0x100.</td></tr>
<tr><td>0x80</td><td>SET</td><td>NVS</td><td>Feature Select</td><td>UDINT</td><td>Selects the feature to enable for this point.</td><td>If a feature is selected that is not supported by a particular point type, a general status error 0x20 - Invalid Parameter is returned. For possible features see “Features for 0x08 - Discrete Input Point, Attribute 0x80” on page 19.</td></tr>
<tr><td>0x81</td><td>SET</td><td>NVS</td><td>Feature Enable</td><td>BOOL</td><td>Enables the selected feature.</td><td>0 = Selected feature is not enabled. See Feature Select Semantics for further details.
1 = Selected feature is enabled. See Feature Select Semantics for further details.</td></tr>
<tr><td>0x82</td><td>GET</td><td></td><td>Feature Value Type</td><td>USINT</td><td>Specifies data type of the Feature Value attribute.</td><td>Specifies the data type of the Feature Value attribute.
Default value is 0xC8 - UDINT
See Feature Select Semantics for Feature Value Type for a given Feature Select.
Possible values:
0xC4 DINT
0xC8 UDINT</td></tr>
</table>

<table>
<tr><th colspan="7">0x08 - Discrete Input Point - Instance Attributes (Continued)</th></tr>
<tr><th>Attribute ID</th><th>Access Rule</th><th>NV</th><th>Name</th><th>Data Type</th><th>Description</th><th>Semantics</th></tr>
<tr><td>0x83</td><td>GET</td><td></td><td>Feature Value</td><td>Depends on value of feature value type attribute 0X82</td><td>Value of selected feature.</td><td>The value read from this attribute depends on the value of the Clear Feature Value attribute:<table><tr><th>Clear Feature Value</th><th>Feature Value</th></tr><tr><td>0</td><td>Is equal to the current feature value.</td></tr><tr><td>1</td><td>Is equal to the feature value latched when the Clear Feature Value attribute was changed from 0 to 1.</td></tr></table></td></tr>
<tr><td>0x84</td><td>SET</td><td>V</td><td>Clear Feature Value</td><td>BOOL</td><td></td><td>The effect of writing a value to this attribute depends on the previous value:<table><tr><th>Previous Value</th><th>Next Value</th><th>Behavior</th></tr><tr><td>0</td><td>0</td><td>Is equal to the current feature value.</td></tr><tr><td>0</td><td>1</td><td>Is equal to the feature value latched when the Clear Feature Value attribute was changed from 0 to 1.</td></tr><tr><td>1</td><td>1</td><td>No effect. The feature value latched when Clear Feature Value attribute was changed from 0 to 1 is returned in response to Feature Value reads.</td></tr><tr><td>1</td><td>0</td><td>The current feature value can now be read from the Feature Value attribute.</td></tr></table></td></tr>
</table>

<table>
<tr><th colspan="7">0x08 - Discrete Input Point - Instance Attributes (Continued)</th></tr>
<tr><th>Attribute ID</th><th>Access Rule</th><th>NV</th><th>Name</th><th>Data Type</th><th>Description</th><th>Semantics</th></tr>
<tr><td>0x85</td><td>SET</td><td>V</td><td>On Latch</td><td>BOOL</td><td>Latches any off to on transition.</td><td>Get 0 = no transition latched
Get 1 = transition latched
Writes to this attribute may be used to clear the latch state via implicit messaging. (When communicating via explicit messaging, the Read And Clear service should be used to clear the latch state.) To prevent inadvertently clearing the latch multiple times, the clear latch operation is edge-sensitive with respect to the value written to this attribute, and the value read from this attribute is always zero after a 1 has been written to it. This provides positive indication to the scanner of completion of the clear latch operation. When the scanner subsequently writes a 0 to this attribute, the actual latch state can be read from this attribute (including any edge latched since the clear latch operation was initiated.)
A typical clear latch operation via implicit messaging consists of the following steps:
1. Set this attribute to 1.
2. Wait until 0 is read from this attribute.
3. Set this attribute to 0.
The following table describes the interaction between the value written to this attribute and the value read from this attribute.
<table>
<tr><th>Write Value</th><th>Previous Write Value</th><th>Behavior</th><th>Subsequent Read Value</th></tr>
<tr><td>0</td><td>0</td><td>Has no effect.</td><td>latch state</td></tr>
<tr><td>1</td><td>0</td><td>Clear the latch state.</td><td>0</td></tr>
<tr><td>1</td><td>1</td><td>Has no effect. A subsequent read from this attribute returns 0, though the actual latch is monitoring the input signal for latch edges.</td><td>0</td></tr>
<tr><td>0</td><td>1</td><td>Subsequent reads of this attribute return the latch state.</td><td>latch state</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><th colspan="7">0x08 - Discrete Input Point - Instance Attributes (Continued)</th></tr>
<tr><th>Attribute ID</th><th>Access Rule</th><th>NV</th><th>Name</th><th>Data Type</th><th>Description</th><th>Semantics</th></tr>
<tr><td>0x86</td><td>SET</td><td>V</td><td>On Latch</td><td>BOOL</td><td>Latches any on to off transition.</td><td>Get 0 = no transition latched
Get 1 = transition latched
Writes to this attribute may be used to clear the latch state via implicit messaging. (When communicating via explicit messaging, the Read And Clear service should be used to clear the latch state.) To prevent inadvertently clearing the latch multiple times, the clear latch operation is edge-sensitive with respect to the value written to this attribute, and the value read from this attribute is always zero after a 1 has been written to it. This provides positive indication to the scanner of completion of the clear latch operation. When the scanner subsequently writes a 0 to this attribute, the actual latch state can be read from this attribute (including any edge latched since the clear latch operation was initiated.)
A typical clear latch operation via implicit messaging consists of the following steps:
1. Set this attribute to 1.
2. Wait until 0 is read from this attribute.
3. Set this attribute to 0.
The following table describes the interaction between the value written to this attribute and the value read from this attribute.
<table>
<tr><th>Write Value</th><th>Previous Write Value</th><th>Behavior</th><th>Subsequent Read Value</th></tr>
<tr><td>0</td><td>0</td><td>Has no effect.</td><td>latch state</td></tr>
<tr><td>1</td><td>0</td><td>Clear the latch state.</td><td>0</td></tr>
<tr><td>1</td><td>1</td><td>Has no effect. A subsequent read from this attribute returns 0, though the actual latch is monitoring the input signal for latch edges.</td><td>0</td></tr>
<tr><td>0</td><td>1</td><td>Subsequent reads of this attribute return the latch state.</td><td>latch state</td></tr>
</table>
</td></tr>
</table>

Features for 0x08 - Discrete Input Point, Attribute 0x80

0x08 - Discrete Input Point, Attribute 0x80 Possible Features					
Feature Select Code	**Constraints**	**Feature Enable**	**Feature Value Type**	**Feature Value**	**Clear Feature Value**
0 = No feature enabled	This feature can always be selected.	Has no effect.	0xC8 UDINT	is zero	has no effect
1 = Off to On edge counter.	This feature can be selected for any discrete input point associated with a low density (<= 4 channel) digital module. Also currently limited to units that support Advanced Digital Features (SNAP-PAC-R1, SNAP-PAC-EB1, etc.)	0 = Feature value does not increment. 1 = Feature Value is incremented for each off to on transition of the Value attribute. When Feature Select is initially set to 1, the Feature Enable attribute is automatically set to 1.	0xC8 UDINT	Number of off to on transitions in the Value attribute while Feature Enable is set to 1.	Clears the Feature Value attribute according to the semantics described in the Clear Feature Value Attribute.
2 = Total Time On	This feature can only be selected for Discrete Input or Discrete Output points associated with low density (<= 4 channels) digital modules.	Has no effect. Always 0.	0xC8 UDINT	Total time the Value attribute is on in units of 100 microseconds (usec.)	Clears the Feature Value attribute according to the semantics described in the Clear Feature Value Attribute.
4 = Quadrature Counter	This feature can only be selected for Advanced Digital Input points. Two adjacent digital input channels are required to provide quadrature counts. Rising edges on the A or even channel increase the quadrature count. Rising edges on the B or odd channel decrement the quadrature count. The B channel is always the next instance above the A channel.	Is automatically set to 1 when this feature is selected. Feature Enable must remain 1 for this feature to work properly.	0xC4 DINT	Quadrature counts. (Can be read on the A channel or the B channel)	Clears the Feature Value attribute according to the semantics described in the Clear Feature Value Attribute.
8 = Continuous Frequency Measurement, no auto-zero	This feature can only be selected for Advanced Digital Input or Output points	Has no effect. Always 0.	0xC8 UDINT	Continuous frequency measurement of Value measured as the number of rising edges observed per second.	Clears the Feature Value attribute according to the semantics described in the Clear Feature Value Attribute.

0x08 - Discrete Input Point, Attribute 0x80 Possible Features (Continued)					
Feature Select Code	**Constraints**	**Feature Enable**	**Feature Value Type**	**Feature Value**	**Clear Feature Value**
9 = Pulse On Measurement	This feature can only be selected for Advanced Digital Input or Output points.	Has no effect. Always 0.	0xC8 UDINT	Pulse measurement is armed when Feature Select is set to = 9. On pulse measurement begins with the first off to on transition in the Value attribute after Feature Select is set = 9. The subsequent on to off transition in the Value attribute completes the on pulse measurement. When the pulse measurement has completed as determined by Feature Select = 0, Feature Value contains the pulse length in units of 100 microseconds (usec.) During measurement (Feature Select = 9) this attribute displays the current value of the pulse measurement counter in units of 100 usec.	Does not clear the feature value attribute. However, the other semantics described in the Clear Feature Value Attribute still hold.
10 = Pulse Off Measurement	This feature can only be selected for Advanced Digital Input or Output points. Measurement is armed when Feature Select is set to = 10. Off pulse measurement begins with the first on to off transition in the Value attribute after Feature Select is set = 10. The subsequent off to on transition in the Value attribute completes the off pulse measurement.	Has no effect. Always 0.	0xC8 UDINT	When the pulse measurement has completed as determined by Feature Select = 0, Feature Value is the pulse length in units of 100 microseconds (usec.) During measurement (Feature Select = 9) this attribute displays the current value of the pulse measurement counter in units of 100 usec.	Does not clear the feature value attribute. However, the other semantics described in the Clear Feature Value Attribute still hold.

0x08 - Discrete Input Point, Attribute 0x80 Possible Features (Continued)					
Feature Select Code	**Constraints**	**Feature Enable**	**Feature Value Type**	**Feature Value**	**Clear Feature Value**
11 = One-time Period Measurement	This feature can only be selected for Advanced Digital Input or Output points. The resolution of this measurement is limited to 100usec. It can not be used to measure frequencies greater than 10000Hz (periods < 100usec.) The accuracy of this measurement is dependant on the signal frequency: Max Error = 100% * 0.0001s * Frequency. (Please see turn on/turn off times in the Digital Input Module data sheet to determine if the module places any additional constraints on the frequency of the signal measured.) Measurement is armed when Feature Select is set to = 11. Measurement begins with the first transition in the Value attribute after Feature Select is set = 11. The subsequent transition of the same type of the Value attribute completes the measurement. Feature Select is set to 0 when the measurement is completed.	Has no effect. Always 0.	0xC8 UDINT	When the measurement has completed as determined by Feature Select = 0, Feature Value is the measured period in units of 100 microseconds (usec.) During measurement (Feature Select = 11) this attribute displays the current value of the measurement counter in units of 100 usec.	Does not clear the feature value attribute. However, the other semantics described in the Clear Feature Value Attribute still hold.

0x08 - Discrete Input Point, Attribute 0x80 Possible Features (Continued)					
Feature Select Code	**Constraints**	**Feature Enable**	**Feature Value Type**	**Feature Value**	**Clear Feature Value**
12 = One-time Frequency Measurement	This feature can only be selected for Advanced Digital Input or Output points. This measurement counts the number of cycles that occur during a 1 second measurement window. Signals slower than 1Hz can not be measured. The accuracy of this measurement is dependant on the signal frequency: Max Error = 100% 1Hz/Frequency. (Please see turn on/turn off times in the Digital Input Module data sheet to determine if the module places any additional constraints on the frequency of the signal measured.) Measurement is armed when Feature Select is set to = 12. Feature Enable is automatically set to = 1 when measurement is armed.	Is automatically set to 1 when measurement is armed. Is automatically set to 0 when measurement has completed. Do not set this attribute during measurement.	0xC8 UDINT	When the measurement has completed as determined by Feature Enable = 0, or Feature Select = 0, Feature Value is the measured frequency in units of Hz. During measurement (Feature Enable = 1 or Feature Select = 12) this attribute displays the current value of the measurement counter in units of Hz.	Does not clear the feature value attribute. However, the other semantics described in the Clear Feature Value Attribute still hold.
18 = Total Time Off	This feature can only be selected for Discrete Input or Discrete Output points associated with low density (<= 4 channels) digital modules.	Has no effect.	0xC8 UDINT	Total time the Value attribute is off in units of 100 microseconds (usec.)	Set 0 = no effect Set 1 = clear Feature Value, setting it to zero. Get always returns 0.

0x08 - Discrete Input Point, Attribute 0x80 Possible Features (Continued)					
Feature Select Code	**Constraints**	**Feature Enable**	**Feature Value Type**	**Feature Value**	**Clear Feature Value**
0xXXYY0004 = Quadrature Counter, with Index, XX = Index point (see constraints), YY= index point edge.	Two adjacent Advanced Digital Input points are required to provide quadrature counts and a third Advanced Digital Input point is specified to provide index functionality. The A point always has an odd instance id and the instance id of the B point is always equal to the instance id of the A point plus one. Rising edges on the A point increase the quadrature count. Rising edges on the B point decrement the quadrature count. When setting the Feature Select attribute for the A or B points, the Feature Select attribute for the remaining (B or A) point is automatically set to the same value. The Index point is specified in bits 24-31 of the Feature Select attribute for the A and B points. This 8 bit value specifies the index channel point using the following formula: bits 24-31 = (Index point instance Id-1)/4 + ((Index point instance Id - 1) %4). The index point may not be the same as the A point or the B point. The Feature Select attribute for the index point will automatically be set to 0xWWYY0041, where WW specifies the instance id of the A point according to the following formula: 1 + (64*(WW/4)) + WW%4.	Is automatically set to 1 when this feature is selected. Feature Enable must remain 1 for this feature to work properly.	0xC4 DINT	A point => Quadrature counts. B point => Quadrature counts since last index edge was detected. Index point => A quadrature counts when last index edge was detected.	Clears the Feature Value attribute according to the semantics described in the Clear Feature Value Attribute.

0x08 - Discrete Input Point, Attribute 0x80 Possible Features (Continued)					
Feature Select Code	**Constraints**	**Feature Enable**	**Feature Value Type**	**Feature Value**	**Clear Feature Value**
0xXXYY0004 = Quadrature Counter (continued)	The index edge is specified in Feature Select bits 16-23, where 01 = rising edge, and 02 = falling edge. When an index edge of the type specified is detected, the Feature Value of the index point is set to the Feature Value for the A point, and the Feature Value for the B point is set to 0. That is, during operation quadrature counts for A point = quadrature counts for B point + Feature Value for index point.				
0xWWYY0041 = Quadrature index point. DO NOT CONFIGURE THIS VALUE. Included for specification/documentation purposes only. This value is automatically configured on the index point when the A point or B point of a quadrature input pair is configured. WW = specifies A point (see constraints). YY= specifies the index edge.	Two adjacent Advanced Digital Input points are required to provide quadrature counts and a third Advanced Digital Input point is specified to provide index functionality. The A point always has an odd instance id and the instance id of the B point is always equal to the instance id of the A point plus one. Rising edges on the A point increase the quadrature count. Rising edges on the B point decrement the quadrature count. When setting the Feature Select attribute for the A or B points, the Feature Select attribute for the remaining (B or A) point is automatically set to the same value.	Is automatically set to 1 when this feature is selected. Feature Enable must remain 1 for this feature to work properly.	0xC4 DINT	A point => Quadrature counts. B point => Quadrature counts since last index edge was detected. Index point => A quadrature counts when last index edge was detected.	Clears the Feature Value attribute according to the semantics described in the Clear Feature Value Attribute.

0x08 - Discrete Input Point, Attribute 0x80 Possible Features (Continued)					
Feature Select Code	**Constraints**	**Feature Enable**	**Feature Value Type**	**Feature Value**	**Clear Feature Value**
0xWWYY0041 = Quadrature index point (continued)	The Index point is specified in bits 24-31 of the Feature Select attribute for the A and B points. This 8 bit value specifies the index channel point using the following formula: bits 24-31 = (Index point instance Id-1)/4 + ((Index point instance Id - 1) %4). The index point may not be the same as the A point or the B point. The Feature Select attribute for the index point will automatically be set to 0xWWYY0041, where WW specifies the instance id of the A point according to the following formula: 1 + (64*(WW/4)) + WW%4. The index edge is specified in Feature Select bits 16-23, where 01 = rising edge, and 02 = falling edge. When an index edge of the type specified is detected, the Feature Value of the index point is set to the Feature Value for the A point, and the Feature Value for the B point is set to 0. That is, during operation quadrature counts for A point = quadrature counts for B point + Feature Value for index point.				

Common Services

- 0x0E - Get Attribute Single
- 0x10 - Set Attribute Single

Vendor Specific Services

- 0x32 - Read And Clear Attribute

  Description
  Returns the contents of the specified attribute and then sets the contents to 0. It may be used with the On Latch, Off Latch, and Feature Value attributes.

  Request Service Data Field Parameters
  None

  Success Response Service Data Field Parameters

Name	Data Type	Description Of Parameter
Attribute Data	Object/class attribute specific structure	Contains the requested attribute data

0x09 - Discrete Output Point

The Discrete Output Point object class provides access to the output channels of SNAP-PAC digital I/O modules.

Class Attributes

0x09 - Discrete Output Point - Class Attributes						
Attribute ID	**Access Rule**	**NV**	**Name**	**Data Type**	**Description**	**Semantics**
0x1	GET		Revision	UINT	Revision of this open object	The revision of the Discrete Output Point specified by the CIP Common Specification. The current value assigned to this attribute is 2.
0x64	GET		Vendor Specific Revision	UINT	Revision of vendor specific extension to this open object	The revision of our vendor specific extensions to the open Discrete Output Point specified by the CIP Common Specification. The current value assigned to this attribute is 1.

Instance Attributes

Each channel of I/O can be uniquely identified by channel number and slot number. The channel number can range from 0 through 31, and is unique for each channel on a particular module. The module slot number can range from 0 through 15 and is printed next to each module slot on the I/O rack. The instance id of the Discrete Output Point object associated with a given channel number and slot is determined through the following equations:

Instance Id = 1 + (Module Slot Number)*64 + (Module Channel Number)
Module Slot Number = RoundDown((Instance Id - 1)/64)
Channel Number = (Instance Id - 1) % 64

0x09 - Discrete Output Point - Instance Attributes						
Attribute ID	**Access Rule**	**NV**	**Name**	**Data Type**	**Description**	**Semantics**
0x1	GET		Number Of Attributes	USINT	Number supported in this product	Default value is 20.
0x2	GET		Attribute List	ARRAY OF USINT	List of attributes supported in this product	
0x3	SET		Value	BOOL	Value sent to the output point	0 = Off 1 = On
0x5	GET		Fault Action	BOOL	Action taken on output's value in recoverable fault state. The recoverable fault state is entered when a connection to the Value attribute times out.	On SNAP-PAC devices, the default behavior is to force the output into the off state (0). If the Comm Watchdog feature is enabled, the Fault Action functionality is disabled.
0x6	GET		Fault Value	BOOL	Value written to the Value attribute when this object enters the recoverable fault state.	Fixed at 0.
0x7	GET		Idle Action	BOOL	Action taken on output's value in idle state. The idle state is entered when a zero 32 bit run/idle header is received on a connection to the Value attribute for this object.	On SNAP-PAC devices, the default behavior is to force the output into the off state (0). If the Comm Watchdog feature is enabled, the Idle Action functionality is disabled.
0x8	GET		Idle Value	BOOL	Value written to the Value attribute when this object enters the idle state.	Fixed at 0.
0x64	GET		Module (Slot) Number	UINT	The module slot associated with this I/O point.	The Module Slot Number identifies the slot into which the module associated with this I/O point is installed. The first slot (nearest the brain) is assigned Module Slot Number 0. The module slot number is also related to the instance id for this object as follows: Module Slot Number = RoundDown((Instance Id-1)/64)
0x65	GET		Module Channel Number	UINT	The module channel associated with this I/O point.	The first channel on a module is channel number 0. The module channel number is related to the instance id for this object as follows: Module Channel Number = (Instance Id - 1) % 64

<table>
<tr><th colspan="7">0x09 - Discrete Output Point - Instance Attributes (Continued)</th></tr>
<tr><th>Attribute ID</th><th>Access Rule</th><th>NV</th><th>Name</th><th>Data Type</th><th>Description</th><th>Semantics</th></tr>
<tr><td rowspan="3">0x66</td><td rowspan="3">GET</td><td rowspan="3"></td><td>Module Path</td><td>STRUCT OF</td><td>EPATH for the Module Configuration object associated with this I/O point. This path contains one logical class segment and one logical instance segment.</td><td>Path Size</td></tr>
<tr><td>Path Size</td><td>UINT</td><td>Length (in octets) of the Packed EPATH contained in the Path.</td><td></td></tr>
<tr><td>Path</td><td>PACKED EPATH</td><td>EPATH for the Module Configuration object associated with this I/O point. This path contains one logical class segment and one logical instance segment.</td><td></td></tr>
<tr><td>0x67</td><td>SET</td><td>NVS</td><td>Point Name</td><td>STRING</td><td>Name for this point</td><td>Maximum length of string is 50 characters. The response to attempts to write a name length longer than 50 characters will contain general status code 0x15 (Too much data.)</td></tr>
<tr><td>0x68</td><td>GET</td><td></td><td>Module Type</td><td>UINT</td><td>Read only access to the Module Type attribute for the module associated with this point.</td><td>See the Module Type attribute of the Module Slot Configuration class for more information.<table><tr><th>Module Type</th><th>Description</th></tr><tr><td>0x00</td><td>Standard digital module.</td></tr><tr><td>0xE1</td><td>SNAP-ODC-32-SRC</td></tr><tr><td>0xE2</td><td>SNAP-ODC-32-SNK</td></tr></table></td></tr>
<tr><td>0x69</td><td>SET</td><td>NVS</td><td>Point Type</td><td>UINT</td><td>The point type and associated module type determine the range and the engineering units used by a point.</td><td>The point type is 0x180.</td></tr>
<tr><td>0x80</td><td>SET</td><td>NVS</td><td>Feature Select</td><td>UDINT</td><td>Selects the feature to enable for this point.</td><td>Currently there are no features defined for discrete outputs and this value can not be changed from 0. However, the feature fields are included in the point object to accommodate future feature additions. The feature Value Type is Always 0xC8 UDINT. The Feature Value is Always 0.</td></tr>
<tr><td>0x81</td><td>SET</td><td>NVS</td><td>Feature Enable</td><td>BOOL</td><td>Enables the selected feature.</td><td>0 = Selected feature is not enabled. See Feature Select Semantics for further details.
1 = Selected feature is enabled. See Feature Select Semantics for further details.</td></tr>
</table>

<table>
<tr><th colspan="7">0x09 - Discrete Output Point - Instance Attributes (Continued)</th></tr>
<tr><th>Attribute ID</th><th>Access Rule</th><th>NV</th><th>Name</th><th>Data Type</th><th>Description</th><th>Semantics</th></tr>
<tr><td>0x82</td><td>GET</td><td></td><td>Feature Value Type</td><td>USINT</td><td>Specifies data type of the Feature Value attribute.</td><td>Specifies the data type of the Feature Value attribute. Default value is 0xC8 - UDINT
See Feature Select Semantics for Feature Value Type for a given Feature Select.</td></tr>
<tr><td>0x83</td><td>GET</td><td></td><td>Feature Value</td><td>Depends on value of feature value type attribute 0X82</td><td>Value of selected feature.</td><td>The value read from this attribute depends on the value of the Clear Feature Value attribute:<table><tr><th>Clear Feature Value</th><th>Feature Value</th></tr><tr><td>0</td><td>Is equal to the current feature value.</td></tr><tr><td>1</td><td>Is equal to the feature value latched when the Clear Feature Value attribute was changed from 0 to 1.</td></tr></table></td></tr>
<tr><td>0x84</td><td>SET</td><td>V</td><td>Clear Feature Value</td><td>BOOL</td><td>Sets the Feature Value to zero. Provides a method by which the Feature Value can be cleared via implicit messaging.</td><td>The effect of writing a value to this attribute depends on the previous value:<table><tr><th>Previous Value</th><th>Next Value</th><th>Behavior</th></tr><tr><td>0</td><td>0</td><td>No effect. The current feature value can be read from the Feature Value attribute.</td></tr><tr><td>0</td><td>1</td><td>Latch current Feature Value and clear Feature Value. The feature value read here will be latched and returned in response to Feature Value reads as long as the Clear Feature Value attribute remains set to 1.</td></tr><tr><td>1</td><td>1</td><td>No effect. The feature value latched when Clear Feature Value attribute was changed from 0 to 1 is returned in response to Feature Value reads.</td></tr><tr><td>1</td><td>0</td><td>The current feature value can now be read from the Feature Value attribute.</td></tr></table></td></tr>
<tr><td>0x87</td><td>SET</td><td>NVS</td><td>Comm Watchdog Enable</td><td>BOOL</td><td>Enable/Disable communication watchdog operation. The communication watchdog value is written to the output point when no commands have been received for the amount of time configured in the Communication Watchdog object.</td><td>0 = Disable Comm watchdog operation for this point (default)
1= Enable Comm Watchdog operation for this point.\
if this feature is enabled, the Fault Action & Idle Action functionality is disabled.</td></tr>
</table>

0x09 - Discrete Output Point - Instance Attributes (Continued)						
Attribute ID	**Access Rule**	**NV**	**Name**	**Data Type**	**Description**	**Semantics**
0x88	SET	NVS	Comm Watch-dog Value	BOOL	Value written to the Value attribute if the Comm Watchdog Enable attribute =1 and a Comm Watch-dog timeout occurs.	0 = Turn point off if a comm watchdog timeout occurs. 1 = Turn point on if a comm watchdog timeout occurs.

Common Services

0x0E - Get Attribute Single

0x10 - Set Attribute Single

Vendor Specific Services

0x32 - Read And Clear Attribute

- Description

  Returns the contents of the specified attribute and then sets the contents to 0. It may be used with the Feature Value attribute.

- Request Service Data Field Parameters

  None

- Success Response Service Data Field Parameters

Name	Data Type	Description Of Parameter
Attribute Data	Object/class attribute specific structure	Contains the requested attribute data

0x0A - Analog Input Point

The Analog Input Point class provides access to the input channels of SNAP-PAC analog I/O modules.

Class Attributes

0x0A - Analog Input Point - Class Attributes						
Attribute ID	**Access Rule**	**NV**	**Name**	**Data Type**	**Description**	**Semantics**
0x1	GET		Revision	UINT	Revision of this open object	The revision of the Analog Input Point specified by the CIP Common Specification. The current value assigned to this attribute is 2.
0x64	GET		Vendor Specific Revision	UINT	Revision of vendor specific extension to this open object	The revision of our vendor specific extensions to the open Analog Input Point specified by the CIP Common Specification. The current value assigned to this attribute is 1.

Instance Attributes

Each channel of I/O can be uniquely identified by channel number and slot number. The channel number can range from 0 through 31, and is unique for each channel on a particular module. The module slot number can range from 0 through 15 and is printed next to each module slot on the I/O rack. The instance id of the Analog Input Point object associated with a given channel number and slot is determined through the following equations:

Instance Id = 1 + (Module Slot Number)*64 + (Module Channel Number)

Module Slot Number = RoundDown((Instance Id - 1)/64)

Channel Number = (Instance Id - 1) % 64

0x0A - Analog Input Point - Instance Attributes						
Attribute ID	**Access Rule**	**NV**	**Name**	**Data Type**	**Description**	**Semantics**
0x1	GET		Number Of Attributes	USINT	Number supported in this product	Default value is 22.
0x2	GET		Attribute List	ARRAY OF USINT	List of attributes supported in this product	
0x3	GET		Value	INT	Analog input value. See Semantics for representation details. For analog input value in REAL format, see rValue attribute.	See “Analog Input: Value Attribute Semantics” on page 38.

0x0A - Analog Input Point - Instance Attributes (Continued)						
Attribute ID	**Access Rule**	**NV**	**Name**	**Data Type**	**Description**	**Semantics**
0x64	GET		Module (Slot) Number	UINT	The module slot associated with this I/O point.	The Module Slot Number identifies the slot into which the module associated with this I/O point is installed. The first slot (nearest the brain) is assigned Module Slot Number 0. The module slot number is also related to the instance id for this object as follows: Module Slot Number = RoundDown((Instance Id-1)/64)
0x65	GET		Module Channel Number	UINT	The module channel associated with this I/O point.	The first channel on a module is channel number 0. The module channel number is related to the instance id for this object as follows: Module Channel Number = (Instance Id - 1) % 64
0x66	GET		Module Path	STRUCT OF	EPATH for the Module Configuration object associated with this I/O point. This path contains one logical class segment and one logical instance segment.	
			Path Size	UINT	Length (in octets) of the Packed EPATH contained in the Path.	
			Path	PACKED EPATH	EPATH for the Module Configuration object associated with this I/O point. This path contains one logical class segment and one logical instance segment.	
0x67	SET	NVS	Point Name	STRING	Name for this point	Maximum length of string is 50 characters. The response to attempts to write a name length longer than 50 characters will contain general status code 0x15 (Too much data.)
0x68	GET		Module Type	UINT	Read only access to the Module Type attribute for the module associated with this point.	
0x69	SET	NVS	Point Type	UINT	The point type and associated module type determine the range and the engineering units used by a point.	
0x80	SET	NVS	Feature Select	UDINT	Selects the feature to enable for this point.	Currently there are no features defined for analog inputs and this value can not be changed from 0. However, the feature fields are included in the point object to accommodate future feature additions. The feature Value Type is Always 0xC8 UDINT. The Feature Value is Always 0.

<table>
<tr><th colspan="7">0x0A - Analog Input Point - Instance Attributes (Continued)</th></tr>
<tr><th>Attribute ID</th><th>Access Rule</th><th>NV</th><th>Name</th><th>Data Type</th><th>Description</th><th>Semantics</th></tr>
<tr><td>0x81</td><td>SET</td><td>NVS</td><td>Feature Enable</td><td>BOOL</td><td>Enables the selected feature.</td><td>0 = Selected feature is not enabled. See Feature Select Semantics for further details.
1 = Selected feature is enabled. See Feature Select Semantics for further details.</td></tr>
<tr><td>0x82</td><td>GET</td><td></td><td>Feature Value Type</td><td>USINT</td><td>Specifies data type of the Feature Value attribute.</td><td>Specifies the data type of the Feature Value attribute. Default value is 0xC8 - UDINT See Feature Select Semantics for Feature Value Type for a given Feature Select.</td></tr>
<tr><td>0x83</td><td>GET</td><td></td><td>Feature Value</td><td>Depends on value of feature value type attribute 130</td><td>Value of selected feature.</td><td>The value read from this attribute depends on the value of the Clear Feature Value attribute:.
<table>
<tr><th>Clear Feature Value</th><th>Feature Value</th></tr>
<tr><td>0</td><td>Is equal to the current feature value.</td></tr>
<tr><td>1</td><td>Is equal to the feature value latched when the Clear Feature Value attribute was changed from 0 to 1.</td></tr>
</table></td></tr>
<tr><td>0x84</td><td>SET</td><td></td><td>Clear Feature Value</td><td>BOOL</td><td>Sets the Feature Value to zero. Provides a method by which the Feature Value can be cleared via implicit messaging.</td><td>The effect of writing a value to this attribute depends on the previous value:.
<table>
<tr><th>Previous Value</th><th>Next Value</th><th>Behavior</th></tr>
<tr><td>0</td><td>0</td><td>No effect. The current feature value can be read from the Feature Value attribute.</td></tr>
<tr><td>0</td><td>1</td><td>Latch current Feature Value and clear Feature Value. The feature value read here will be latched and returned in response to Feature Value reads as long as the Clear Feature Value attribute remains set to 1.</td></tr>
<tr><td>1</td><td>1</td><td>No effect. The feature value latched when Clear Feature Value attribute was changed from 0 to 1 is returned in response to Feature Value reads.</td></tr>
<tr><td>1</td><td>0</td><td>The current feature value can now be read from the Feature Value attribute.</td></tr>
</table></td></tr>
</table>

<table>
<tr><th colspan="7">0x0A - Analog Input Point - Instance Attributes (Continued)</th></tr>
<tr><th>Attribute ID</th><th>Access Rule</th><th>NV</th><th>Name</th><th>Data Type</th><th>Description</th><th>Semantics</th></tr>
<tr><td>0x85</td><td>SET</td><td>0</td><td>Minimum rValue</td><td>REAL</td><td>Minimum value of rValue attribute measured.</td><td>The following table describes the interaction between the value written to this attribute and the value read from this attribute.
<table>
<tr><th>Write Value</th><th>Previous Write Value</th><th>Behavior</th><th>Subsequent Read Value</th></tr>
<tr><td>0.0</td><td>0.0</td><td>Has no effect.</td><td>minimum rValue</td></tr>
<tr><td>positive infinity</td><td>0.0</td><td>Clear the minimum rValue</td><td>positive infinity</td></tr>
<tr><td>positive infinity</td><td>positive infinity</td><td>Has no effect. A subsequent read from this attribute returns positive infinity, though the device is monitoring the input signal for minimum rValue.</td><td>positive infinity</td></tr>
<tr><td>0.0</td><td>positive infinity</td><td>Subsequent reads of this attribute return the minimum rValue</td><td>minimum rValue</td></tr>
</table>
Writes to this attribute may be used to clear the latched minimum rValue via implicit messaging. (When communicating via explicit messaging, the Read And Clear service should be used to clear the latched minimum rValue.) To prevent inadvertently clearing the latched value multiple times, the clear latch operation is edge-sensitive with respect to the value written to this attribute, and the value read from this attribute is always positive infinity after a positive infinity has been written to it. This provides positive indication to the scanner of completion of the clear latch operation. When the scanner subsequently writes a 0.0 to this attribute, the actual latched minimum rValue can be read from this attribute. (Note, in RSLogix 5000, positive infinity is denoted 1.$)
A typical clear minimum rValue operation via implicit messaging consists of the following steps:
1. Set this attribute to positive infinity.
2. Wait until positive infinity is read from this attribute.
3. Set this attribute to 0.0.</td></tr>
</table>

<table>
<tr><th colspan="7">0x0A - Analog Input Point - Instance Attributes (Continued)</th></tr>
<tr><th>Attribute ID</th><th>Access Rule</th><th>NV</th><th>Name</th><th>Data Type</th><th>Description</th><th>Semantics</th></tr>
<tr><td rowspan="2">0x86</td><td rowspan="2">SET</td><td rowspan="2">0</td><td rowspan="2">Maximum rValue</td><td rowspan="2">REAL</td><td rowspan="2">Maximum value of rValue attribute measured.</td><td>The following table describes the interaction between the value written to this attribute and the value read from this attribute.
<table>
<tr><th>Write Value</th><th>Previous Write Value</th><th>Behavior</th><th>Subsequent Read Value</th></tr>
<tr><td>0.0</td><td>0.0</td><td>Has no effect.</td><td>maximum rValue</td></tr>
<tr><td>negative infinity</td><td>0.0</td><td>Clear the maximum rValue</td><td>negative infinity</td></tr>
<tr><td>negative infinity</td><td>negative infinity</td><td>Has no effect. A subsequent read from this attribute returns negative infinity, though the device is monitoring the input signal for maximum rValue.</td><td>negative infinity</td></tr>
<tr><td>0.0</td><td>negative infinity</td><td>Subsequent reads of this attribute return the maximum rValue</td><td>maximum rValue</td></tr>
</table></td></tr>
<tr><td>Writes to this attribute may be used to clear the latched maximum rValue via implicit messaging. (When communicating via explicit messaging, the Read And Clear service should be used to clear the latched maximum rValue.) To prevent inadvertently clearing the latched value multiple times, the clear latch operation is edge-sensitive with respect to the value written to this attribute, and the value read from this attribute is always negative infinity after a negative infinity has been written to it. This provides positive indication to the scanner of completion of the clear latch operation. When the scanner subsequently writes a 0.0 to this attribute, the actual latched maximum rValue can be read from this attribute. (Note, in RSLogix 5000, negative infinity is denoted -1.$)
A typical clear maximum rValue operation via implicit messaging consists of the following steps:
1. Set this attribute to negative infinity.
2. Wait until negative infinity is read from this attribute.
3. Set this attribute to 0.0.</td></tr>
<tr><td>0x89</td><td>GET</td><td></td><td>rValue</td><td>REAL</td><td>Analog input value given in engineering units in IEEE floating point format.</td><td>The engineering units used are determined by the input point type. The value provided includes any configured Gain, Offset and Filtering.</td></tr>
</table>

0x0A - Analog Input Point - Instance Attributes (Continued)						
Attribute ID	**Access Rule**	**NV**	**Name**	**Data Type**	**Description**	**Semantics**
0x8A	SET	NVS	Low Scaled Engineering Units	REAL	Bottom of input range in engineering units.	The scale of an analog input point defines the range of values returned, in engineering units (before Offset or Gain is applied.) The default scale of an input point is determined by input point type and can be read from the Low Scaled Engineering Units and the High Scaled Engineering Units attributes. The default scale can be changed by writing to these attributes. If Low Scaled Engineering Units and High Scaled Engineering Units are set to the same value, they will be set to default scale values instead. The scale can not be changed for thermocouple input types.
0x8B	SET	NVS	High Scaled Engineering Units	REAL	Top of input range in engineering units.	
0x8C	SET	NVS	Offset	REAL	Offset applied to measured input value.	Offset is applied to the measured analog input value (after scaling) as follows: rValue = (Gain * measured value) + Offset rValue = (Gain * measured value) + Offset
0x8D	SET	NVS	Gain	REAL	Gain applied to measure input value.	If gain is non-zero, it is applied to the measured analog input value (after scaling) as follows: rValue = (Gain * measure value) + Offset
0x8E	SET	NVS	Averaging Filter Weight	REAL	The weight applied to each sample during averaging. Average filtering is used to smooth analog input signals that are erratic or change suddenly. The formula used for filtering is Y = (X - Y)/W + Y, where Y is the filtered value, X is the new unfiltered value, and W is the filter weight. Filtering is applied to the measured analog value after scaling but before gain and offset are applied.	If Averaging Filter Weight is < 0.5, no average filtering is performed. Average filtering is enabled, if Average Filter Weight is set to >0.5, the rValue will be set to the non-filtered value the next time the analog input point is sampled.

Common Services

0x0E - Get Attribute Single

0x10 - Set Attribute Single

Vendor Specific Services

0x32 - Read And Clear Attribute

- Description

  Returns the contents of the specified attribute and then sets the contents to 0. It may be used with the Minimum rValue, Maximum rValue and Feature Value attributes.

- Request Service Data Field Parameters

  None

- Success Response Service Data Field Parameters

Name	Data Type	Description Of Parameter
Attribute Data	Object/class attribute specific structure	Contains the requested attribute data

Analog Input: Value Attribute Semantics

The Value attribute provides the input point value in proportional counts for linear inputs and is given in engineering units, or (engineering units)/10 for non-linear inputs. Please see the rValue attribute for the input point value in engineering units represented as a REAL data type.

Linear Input Points

For linear input points, the value is given in proportional counts. The following tables can be used to convert proportional counts to engineering units for various linear input points using the following formula:

Engineering Units = (Proportional Counts) *((Engineering Units @ high scale) - (Engineering Units @ low scale))/((25000 - (Counts @ low scale)))

Legend	Low Scale
	High Scale

NOTE: To read temperature with a Resistance Temperature Detector (RTD), use engineering units. This includes the PT100 and other temperature sensor types. It only applies to the SNAP-AIRTD module.

NOTE: For load cell modules, use engineering units only.

For Linear Input Types: ±100V, ±50V,±10V, ±5V, ±1V, ±500mV, ±150mV, ±75mV,±50mV,±25mV,±20mA, ±1mA, ±100%, 15.4541 to -1.4541PH	
Value (decimal)	**Engineering Units**
-32768	out of range
-27500	Low scale + (0.1*Low scale) (e.g. -110V)
-25000	Low scale (e.g. -100V)
0	0
25000	High scale (e.g. +100V)
27500	High scale + (0.1*High scale) (e.g. 110V)

For Linear Input Types: 0-250Vrms,0-10V,0-5V,0-10Arms,0-20mA,0-25000Hz,0-40kOhm, 0-20kOhm,0-10kOhm,0-5kOhm,0-400Ohm (RTD),0-100%, 0-2500W,0-2500VA, -273 to 150°C (ICTD)	
Value (decimal)	**Volts**
-32768	out of range
0	Low scale (e.g. 0Vrms)
25000	High scale (e.g. 250Vrms)
27500	High scale + (0.1*High Scale) (e.g. 275 Vrms)

For Linear Input Types: 4-20mA	
Value (decimal)	**Volts**
-32768	out of range
5000	Low scale (e.g. 4mA)
25000	High scale (e.g. 20mA)
27500	High scale + (0.1*High Scale) (e.g. 21mA)

Non-linear Input points

The value attribute for non-linear input points, such as thermocouple inputs, are linearized by the brain and given in units of degrees or degrees/10. The brain is globally configured to use °C or °F, and uses °C by default. The table below indicates whether the Value attribute for a particular thermocouple input type is given in degrees or degrees/10.

For Non-Linear Thermocouple Input Types: Value is given in units of 0.1 °C or °C (0.1°F or °F, if brain is configured for °F)									
Thermocouple Type	**Temperature Range**				**Units given in Value attribute**	**Value Attribute Range**			
	°C		**°F**			**°C (or °C/10)**		**°F (or °F/10)**	
B	42	1820	107.6	3308	degrees	42	1820	108	3308
C	0	2320	32	4208	degrees	0	2320	32	4208
E	-270	1000	-454	1832	degrees/10	-2700	10000	-4540	18320
J	-210	1200	-346	2192	degrees/10	-2100	12000	-3460	21920
K	-270	1372	-454	2501	degrees/10	-2700	13720	-4540	25010
N	-270	1300	-454	2372	degrees/10	-2700	13000	-4540	23720
R, S	-50	1768	-58	3214.4	degrees	-50	1768	-58	3214
T	-270	400	-454	752	degrees/10	-2700	4000	-4540	7520

NOTE: To read a linearized PT100 temperature sensor with the SNAP-AIRTD module, access the engineering units.

0x0B - Analog Output Point

The Analog Output Point class provides access to the output channels of SNAP-PAC analog I/O modules.

Class Attributes

0x0B - Analog Output Point - Class Attributes						
Attribute ID	**Access Rule**	**NV**	**Name**	**Data Type**	**Description**	**Semantics**
0x1	GET		Revision	UINT	Revision of this open object	The revision of the Analog Output Point specified by the CIP Common Specification. The current value assigned to this attribute is 2.
0x64	GET		Vendor Specific Revision	UINT	Revision of vendor specific extension to this open object	The revision of our vendor specific extensions to the open Analog Output Point specified by the CIP Common Specification. The current value assigned to this attribute is 1.

Instance Attributes

Each channel of I/O can be uniquely identified by channel number and slot number. The channel number can range from 0 through 31, and is unique for each channel on a particular module. The module slot number can range from 0 through 15 and is printed next to each module slot on the I/O rack. The instance id of the Analog Output Point object associated with a given channel number and slot is determined through the following equations:

Instance Id = 1 + (Module Slot Number)*64 + (Module Channel Number)

Module Slot Number = RoundDown((Instance Id - 1)/64)

Channel Number = (Instance Id - 1) % 64

0x0B - Analog Output Point - Instance Attributes						
Attribute ID	**Access Rule**	**NV**	**Name**	**Data Type**	**Description**	**Semantics**
0x1	GET		Number Of Attributes	USINT	Number supported in this product	Default value is 21.
0x2	GET		Attribute List	ARRAY OF USINT	List of attributes supported in this product	
0x3	SET	V	Value	INT	Analog output value. See Semantics for representation details. For analog output value in REAL format, see rValue attribute.	See “Analog Output: Value Attribute Semantics” on page 44

0x0B - Analog Output Point - Instance Attributes (Continued)						
Attribute ID	**Access Rule**	**NV**	**Name**	**Data Type**	**Description**	**Semantics**
0x64	GET		Module (Slot) Number	UINT	The module slot associated with this I/O point.	The Module Slot Number identifies the slot into which the module associated with this I/O point is installed. The first slot (nearest the brain) is assigned Module Slot Number 0. The module slot number is also related to the instance id for this object as follows: Module Slot Number = RoundDown((Instance Id-1)/64)
0x65	GET		Module Channel Number	UINT	The module channel associated with this I/O point.	The first channel on a module is channel number 0. The module channel number is related to the instance id for this object as follows: Module Channel Number = (Instance Id - 1) % 64
0x66	GET		Module Path	STRUCT OF	EPATH for the Module Configuration object associated with this I/O point. This path contains one logical class segment and one logical instance segment.	
			Path Size	UINT	Length (in octets) of the Packed EPATH contained in the Path.	
			Path	PACKED EPATH	EPATH for the Module Configuration object associated with this I/O point. This path contains one logical class segment and one logical instance segment.	
0x67	SET	NVS	Point Name	STRING	Name for this point	Maximum length of string is 50 characters. The response to attempts to write a name length longer than 50 characters will contain general status code 0x15 (Too much data.)
0x68	GET		Module Type	UINT	Read only access to the Module Type attribute for the module associated with this point.	
0x69	SET	1	Point Type	UINT	The point type and associated module type determine the range and the engineering units used by a point.	
0x80	SET	NVS	Feature Select	UDINT	Selects the feature to enable for this point.	Currently there are no features defined for analog outputs and this value can not be changed from 0. However, the feature fields are included in the point object to accommodate future feature additions. The feature Value Type is Always 0xC8 UDINT. The Feature Value is Always 0.

<table>
<tr><th colspan="7">0x0B - Analog Output Point - Instance Attributes (Continued)</th></tr>
<tr><th>Attribute ID</th><th>Access Rule</th><th>NV</th><th>Name</th><th>Data Type</th><th>Description</th><th>Semantics</th></tr>
<tr><td>0x81</td><td>SET</td><td>NVS</td><td>Feature Enable</td><td>BOOL</td><td>Enables the selected feature.</td><td>0 = Selected feature is not enabled. See Feature Select Semantics for further details.
1 = Selected feature is enabled. See Feature Select Semantics for further details.</td></tr>
<tr><td>0x82</td><td>GET</td><td></td><td>Feature Value Type</td><td>USINT</td><td>Specifies data type of the Feature Value attribute.</td><td>Specifies the data type of the Feature Value attribute. Default value is 0xC8 - UDINT
See Feature Select Semantics for Feature Value Type for a given Feature Select.</td></tr>
<tr><td>0x83</td><td>GET</td><td></td><td>Feature Value</td><td>Depends on value of feature value type attribute 130</td><td>Value of selected feature.</td><td>he value read from this attribute depends on the value of the Clear Feature Value attribute:<table><tr><th>Clear Feature Value</th><th>Feature Value</th></tr><tr><td>0</td><td>Is equal to the current feature value.</td></tr><tr><td>1</td><td>Is equal to the feature value latched when the Clear Feature Value attribute was changed from 0 to 1.</td></tr></table>See Feature Select Semantics for further details.</td></tr>
<tr><td>0x84</td><td>SET</td><td></td><td>Clear Feature Value</td><td>BOOL</td><td>Sets the Feature Value to zero.</td><td>The effect of writing a value to this attribute depends on the previous value:<table><tr><th>Previous Value</th><th>Next Value</th><th>Behavior</th></tr><tr><td>0</td><td>0</td><td>No effect. The current feature value can be read from the Feature Value attribute.</td></tr><tr><td>0</td><td>1</td><td>Latch current Feature Value and clear Feature Value. The feature value read here will be latched and returned in response to Feature Value reads as long as the Clear Feature Value attribute remains set to 1.</td></tr><tr><td>1</td><td>1</td><td>No effect. The feature value latched when Clear Feature Value attribute was changed from 0 to 1 is returned in response to Feature Value reads.</td></tr><tr><td>1</td><td>0</td><td>The current feature value can now be read from the Feature Value attribute.</td></tr></table></td></tr>
</table>

0x0B - Analog Output Point - Instance Attributes (Continued)						
Attribute ID	**Access Rule**	**NV**	**Name**	**Data Type**	**Description**	**Semantics**
0x87	SET	NVS	Comm Watchdog Enable	BOOL	Enable/Disable communication watchdog operation. The communication watchdog value is written to the output point when no commands have been received for the amount of time configured in the Communication Watchdog object.	0 = Disable Comm watchdog operation for this point (default) 1= Enable Comm Watchdog operation for this point.
0x88	SET	NVS	Comm Watchdog Value	REAL	Value written to the rValue attribute if the Comm Watchdog Enable attribute =1 and a Comm Watchdog timeout occurs.	
0x89	SET		rValue	REAL	Analog input value given in engineering units in IEEE floating point format.	The engineering units used are determined by the output point type. See Semantics for the Value attribute.
0x8A	SET	NVS	Lower Scaled Engineering Units	REAL	Bottom of rValue range in engineering units.	The scale of an analog output point defines the range of values accepted in the rValue attribute, in engineering units. The default scale of an output point is determined by the point type and can be read from the Lower Scaled Engineering Units and the Upper Scaled Engineering Units attributes. The default scale can be changed by writing to these attributes.
0x8B	SET	NVS	Upper Scaled Engineering Units	REAL	Top of rValue range in engineering units.	
0x90	SET	NVS	Low Clamp	REAL	Minimum value sent to output point. Any attempts to set the output point to a lower value will result in the Low Clamp value being sent to the output point.	If High Clamp = 0.0, and Low Clamp = 0.0, clamping is disabled.

0x0B - Analog Output Point - Instance Attributes (Continued)						
Attribute ID	**Access Rule**	**NV**	**Name**	**Data Type**	**Description**	**Semantics**
0x8F	SET	NVS	High Clamp	REAL	Maximum value sent to output point. Any attempts to set the output point to a higher value will result in the High Clamp value being sent to the output point. Clamping consists of setting upper and lower limits on values sent to an analog output point so they do not go above or below a specific limit. For example, if you are using a 0-10 VDC output module, but the device attached to one of its points can handle a maximum of only 5 VDC, you can set an upper clamp of 5VDC for that point.	If High Clamp = 0.0, and Low Clamp = 0.0, clamping is disabled.

Common Services

0x0E - Get Attribute Single

0x10 - Set Attribute Single

Vendor Specific Services

0x32 - Read And Clear Attribute

- Description

  Returns the contents of the specified attribute and then sets the contents to 0. It may be used with the Feature Value attribute.

- Request Service Data Field Parameters

  None

- Success Response Service Data Field Parameters

Name	Data Type	Description Of Parameter
Attribute Data	Object/class attribute specific structure	Contains the requested attribute data

Analog Output: Value Attribute Semantics

The Value attribute specifies the output point value in proportional counts. The following formula converts an analog output value in proportional counts to engineering units:

value in engineering units = (proportional counts) * ((Engineering Units @ high scale) - (Engineering Units @ low scale))/(high scale counts)

To specify the output point value in engineering units, as a REAL data type, see the rValue attribute.

High scale counts are either 4095 (0xFFF) or 65535 (0xFFFF).

Legend	Low Scale
	High Scale

For Output Types: ±10V,0-10V,0-20mA,4-20mA on SNAP-AOV-5,SNAP-AOV-25, SNAP-AOV-7, SNAP-AOV-27, SNAP-AOA-3, SNAP-AOA-23, SNAP-AOA23-I, SNAP-AOA-28	
Value (decimal)	**Engineering Units**
0	Low scale (e.g. -10V)
4095	High scale (e.g. +10V)

For Output Types: 4-20mA on SNAP-AOA-3-16	
Value (decimal)	**Engineering Units**
0	Low scale (4mA)
65535	High scale (20mA)

0x66 - Load Cell Input

The Load Cell Input class provides access to the input channel of SNAP load cell modules (SNAP-AILC, SNAP-AILC-2).

Class Attributes

0x66 - Load Cell Input - Class Attributes						
Attribute ID	**Access Rule**	**NV**	**Name**	**Data Type**	**Description**	**Semantics**
0x1	GET		Revision	UINT	Revision of this object	The current value assigned to this attribute is one (01). If updates that require an increase in this value are made, then the value of this attribute increases by 1.

Instance Attributes

Load cell modules have only one physical input, but utilize two logical input channels. Each channel of I/O can be uniquely identified by channel number and slot number. The channel number can range from 0 through 1. Channel 0 provides unfiltered values from the physical input. Channel 1 provides filtered values from the physical input. The module slot number can range from 0 through 15 and is printed next to each module slot on the I/O rack. The instance id of the Load Cell Input object associated with a given channel number and slot is determined through the following equations:

Instance Id = 1 + (Module Slot Number)*64 + (Module Channel Number)

Module Slot Number = RoundDown((Instance Id - 1)/64)

Channel Number = (Instance Id - 1) % 64

0x66 - Load Cell Input - Instance Attributes						
Attribute ID	**Access Rule**	**NV**	**Name**	**Data Type**	**Description**	**Semantics**
0x1	GET		Number Of Attributes	USINT	Number supported in this product	Default value is 26.
0x2	GET		Attribute List	ARRAY OF USINT	List of attributes supported in this product	
0x3	GET		Value	INT	The 16-bit load cell input point value in proportional counts. See Semantics for representation details. For 32-bit load cell input value in proportional counts, see Value32 attribute. For load cell input value in REAL format, see rValue attribute. The SNAP-AILC has a single load cell point associated with the first two channels. The first channel provides unfiltered load cell input values while the second channel provides filtered load cell input values. See the Filter Weight attribute and Fast Settle Trigger Level attribute for more details.	See "Load Cell Input: Value and Value32 Attribute Semantics" on page 54.

<table>
<tr><th colspan="7">0x66 - Load Cell Input - Instance Attributes</th></tr>
<tr><th>Attribute ID</th><th>Access Rule</th><th>NV</th><th>Name</th><th>Data Type</th><th>Description</th><th>Semantics</th></tr>
<tr><td>0x64</td><td>GET</td><td></td><td>Module (Slot) Number</td><td>UINT</td><td>The module slot associated with this I/O point.</td><td>The Module Slot Number identifies the slot into which the module associated with this I/O point is installed. The first slot (nearest the brain) is assigned Module Slot Number 1. The Module Slot Number is related to the instance id of the Module Configuration object associated with the module for this I/O point.
Module Slot Number = Module Configuration Object Instance Id
The module number is also related to the instance id for this I/O point as follows:
Module Number = 1 + RoundDownToNearestInt((I/O Point Instance Id-1)/64)</td></tr>
<tr><td>0x65</td><td>GET</td><td></td><td>Module Channel Number</td><td>UINT</td><td>The module channel associated with this I/O point.</td><td>The Module Slot Number identifies the slot into which the module associated with this I/O point is installed. The first slot (nearest the brain) is assigned Module Slot Number 1. The Module Slot Number is related to the instance id of the Module Configuration object associated with the module for this I/O point.
Module Slot Number = Module Configuration Object Instance Id
The module number is also related to the instance id for this I/O point as follows:
Module Number = 1 + RoundDownToNearestInt((I/O Point Instance Id-1)/64)</td></tr>
<tr><td rowspan="3">0x66</td><td rowspan="3">GET</td><td rowspan="3"></td><td>Module Path</td><td>STRUCT OF</td><td>EPATH for the Module Configuration object associated with this I/O point. This path contains one logical class segment and one logical instance segment.</td><td></td></tr>
<tr><td>Path Size</td><td>UINT</td><td>Length (in octets) of the Packed EPATH contained in the Path.</td><td></td></tr>
<tr><td>Path</td><td>PACKED EPATH</td><td>EPATH for the Module Configuration object associated with this I/O point. This path contains one logical class segment and one logical instance segment.</td><td></td></tr>
<tr><td>0x67</td><td>SET</td><td>NVS</td><td>Point Name</td><td>STRING</td><td>Name for this point</td><td>Maximum length of string is 50 characters. The response to attempts to write a name length longer than 50 characters will contain general status code 0x15 (Too much data.)</td></tr>
<tr><td>0x68</td><td>GET</td><td></td><td>Module Type</td><td>UINT</td><td>Read only access to the Module Type attribute for the module associated with this point.</td><td>Possible module types are:
<table>
<tr><th>Module Type</th><th>Description</th></tr>
<tr><td>0x0B</td><td>SNAP-AILC</td></tr>
<tr><td>0x0C</td><td>SNAP-AILC-2</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><th colspan="7">0x66 - Load Cell Input - Instance Attributes</th></tr>
<tr><th>Attribute ID</th><th>Access Rule</th><th>NV</th><th>Name</th><th>Data Type</th><th>Description</th><th>Semantics</th></tr>
<tr><td>0x69</td><td>SET</td><td>NVS</td><td>Point Type</td><td>UINT</td><td>The point type and associated module type determine the range and the engineering units used by a point.</td><td></td></tr>
<tr><td>0x80</td><td>SET</td><td>NVS</td><td>Feature Select</td><td>UDINT</td><td>Selects the feature to enable for this point.</td><td>Currently there are no features defined for load cell inputs and this value can not be changed from 0. However, the feature fields are included in the point object to accommodate future feature additions. The feature Value Type is always 0xC8 UDINT. The Feature Value is always 0.</td></tr>
<tr><td>0x81</td><td>SET</td><td>NVS</td><td>Feature Enable</td><td>BOOL</td><td>Enables the selected feature.</td><td>0 = Selected feature is not enabled. See Feature Select Semantics for further details.
1 = Selected feature is enabled. See Feature Select Semantics for further details.</td></tr>
<tr><td>0x82</td><td>GET</td><td></td><td>Feature Value</td><td>USINT</td><td>Specifies data type of the Feature Value attribute.</td><td>Specifies the data type of the Feature Value attribute. Default value is 0xC8 - UDINT
See Feature Select Semantics for Feature Value Type for a given Feature Select.</td></tr>
<tr><td>0x83</td><td>GET</td><td></td><td>Feature Value</td><td>DEPENDS ON VALUE OF FEATURE VALUE TYPE ATTRIBUTE 130</td><td>Value of selected feature.</td><td>The value read from this attribute depends on the value of the Clear Feature Value attribute:
<table>
<tr><th>Clear Feature Value</th><th>Feature Value</th></tr>
<tr><td>0</td><td>Is equal to the current feature value.</td></tr>
<tr><td>1</td><td>Is equal to the feature value latched when the Clear Feature Value attribute was changed from 0 to 1.</td></tr>
</table>
See Feature Select Semantics for further details.</td></tr>
</table>

<table>
<tr><th colspan="7">0x66 - Load Cell Input - Instance Attributes</th></tr>
<tr><th>Attribute ID</th><th>Access Rule</th><th>NV</th><th>Name</th><th>Data Type</th><th>Description</th><th>Semantics</th></tr>
<tr><td>0x84</td><td>SET</td><td>V</td><td>Clear Feature Value</td><td>BOOL</td><td>Provides a method by which the feature value can be cleared via implicit messaging.</td><td>The affect of writing a value to this attribute depends on the previous value:
<table>
<tr><th>Previous Value</th><th>Next Value</th><th>Behavior</th></tr>
<tr><td>0</td><td>0</td><td>No effect. The current feature value can be read from the Feature Value attribute.</td></tr>
<tr><td>0</td><td>1</td><td>Latch current Feature Value and clear Feature Value. The feature value read here will be latched and returned in response to Feature Value reads as long as the Clear Feature Value attribute remains set to 1.</td></tr>
<tr><td>1</td><td>1</td><td>No effect. The feature value latched when Clear Feature Value attribute was changed from 0 to 1 is returned in response to Feature Value reads.</td></tr>
<tr><td>1</td><td>0</td><td>The current feature value can now be read from the Feature Value attribute.</td></tr>
</table></td></tr>
</table>

<table>
<tr><th colspan="6">0x66 - Load Cell Input - Instance Attributes</th></tr>
<tr><th>Attribute ID</th><th>Access Rule</th><th>NV</th><th>Name</th><th>Data Type</th><th>Description</th><th>Semantics</th></tr>
<tr><td>0x85</td><td>SET</td><td>V</td><td>Minimum rValue</td><td>REAL</td><td>Minimum value of rValue attribute measured.</td><td>
The following table describes the interaction between the value written to this attribute and the value read from this attribute.
<table>
<tr><th>Write Value</th><th>Previous Write Value</th><th>Behavior</th><th>Subsequent Read Value</th></tr>
<tr><td>0.0</td><td>0.0</td><td>Has no effect.</td><td>minimum rValue</td></tr>
<tr><td>positive infinity</td><td>0.0</td><td>Clear the minimum rValue</td><td>positive infinity</td></tr>
<tr><td>positive infinity</td><td>positive infinity</td><td>Has no effect. A subsequent read from this attribute returns positive infinity, though the device is monitoring the input signal for minimum rValue.</td><td>positive infinity</td></tr>
<tr><td>0.0</td><td>positive infinity</td><td>Subsequent reads of this attribute return the maximum rValue</td><td>minimum rValue</td></tr>
</table>
Writes to this attribute may be used to clear the latched minimum rValue via implicit messaging. (When communicating via explicit messaging, the Read And Clear service should be used to clear the latched minimum rValue.) To prevent inadvertantly clearing the latched value multiple times, the clear latch operation is edge-sensitive with respect to the value written to this attribute, and the value read from this attribute is always positive infinity after a positive infinity has been written to it. This provides positive indication to the scanner of completion of the clear latch operation. When the scanner subsequently writes a 0.0 to this attribute, the actual latched minimum rValue can be read from this attribute. (Note, in RSLogix 5000, positive infinity is denoted 1.$)
A typical clear minimum rValue operation via implicit messaging consists of the following steps:
1. Set this attribute to positive infinity.
2. Wait until positive infinity is read from this attribute.
3. Set this attribute to 0.0.
</td></tr>
</table>

0x66 - Load Cell Input - Instance Attributes						
Attribute ID	**Access Rule**	**NV**	**Name**	**Data Type**	**Description**	**Semantics**
0x86	SET	V	Maximum rValue	REAL	Maximum value of rValue attribute measured.	Writes to this attribute may be used to clear the latched minimum rValue via implicit messaging. (When communicating via explicit messaging, the Read And Clear service should be used to clear the latched minimum rValue.) To prevent inadvertantly clearing the latched value multiple times, the clear latch operation is edge-sensitive with respect to the value written to this attribute, and the value read from this attribute is always positive infinity after a positive infinity has been written to it. This provides positive indication to the scanner of completion of the clear latch operation. When the scanner subsequently writes a 0.0 to this attribute, the actual latched minimum rValue can be read from this attribute. (Note, in RSLogix 5000, positive infinity is denoted 1.$) A typical clear minimum rValue operation via implicit messaging consists of the following steps: 1. Set this attribute to positive infinity. 2. Wait until positive infinity is read from this attribute. 3. Set this attribute to 0.0.
0x89	GET		rValue	REAL	Analog input value in % of full scale magnitude given in engineering units in IEEE floating point format.	For Load cell input value in proportional counts, see the Value and Value32 attributes. The SNAP-AILC has a single load cell point associated with the first two channels. The first channel provides unfiltered Load Cell input values while the second channel provides filtered Load Cell input values. See the Filter Weight attribute and Fast Settle Trigger Level attribute for more details.
0x8A	SET	NVS	Low Scaled Engineering Units	REAL	Bottom of input range in engineering units.	The scale of an input point defines the range of values returned, in engineering units (before Offset or Gain is applied.) The default scale of an input point is determined by input point type and can be read from the Low Scaled Engineering Units and the High Scaled Engineering Units attributes. The default scale can be changed by writing to these attributes. If Low Scaled Engineering Units and High Scaled Engineering Units are set to the same value, they will be set to default scale values instead.
0x8B	SET	NVS	High Scaled Engineering Units	REAL	Top of input range in engineering units.	
0x8C	SET	NVS	Offset	REAL	Offset applied to measured input value.	Offset is applied to the measured input value (after scaling) as follows: rValue = (Gain * measured value) + Offset rValue = (Gain * measured value) + Offset
0x8D	SET	NVS	Gain	REAL	Gain applied to measure input value.	If gain is non-zero, it is applied to the measured input value (after scaling) as follows: rValue = (Gain * measure value) + Offset

0x66 - Load Cell Input - Instance Attributes						
Attribute ID	**Access Rule**	**NV**	**Name**	**Data Type**	**Description**	**Semantics**
0x8E	SET	NVS	Averaging Filter Weight	REAL	Note: This attribute sets the filter weight for the averaging filter performed on the brain. The second channel on a SNAP-AILC module provides an average filtered value that is filtered on the module. See the Load Cell Average Filter Weight attribute for details on configuring that filter. The weight applied to each sample during averaging. Average filtering is used to smooth analog input signals that are erratic or change suddenly. The formula used for filtering is Y = (X - Y)/W + Y, where Y is the filtered value, X is the new unfiltered value, and W is the filter weight. Filtering is applied to the measured analog value after scaling but before gain and offset are applied.	If Averaging Filter Weight is < 0.5, no average filtering is performed. Average filtering is enabled, if Average Filter Weight is set to >0.5, the rValue will be set to the non-filtered value the next time the analog input point is sampled.
0x8F	SET	NVS	Load Cell Averaging Filter Weight	USINT	The configured filter weight for the load cell averaging filter performed on the SNAP-AILC input point. The filtered value can be read from the Value attribute of channel one (the second channel) of the SNAP-AILC module. The Value attribute of channel zero (the first channel) of the SNAP-AILC module provides unfiltered data.	The filter weight can be set by using the Load Cell Averaging Filter Weight attribute for either channel 0 or channel 1. Writes to this attribute for the channel 0 will be duplicated on channel 1 and vica versa. The weight can be specified from 0 - 255 at the expense of settling time and defaults to 128. A value of 0 or 1 disables the filter. The filtered reading is calculated from the first channel after each ADC conversion using the following formulat: Y = (X - Y)/W + Y, where Y is the filtered value, X is the new unfiltered value, and W is the filter weight.

0x66 - Load Cell Input - Instance Attributes						
Attribute ID	**Access Rule**	**NV**	**Name**	**Data Type**	**Description**	**Semantics**
0x91	SET	NVS	Fast Settle Level	INT	The filtered weight is reduced when the difference between the ADC data and the filtered data is greater than the Fast Settle Level. This feature is useful to decrease settling time when there are large step changes in the load cell output and a large filter weight. For more information on the weight filtering and fast settling features, see the commands Set Analog Load Cell Filter Level and Set Analog Load Cell Fast Settle Level in the ioControl Command Reference (Opto 22 form #1301).	The fast settle level can be set by using the Fast Settle Level attribute for either the first or second channel. Writes to this attribute for the first channel will be duplicated on the second channel and vica versa. Valid values are 0 to 32767. A value of 0, 1 or 32767 turns the Fast Settle feature off.
0x92	GET		Value32	DINT	The 32-bit load cell input point value in proportional counts. See Semantics for representation details. For 16-bit load cell input value in proportional counts, see Value attribute. For load cell input value in REAL format, see rValue attribute. The SNAP-AILC has a single load cell point associated with the first two channels. The first channel provides unfiltered load cell input values while the second channel provides filtered load cell input values. See the Filter Weight attribute and Fast Settle Trigger Level attribute for more details.	See details.

Common Services

- 0x0E - Get Attribute Single
- 0x10 - Set Attribute Single

Vendor Specific Services

0x32 - Read And Clear Attribute

- Description

Returns the contents of the specified attribute and then sets the contents to 0. It may be used with the Minimum rValue, Maximum rValue and Feature Value attributes.

- Request Service Data Field Parameters

  None

- Success Response Service Data Field Parameters

Name	Data Type	Description of Parameter
Attribute Data	Object/class attribute specific struct	Contains the requested attribute data

Load Cell Input: Value and Value32 Attribute Semantics

The SNAP-AILC Load Cell module uses a 24 bit count to represent the input point value scale of +- 100%. The least 8 bits of this count is truncated to provide the Value attribute. This results in proportional count scale of +- 25000 as with other bipolar input modules. The Value32 attribute provides the entire 24 bit count as a 32 bit integer. Please see the rValue attribute for the input point value in engineering units represented as a REAL data type. The following formula converts proportional counts to % full scale magnitude for the SNAP-AILC load cell input points:

% full scale magnitude = (Proportional Value Counts) *(100/25000)

% full scale magnitude = (Proportional Value32 Counts)*(100/6400000)

Legend	Low Scale
	High Scale

SNAP-AILC Load Cell Input Point Value		
Value (proportional counts)	**Value32 (proportional counts)**	**% full scale magnitude**
-32768	-8388608	out of range
-27500	-7040000	-110
-25000	-6400000	-100
0	0	0
25000	6400000	+100
27500	7040000	+110

0x68 - OptoMMP Request

This object class provides an interface through which OptoMMP requests may be made to the target device via explicit messaging. It provides read/write access to any data in the OptoMMP memory map via the Read Memory Map and Write Memory Map services. These services provide the ability to read or write a configurable number of elements of a specified data type, and handle any byte-swapping required for the elements.

Class Attributes

0x68 - OptoMMP Request - Class Attributes						
Attribute ID	**Access Rule**	**NV**	**Name**	**Data Type**	**Description**	**Semantics**
0x1	GET		Revision	UINT	Revision of this object	The current value assigned to this attribute is one (01). If updates that require an increase in this value are made, then the value of this attribute increases by 1.

Common Services

- 0x0E - Get Attribute Single
- 0x10 - Set Attribute Single

Object Specific Services

- 0x4B - Read Memory Map
  - Description
    This class-level service reads data from the OptoMMP memory map.
  - Request Service Data Field Parameters

Name	Data Type	Description of Parameters
Address	UDINT	OptoMMP memory map address to read. (Address may be obtained from OptoMMP protocol Guide.)
Data Type	UDINT	Data Type for each element read. The data type must be one of the following data type ids: **Data Type ID** / **Description** 0xC4 – DINT - Signed 32-bit integer 0xC6 – USINT - Unsigned 8-bit integer value 0xC8 – UDINT - Unsigned 32-bit integer value 0xCA – UDINT - Unsigned 32-bit IEEE-754 floating point value
Number Of Elements	UDINT	Number of elements to read. This parameter is limited by the following constraint: Number of Elements <= (500 /(Element Size in Bytes))

– Success Response Service Data Field Parameters

Name	Data Type	Description of Parameters
Data	Depends on Data Type and Number Of Elements request parameter values.	Data read from the memory map is returned as an array of elements of length equal to the value passed in the Number Of Elements request parameter. Each element of the array is of data type specified in the Data Type request parameter. This service performs byte-swapping if required so data returned is in Little-Endian format.

- 0x4C - Write Memory Map
  – Description
  This class level service writes data to the OptoMMP memory map.
  – Request Service Data Field Parameters

Name	Data Type	Description of Parameters
Address	UDINT	OptoMMP memory map address to read. (Address may be obtained from OptoMMP protocol Guide.)
Data Type	UDINT	Data Type for each element read. The data type must be one of the following data type ids: Data Type ID / Description 0xC4 – DINT - Signed 32-bit integer 0xC6 – USINT - Unsigned 8-bit integer value 0xC8 – UDINT - Unsigned 32-bit integer value 0xCA – UDINT - Unsigned 32-bit IEEE-754 floating point value
Number Of Elements	UDINT	Number of elements to write. This parameter is limited by the following constraint: Number of Elements < (482/(Element Size in Bytes))
Data	An array of elements of data type specified in Data Type parameter.	Data to be written to OptoMMP memory map. Each element must be in Little-Endian format. This service performs any byte swapping necessary to write the specified data-type into the OptoMMP memory map in Big-Endian format.

– Success Response Service Data Field Parameters
None

0x69 - Scratchpad DINT

The integer scratchpad is an array of 10240 integers that facilitate peer to peer transfer of integer data. Each instance of the Scratchpad DINT class provides access to one 32 bit integer element of the array.

Class Attributes

0x69 - Scratchpad DINT - Class Attributes						
Attribute ID	**Access Rule**	**NV**	**Name**	**Data Type**	**Description**	**Semantics**
0x1	GET		Revision	UINT	Revision of this object class specification.	
0x3	GET		Max Instance	UINT	Maximum instance number for objects of this class. The current value assigned to this attribute is 10240.	

Instance Attributes

0x69 - Scratchpad DINT - Instance Attributes						
Attribute ID	**Access Rule**	**NV**	**Name**	**Data Type**	**Description**	**Semantics**
0x3	SET	NVS	Value	DINT	Value of the 32-bit integer at index (InstanceId - 1) of the integer scratchpad array.	Default value = 0. Valid values are -2147483648 to 2147483647. This attribute can be read or written via implicit messaging as well as explicit messaging. Instance 1 provides access to the element at index 0 of the scratch pad array. Instance 10240 provides access to the element at index 10239 of the scratch pad array.

Common Services

0x0E - Get Attribute Single

0x10 - Set Attribute Single

0x70 - Scratchpad REAL

The float scratchpad is an array of 10240 single precision IEEE-754 floating point values that facilitate peer to peer transfer of float data. Each instance of the Scratchpad REAL class provides access to one float element of the array.

Class Attributes

0x70 - Scratchpad REAL - Class Attributes						
Attribute ID	**Access Rule**	**NV**	**Name**	**Data Type**	**Description**	**Semantics**
0x1	GET		Revision	UINT	Revision of this object class specification.	
0x3	GET		Max Instance	UINT	Maximum instance number for objects of this class. The current value assigned to this attribute is 10240.	

Instance Attributes

0x70 - Scratchpad REAL - Instance Attributes						
Attribute ID	**Access Rule**	**NV**	**Name**	**Data Type**	**Description**	**Semantics**
0x3	SET	NVS	Value	REAL	Value of the single precision IEEE-754 float at index (InstanceId - 1) of the float scratchpad array.	Default value = 0.0. This attribute can be read or written via implicit messaging as well as explicit messaging. Instance 1 provides access to element at index 0 of the scratch pad array. Instance 10240 provides access to the element at index 10239 of the scratch pad array.

Common Services

0x0E - Get Attribute Single

0x10 - Set Attribute Single

0x71 - Scratchpad STRING

The string scratchpad is an array of 64 STRING elements that facilitate peer-to-peer transfer of STRING data. Each instance of the Scratchpad STRING class provides access to one STRING element of the array. Each STRING consists of a 32-bit length field followed by up to 128 bytes of string data. The value of the length field indicates the number of bytes of data following the length field.

Class Attributes

0x71 - Scratchpad STRING - Class Attributes						
Attribute ID	**Access Rule**	**NV**	**Name**	**Data Type**	**Description**	**Semantics**
0x1	GET		Revision	UINT	Revision of this object class specification.	
0x2	GET		Max Instance	UINT	Maximum instance number for objects of this class. The current value assigned to this attribute is 64.	

Instance Attributes

0x71 - Scratchpad STRING - Instance Attributes						
Attribute ID	**Access Rule**	**NV**	**Name**	**Data Type**	**Description**	**Semantics**
0x3	SET	NVS	Value	STRUCT OF	Value at index (InstanceId - 1) of the string scratchpad array. Each string consists of a 32-bit Character Count field followed by a Content field with up to 128 bytes of string data. The value of the length field indicates the number of bytes of data following the length field.	The string scratchpad index associated with a Scratchpad STRING object instance is determined by the equation: index = InstanceID - 1 The default string value is a zero length string (Character Count = 0, and an empty Content field). The maximum Character Count allowed is 128. This attribute can be accessed via explicit or implicit messaging. If this attribute is set to a string with a Character Count greater than 128 via explicit messaging, the operation is ignored and a general status code 0x15 (Too much data) is returned. Attempts to set this attribute to a value with Character Count greater than 128 via implicit messaging are ignored. For implicit messaging, the width of the assembly field associated with this attribute should be equal to the maximum Character Count expected to be read or written + 4 bytes (for the Character Count field.) For example, to read or write values with Character Count up to 64, an assembly field at least 64+4=68 bytes wide must be configured. If the assembly field width is less than 4 bytes, read and write operations will be ignored. If the assembly field width is not long enough to hold the entire value to be read, the Content field will be truncated to fit the assembly field width. If the assembly field width is not long enough to hold the entire value to be written, the write operation is ignored.
			Character Count	DINT	Number of octets in the Content field	

0x71 - Scratchpad STRING - Instance Attributes (Continued)						
Attribute ID	Access Rule	NV	Name	Data Type	Description	Semantics
			Content	OCTET STRING (0..128)	Octets containing string contents (1 octet per character). The maximum character count allowed is 128.	

Common Services

0x0E - Get Attribute Single

0x10 - Set Attribute Single

0x73 - Pulse and TPO Generator

Each instance of this class provides access to a pulse generator for a discrete output of the same instance. Pulse generators are capable of producing single pulses, pulse trains, or continuous square waves of arbitrary duty cycle. A pulse is modeled as an optional delay phase (prior to the start of the pulse period) an A phase (the pulse) and a B phase (the remainder of the pulse period.) For example, the following diagram depicts the output of a pulse generator configured with a non-zero delay, a 50% duty cycle, and 2 pulses:

After configuring the pulse generator, it can be started by writing a non-zero value to the State attribute. When the pulse is completed, the pulse generator returns to the STOPPED state. The pulse generator can be forced into the STOPPED state by writing zero to the State attribute. The pulse generator may also be started and stopped via implicit messaging using the Start and Stop attributes.

Class Attributes

0x73 - Pulse Generator - Class Attributes						
Attribute ID	Access Rule	NV	Name	Data Type	Description	Semantics
0x1	GET		Revision	UINT	Revision of this object	The current value assigned to this attribute is one (01). If updates that require an increase in this value are made, then the value of this attribute increases by 1.

Instance Attributes

Each pulse generator is associated with a discrete output channel which can be uniquely identified by channel number and slot number. The channel number can range from 0 through 31, and is unique for each channel on a particular module. The module slot number can range from 0 through 15 and is printed next to each module slot on the I/O rack. The instance id of the Pulse Generator object associated with a given channel number and slot is determined through the following equations:

Instance Id = 1 + (Module Slot Number)*64 + (Module Channel Number)
Module Slot Number = RoundDown((Instance Id - 1)/64)
Channel Number = (Instance Id - 1) % 64

<table>
<tr><th colspan="7">0x73 - Pulse Generator - Instance Attributes</th></tr>
<tr><th>Attribute ID</th><th>Access Rule</th><th>NV</th><th>Name</th><th>Data Type</th><th>Description</th><th>Semantics</th></tr>
<tr><td>0x5</td><td>SET</td><td>NVS</td><td>Period</td><td>REAL</td><td>The length of the pulse period (on time + off time) in seconds.</td><td>Default value = 0.0s. The minimum period that can be specified depends on the brain type. The accuracy with which the pulse generator produces an output signal with the specified period depends on the product type and whether the discrete output is on a standard density module (<= 4 points per module) or a high density module.
For discrete output points on standard density modules (<=4 points) the minimum period and accuracy are given in the following table. The accuracy is equal to the Digital Feature Scan Interval. The accuracy given in the following table reflects the default setting for Digital Feature Scan Interval. If this setting is modified, pulse period accuracy will also change.
<table><tr><th>Product</th><th>Minimum Period (seconds)</th><th>Accuracy (seconds)</th></tr><tr><td>SNAP-PAC-R1</td><td>0.006</td><td>0.001</td></tr><tr><td>SNAP-PAC-R2</td><td>0.100</td><td>0.001</td></tr><tr><td>SNAP-PAC-EB1</td><td>0.040</td><td>0.005</td></tr><tr><td>SNAP-PAC-EB2</td><td>0.100</td><td>0.005</td></tr></table>For discrete output points on high density modules (>4 points), the minimum period and accuracy are 1s for all products.
The maximum period that can be specified is 4,294,967 seconds (49.7 days).</td></tr>
<tr><td>0x6</td><td>SET</td><td>NVS</td><td>Percent</td><td>REAL</td><td>The length of the A phase of the pulse signal (on time if the Invert attribute is 0) expressed as a percent of the signal period.</td><td>Default value = 0.0. This attribute is related to the length of the A phase of the pulse signal (on time if the Invert attribute is 0) as follows:
<A Time> = <Period>*<Percent>/100.0</td></tr>
</table>

0x73 - Pulse Generator - Instance Attributes (Continued)						
Attribute ID	**Access Rule**	**NV**	**Name**	**Data Type**	**Description**	**Semantics**
0x7	SET	NVS	Delay	REAL	The length of the optional delay phase in seconds. A value of 0 eliminates the delay phase from the signal.	Default value = 0.0. If this value is > 0 the delay phase is enabled. The delay phase occurs between the start of the waveform and the beginning of the first A phase.
0x8	SET	NVS	Pulse Quantity	UDINT	The number of periods included in the output waveform.	Default value = 0. A value of 0 specifies a continuous wave form the repeats the configured pulse period indefinitely. A value of 1 or more specifies a wave form containing the specified number of pulses.
0x9	SET	NVS	Invert	BOOL	Inverts the output pulse signal. (Signal is OFF during the A phase and ON during the B phase and optional delay phase.)	Default value = 0. A value of 1 inverts the output pulse signal.
0xA	SET	V	Start	BOOL	This attribute activates (or re-activates) the pulse generator when it is changed from 0 to 1.	This attribute provides positive feedback for starting the pulse generator via implicit messaging. By including this attribute in both the input and output data images, the pulse generator can be activated by changing the output value for this attribute from 0 to 1. When the input value becomes 1, the pulse generator has been started and the output value may be set back to 0.
0xB	SET	V	Stop	BOOL	This attribute stops the pulse generator when it is changed from 0 to 1.	This attribute provides positive feedback for stopping the pulse generator via implicit messaging. By including this attribute in both the input and output data images, the pulse generator can be stopped by changing the output value for this attribute from 0 to 1. When the input value becomes 1, the pulse generator has been stopped and the output value may be set back to 0.
0xC	SET	V	State	BOOL	The current state of the pulse generator.	The current state of the pulse generator is encoded as follows: 0 = STOPPED: The pulse generator is not running. 1 = ACTIVE: The pulse generator is running. If 1 is written to this attribute while the pulse generator is STOPPED, it will start. If 0 is written to this attribute while the pulse generator is ACTIVE, it will stop. To start/stop pulse generator via implicit messaging, use Start or Stop attributes instead of the State attribute.

0x73 - Pulse Generator - Instance Attributes (Continued)						
Attribute ID	**Access Rule**	**NV**	**Name**	**Data Type**	**Description**	**Semantics**
0xD	GET		Pulses Remaining	UDINT	The number of pulses remaining to be generated.	If the Pulse Quantity attribute is non-zero, this attribute counts down from the Pulse Quantity value as each pulse is generated. It is decremented at the end of the B phase/beginning of the A phase. If the PulseQuantity attribute is zero, the generated signal is continuous, and this value remains 0.

Common Services

- 0x0E - Get Attribute Single
- 0x10 - Set Attribute Single

0x74 - Ramp Controller

- The Ramp Controller object class gradually changes an analog output value. Each Ramp Controller object is associated with the Analog Output object with matching instance ID. If a Ramp Controller is configured with non-zero Rate, and the End Value is changed, the Analog Output rValue attribute is changed linearly (or *ramped*) until it equals the End Value. The Rate attribute indicates the rate at which the rValue is changed. While the rValue is being ramped, the ramp controller adds or subtracts Rate*0.050s every 50ms until the rValue equals the End Value. Ramp Controller functionality is disabled or canceled by setting the Rate attribute to zero.

Class Attributes

0x74 - Ramp Controller - Class Attributes						
Attribute ID	**Access Rule**	**NV**	**Name**	**Data Type**	**Description**	**Semantics**
0x1	GET		Revision	UINT	Revision of this object	The current value assigned to this attribute is one (01). If updates that require an increase in this value are made, then the value of this attribute increases by 1.

Instance Attributes

Each Ramp Controller object ramps the Analog Output object with the same instance ID. The relationship between Ramp Controller object instance ID and slot/channel number of the analog output it ramps is:

Instance Id = 1 + (Module Slot Number)*64 + (Module Channel Number)
Module Slot Number = RoundDown((Instance Id - 1)/64)
Channel Number = (Instance Id - 1) % 64

0x74 - Ramp Controller - Instance Attributes						
Attribute ID	**Access Rule**	**NV**	**Name**	**Data Type**	**Description**	**Semantics**
0x1	SET	NVS	End Value	REAL	Specifies the value to which the associated analog output point is ramped at the rate specified by the Rate attribute. The Rate attribute must be set to a non-zero value to enable the ramp controller.	If the End Value is changed while Rate is not zero, the Ramp Controller immediately begins ramping the analog output to the new End Value. If both the Rate attribute and the End Value attribute must be changed, change the Rate first, then change the End Value. If the Rate attribute is 0.0, no ramping of the associated analog output occurs When using the ramp controller to ramp an analog output rValue, do not set the analog output rValue attribute directly.
0x2	SET	NVS	Rate	REAL	The rate at which the analog output rValue is ramped to the End Value specified in analog output engineering units/second. A value of 0.0 disables the ramp controller.	The default Rate is 0.0. To enable the ramp controller, the Rate must be set to a value other than 0.0. Whenever a non-zero value is written to this attribute, the analog output is ramped from its current value to the End Value. If the Rate is not equal to 0.0, the ramp controller adds or subtracts Rate*0.050s to the analog output value every 50ms until the analog output value equals the End Value. If this attribute is changed to 0.0 while actively ramping (the Ramping attribute equals 1), ramping is canceled immediately and the Ramp Controller is disabled.
0x5	GET		State	BOOL	Indicates whether the ramp controller is actively ramping the associated analog output.	The possible values are: 0 = Not ramping: The ramp controller is not actively ramping an analog output. 1 = Ramping: The ramp controller is actively ramping an analog output to the value specified in the End Value attribute at the rate specified by the Rate attribute.

Common Services

- 0x0E - Get Attribute Single

- 0x10 - Set Attribute Single

0x75 - PID Loop Controller

Each instance of this class models a PID loop controller. Instance IDs 1-96 correspond to PID Loop Controllers 0-95 on the SNAP-PAC device. The instance attributes of this class may be read via explicit messaging using either the Get_Attribute_Single or Get_Attributes_All services. The instance attributes of this class may be read via implicit messaging by adding the desired attributes to an Assembly format.

Class Attributes

0x75 - PID Loop Controller - Class Attributes						
Attribute ID	**Access Rule**	**NV**	**Name**	**Data Type**	**Description**	**Semantics**
0x1	GET		Revision	UINT	Revision of this object class specification.	
0x2	GET		Max Instance	UINT	Maximum instance number for objects of this class. The current value assigned to this attribute is 96.	The largest instance number of a created object at this class hierarchy level.

Instance Attributes

<table>
<tr><th colspan="7">0x75 - PID Loop Controller - Instance Attributes</th></tr>
<tr><th>Attribute ID</th><th>Access Rule</th><th>NV</th><th>Name</th><th>Data Type</th><th>Description</th><th>Semantics</th></tr>
<tr><td>0x1</td><td>SET</td><td>NVS</td><td>Algorithm</td><td>UDINT</td><td>The PID algorithm used.</td><td>Valid values are:
0: None
5: Velocity (Type C)
6: ISA
7: Parallel
8: Interacting
Default algorithm: None (0)
Key to terms used in algorithm definitions:
<table>
<tr><th>Terms</th><th>Definition</th></tr>
<tr><td>PV</td><td>Process variable; the input to the PID</td></tr>
<tr><td>SP</td><td>Setpoint</td></tr>
<tr><td>InLo, InHi</td><td>Range of the input</td></tr>
<tr><td>Gain</td><td>Proportional tuning parameter. Unitless. May be negative.</td></tr>
<tr><td>TuneI</td><td>Integral tuning parameter. In units of seconds. Increasing magnitude increases influence on output.</td></tr>
<tr><td>TuneD</td><td>Derivative tuning parameter. In units of seconds. Increasing magnitude increases influence on output.</td></tr>
<tr><td>Output</td><td>Output from the PID</td></tr>
<tr><td>Err_1</td><td>The Error (PV â€“ SP) from the previous scan</td></tr>
<tr><td>Integral</td><td>Integrator. Anti-windup is applied after the output is determined to be within bounds.</td></tr>
<tr><td>PvIn_1, PvIn2</td><td>PV from the previous scan and the scan before that.</td></tr>
<tr><td>Scantime</td><td>Actual scan time (time since previous scan)</td></tr>
</table></td></tr>
</table>

0x75 - PID Loop Controller - Instance Attributes (Continued)						
Attribute ID	**Access Rule**	**NV**	**Name**	**Data Type**	**Description**	**Semantics**
0x1 (continued)						Algorithm Definitions: • Equations Common to All Algorithms – Err = PV - SP – Span = (OutHi - OutLo) / (InHi - InLo) – Output = Output + FeedForward * TuneFF • Velocity (Type C) Algorithm – TermP = (PV - PV_1) – TermI = TuneI * ScanTime * Err – TermD = TuneD / ScanTime * (PvIn - 2 * PvIn_1 + PvIn_2) – ChangeInOutput = Span * Gain * (TermP + TermI + TermD) • Non-velocity Algorithms These equations were derived from the article A Comparison of PID Control Algorithms by John P. Gerry in Control Engineering (March 1987). These three equations are the same except for the tuning coefficients; converting from one equation to another is merely a matter of converting the tuning coefficients. Equations common to all but the velocity algorithm: – Integral = Integral + Err – TermP = Err TermI = TuneI * ScanTime * Integral – TermD = (TuneD / ScanTime) * (PvIn - PvIn_1) Ideal or ISA Algorithm: Output = Span * Gain * (TermP + TermI + TermD) Parallel Algorithm: Output = Span * (Gain * TermP + TermI + TermD) Interacting Algorithm: Output = Span * Gain * (TermP + TermI) * (1 + TermD) NOTE: In SNAP PAC PIDs, the derivative is applied only to the process variable (the input) and not to the setpoint. This means you can change the setpoint without causing spikes in the derivative term. These PIDs also prevent integral windup by back calculating the integral without the derivative term. The feedforward term (“bias”) is added before output clamping and has a tuning factor.
0x2	SET	NVS	Mode	UDINT	The mode in which the PID loop is currently running.	Valid Modes 0: Automatic - PID loop is running and is controlling the output. 1: Manual = PID loop is stopped and is not controlling the output. Default value = 0
0x3	SET	NVS	Scan Rate	REAL	Interval in seconds between PID loop calculations.	Minimum value is 0.001 (1 millisecond). Scan time should be greater than system lag (the time it takes for the controller output to have a measurable effect on the system). Also consider other PIDs and tasks on the remote I/O competing for processing power.

0x75 - PID Loop Controller - Instance Attributes (Continued)						
Attribute ID	**Access Rule**	**NV**	**Name**	**Data Type**	**Description**	**Semantics**
0x4	SET	NVS	Input Source	UDINT	Specifies the OptoMmp address associated with the input source for the PID loop. The data type of the input source must be REAL.	Example input sources: 0xF0260000 - Analog input point at slot 0, channel 0 0xF210048C - Output attribute of PID Loop instance ID = 10 0x00000000 - Value of Input attribute of this PID loop. Host determines input value by setting the Input attribute value. Default value = 0x00000000 See Form 1465, *OptoMMP Protocol Guide* for additional addresses.
0x5	SET	NVS	Input Low Range	REAL	The minimum valid value of the Input attribute.	See Configuration Flags attribute for optional responses to out-of-range input. When input is lower than this value, Bit 0 of the Status Flags attribute is set to 1.
0x6	SET	NVS	Input High Range	REAL	The maximum valid value of the Input attribute.	See Configuration Flags attribute for optional responses to out-of-range input. When input is higher than this value, Bit 1 of the Status Flags attribute is set to 1.
0x8	SET	NVS	Setpoint Source	UDINT	Specifies the OptoMmp address associated with the setpoint source for the PID loop. The data type of the setpoint source must be REAL.	Example setpoint sources: 0xF0260000 - Analog input point at slot 0, channel 0 0xF210048C - Output attribute of PID Loop instance ID = 10 0x00000000 - Value of Input attribute of this PID loop. Host determines input value by setting the Input attribute value. Default value = 0x00000000 See Form 1465, *OptoMMP Protocol Guide* for additional addresses.
0x9	SET	NVS	Output Destination	UDINT	Specifies the OptoMmp address associated with the output destination for the PID loop calculation. The data type of the destination output for the PID loop calculation must be REAL.	Example output destinations: 0xF02A1000 - Analog output point at slot 1, channel 0 0x00000000 - Output attribute of this PID loop. Host can read the output attribute value. This configuration is appropriate to use the output to control the setpoint or input of another PID. Default value = 0x00000000 See Form 1465, *OptoMMP Protocol Guide* for additional addresses.
0xA	SET		Output Lower Clamp	REAL	Prevents output value from falling below the specified value.	
0xB	SET		Output Upper Clamp	REAL	Prevents output value from exceeding the specified value.	

0x75 - PID Loop Controller - Instance Attributes (Continued)						
Attribute ID	**Access Rule**	**NV**	**Name**	**Data Type**	**Description**	**Semantics**
0xC	SET		Minimum Output Change	REAL	The minimum output change allowed (compared to the current output value). Default value is 0, which disables this feature.	
0xD	SET		Maximum Output Change	REAL	The maximum output change allowed (compared to the current output value). Default value is zero which disabled this feature.	
0xE	SET		Output When Input Low	REAL	This optional configuration attribute specifies the output value when the input value falls below Input Low Range.	For more information, see semantics for Force Output On Bad Input attribute.
0xF	SET		Output When Input High	REAL	This optional configuration attribute specifies the output value when the input value rises above Input High Range.	For more information, see semantics for Force Output On Bad Input attribute.

0x75 - PID Loop Controller - Instance Attributes (Continued)						
Attribute ID	**Access Rule**	**NV**	**Name**	**Data Type**	**Description**	**Semantics**
0x10	SET		Configuration Flags	DWORD	32 bit mask containing configuration flags.	Valid Values Bit 0 = 0: (Default) the PID algorithm uses the Input attribute value. Bit 0 = 1: Enables Square Root Input feature. The PID algorithm uses the square root of the Input attribute value. Bit 1 = 0: (Default) freeze output value while input is outside valid range. Bit 1 = 1: Force output to a predetermined value if Input attribute value is outside valid range defined by Input Low Range and Input High Range attributes. The value forced is: • Output When Input Low (if Input is below Input Low Range.) • Output When Input High (if Input is above Input High Range.) Bit 2 = 0: (Default) stay in automatic mode when input is out of range and continue control when input goes back into range. Bit 2 = 1: Switch to Manual mode when input attribute value is outside valid range defined by input Low Range and Input High Range attributes.
0x11	SET		Input	REAL	The last value sampled from Input.	If Input Source attribute value is 0, the host can set this attribute value.
0x12	SET		Setpoint	REAL	The last value sampled from Setpoint.	If the Setpoint Source attribute value is zero, the host can set this attribute value.
0x13	SET		Output	REAL	The current value of the output.	If the Mode attribute value is 1 (Manual) the host can set this attribute.
0x14	SET		FeedForward	REAL	Feed forward input value for the PID loop.	Default value is 0.
0x15	SET		Gain	REAL	Specifies the gain used for the configured PID algorithm.	
0x16	SET		Integral	REAL	Specifies the integral setting for the PID algorithm.	
0x17	SET		Derivative	REAL	Specifies the derivative setting for the PID algorithm.	
0x18	SET		FeedForward Gain	REAL	Specifies the Feed Forward Gain for the PID algorithm.	

0x75 - PID Loop Controller - Instance Attributes (Continued)						
Attribute ID	**Access Rule**	**NV**	**Name**	**Data Type**	**Description**	**Semantics**
0x19	GET		Scan Count	UDINT	Number of PID loop calculations performed. This value increments each time the PID output is recalculated.	
0x1A	GET		Status Flags	DWORD	32 bit mask indicating current status of PID loop. Bit 0 is least significant bit. Default value is 0.	Bit 0 is least significant bit. Default value is 0. • Bit 0 = 1: Input value is below valid range • Bit 1 = 1: Input value is above valid range • Bit 2 = 1: Output value is currently forced.
0x1B	GET		Current Error	REAL	Error value (input - setpoint) from most recent calculation.	
0x1C	GET		Current TermP	REAL	TermP value from most recent calculation.	
0x1D	GET		Current TermI	REAL	TermI value from most recent calculation.	
0x1E	GET		Current TermD	REAL	TermD value from most recent calculation.	
0x1F	GET		Current Integral	REAL	Integral value from most recent calculation.	

Common Services

0x01 - Get Attributes All

- At the class level this service is not supported.

- At the Instance level, the order of the attributes returned in the "Object/Service specific reply data" portion of the Get_Attributes_All response is as follows:

0x75 - PID Loop Controller - Common Services		
Name	**Data Type**	**Description**
0x1 - Algorithm	UDINT	The PID algorithm used.
0x2 - Mode	UDINT	The mode in which the PID loop is currently running.
0x3 - Scan Rate	REAL	Interval in seconds between PID loop calculations.
0x4 - Input Source	UDINT	Specifies the OptoMmp address associated with the input source for the PID loop. The data type of the input source must be REAL.
0x5 - Input Low Range	REAL	The minimum valid value of the Input attribute.
0x6 - Input High Range	REAL	The maximum valid value of the Input attribute.
0x8 - Setpoint Source	UDINT	Specifies the OptoMmp address associated with the setpoint source for the PID loop. The data type of the setpoint source must be REAL.
0x9 - Output Destination	UDINT	Specifies the OptoMmp address associated with the output destination for the PID loop calculation. The data type of the destination output for the PID loop calculation must be REAL.
0xA - Output Lower Clamp	REAL	Prevents output value from exceeding the specified value.
0xB - Output Upper Clamp	REAL	Prevents output value from falling below the specified value.
0xC - Minimum Output Change	REAL	The minimum output change allowed (compared to the current output value). Default value is 0, which disables this feature.
0xD - Maximum Output Change	REAL	The maximum output change allowed (compared to the current output value). Default value is zero which disabled this feature.
0xE - Output When Input Low	REAL	This optional configuration attribute specifies the output value when the input value falls below Input Low Range.
0xF - Output When Input High	REAL	This optional configuration attribute specifies the output value when the input value rises above Input High Range.
0x10 - Configuration Flags	DWORD	32 bit mask containing configuration flags.
0x11 - Input	REAL	The last value sampled from Input.
0x12 - Setpoint	REAL	The last value sampled from Setpoint.
0x13 - Output	REAL	The current value of the output.
0x14 - FeedForward	REAL	Feed forward input value for the PID loop.
0x15 - Gain	REAL	Specifies the gain used for the configured PID algorithm.
0x16 - Integral	REAL	Specifies the integral setting for the PID algorithm.
0x17 - Derivative	REAL	Specifies the derivative setting for the PID algorithm.
0x18 - FeedForward Gain	REAL	Specifies the Feed Forward Gain for the PID algorithm.
0x19 - Scan Count	UDINT	Number of PID loop calculations performed. This value increments each time the PID output is recalculated.
0x1A - Status Flags	DWORD	32 bit mask indicating current status of PID loop. Bit 0 is least significant bit. Default value is 0.

0x75 - PID Loop Controller - Common Services (Continued)		
Name	**Data Type**	**Description**
0x1B - Current Error	REAL	Error value (input - setpoint) from most recent calculation.
0x1C - Current TermP	REAL	TermP value from most recent calculation.
0x1D - Current TermI	REAL	TermI value from most recent calculation.
0x1E - Current TermD	REAL	TermD value from most recent calculation.
0x1F - Current Integral	REAL	Integral value from most recent calculation.

- 0x0E - Get Attribute Single
- 0x10 - Set Attribute Single

0x76 - Serial Port

Modules that provide serial ports are:

- SNAP-SCM-232 (2 ports)
- SNAP-SCM-485-422 (2 ports in 2-wire mode, 1 port in 4-wire mode)
- SNAP-SCM-PROFI (1 port)
- SNAP-SCM-MCH16 (1 port)

Each Serial Port object provides access to a serial port on a SNAP-SCM module for transmission, reception and configuration. Modules supported are SNAP-SCM-232, SNAP-SCM-485-422, SNAP-SCM-PROFI, SNAP-SCM-MCH16

Class Attributes

0x76 - Serial Port - Class Attributes						
Attribute ID	**Access Rule**	**NV**	**Name**	**Data Type**	**Description**	**Semantics**
0x1	GET		Revision	UINT	Revision of this object class specification is 1.	

Instance Attributes

Instance Id = 1 + (Module Slot Number)*64 + (Module Channel Number), where Module Channel Number = 0 for channel A and 1 for channel B.
Modules that provide serial ports are:

Module Slot Number = RoundDown((Instance Id - 1)/64)
Channel Number = (Instance Id - 1) % 64

0x76 - Serial Port - Instance Attributes						
Attribute ID	**Access Rule**	**NV**	**Name**	**Data Type**	**Description**	**Semantics**
0x1	SET	V	Transmit Data	STRUCT OF	Append data to the transmit FIFO.	When this attribute is written, all octets in the Data field are placed in the transmit FIFO. The number of octets specified in the Octet Count field specifies how many octets to place in the transmit FIFO of the serial port represented by this object. If the transmit FIFO can not accommodate all the octets specified, no octets are appended to the transmit FIFO and an 0x20 Invalid Parameter general status code is returned with an extended status code of 0xD0 - Too much tx data. Reads from this attribute have no impact on the transmit FIFO and always return Octet Count = 0. To determine the number of octets that can be accommodated by the FIFO, read the Transmit FIFO Free attribute.
			Octet Count	UDINT	Number of octets in the Data field	Octet count can be no larger than 200.
			Data	OCTET STRING (0..200)	Octets of transmit data. The maximum octet count allowed is 200.	

0x76 - Serial Port - Instance Attributes (Continued)						
Attribute ID	**Access Rule**	**NV**	**Name**	**Data Type**	**Description**	**Semantics**
0x2	GET		Receive Data	STRUCT OF	Retrieve data from the receive FIFO.	When this attribute is read, data is retrieved from the receive FIFO and returned as the attribute value. If the EOM Character List attribute is set to 0x00000000, the number of octets retrieved from the FIFO is the lesser of: • The value of the Receive FIFO Ready attribute (number of octets waiting to be received) • The value of the Maximum Octets To Retrieve attribute If EOM characters are configured via the EOM Character List attribute, the Receive Data attribute preserves message boundaries by retrieving all octets associated with a message together, provided the Maximum Octets To Retrieve attribute (receiveLength) has been set to a value that is at least as large as the largest expected message. If EOM characters are configured, the number of octets retrieved from the receive FIFO is: • If an EOM character occurs with in the next receiveLength octets in the receive FIFO—up to and including the next EOM character. • If an EOM character does not occur with in the next receiveLength octets in the receive FIFO - If there are less than receiveLength octets in the receive FIFO, no data is returned. - If there are receiveLength or more octets in the receive FIFO, receiveLength octets are returned. When this attribute is read, the Octet Count field is set to the number of octets retrieved from the receive FIFO and the Data field is filled with the octets. To determine the number of octets in the receive FIFO, read the Receive FIFO Ready attribute.
			Octet Count	UDINT	Number of octets in the Data field	Octet count can be no larger than 200.

<table>
<tr><th colspan="7">0x76 - Serial Port - Instance Attributes (Continued)</th></tr>
<tr><th>Attribute ID</th><th>Access Rule</th><th>NV</th><th>Name</th><th>Data Type</th><th>Description</th><th>Semantics</th></tr>
<tr><td></td><td></td><td></td><td>Data</td><td>OCTET STRING (0..200)</td><td>Octets of received data. The maximum octet count allowed is 200.</td><td></td></tr>
<tr><td>0x3</td><td>SET</td><td>V</td><td>Maximum Octets To Retrieve</td><td>UDINT</td><td>Maximum Number of octets to retrieve from the Receive FIFO when the Receive Data attribute is read.</td><td>The default value for this attribute is 200.
This value can be altered to prevent an overrun of application buffers.
Writes to this value place an upper limit on the number of octets that can be returned when reading the Receive Data attribute.</td></tr>
<tr><td>0x4</td><td>GET</td><td></td><td>Receive FIFO Ready</td><td>UDINT</td><td>Total number of octets in the receive FIFO.</td><td></td></tr>
<tr><td>0x5</td><td>GET</td><td></td><td>Transmit FIFO Free</td><td>UDINT</td><td>Number of new octets that can be accommodated in the transmit FIFO.</td><td>This attribute can be used to ensure the transmit FIFO is not overrun. The number of octets written to the Transmit Data attribute should be less than or equal to the value of this attribute to prevent transmit FIFO overrun.</td></tr>
<tr><td>0x6</td><td>SET</td><td>NVS</td><td>Bit Rate</td><td>UDINT</td><td>Rate at which bits are transmitted and received on the serial port expressed in bits per second.</td><td>The bit rates supported are documented in the SCM module documentation. Choosing an unsupported bit rate results in undefined behavior.</td></tr>
<tr><td>0x7</td><td>SET</td><td>NVS</td><td>Parity</td><td>USINT</td><td>Parity is a method of detecting some types of transmission errors by appending a single parity bit to each character. This attribute selects the type of parity algorithm to use. Default value is 78 (0x4E) =NONE</td><td>Supported parity types are:<table><tr><th>Value</th><th>Description</th></tr><tr><td>78 (0x4E)</td><td>None</td></tr><tr><td>79 (0x4F)</td><td>Odd</td></tr><tr><td>69 (0x45)</td><td>Even</td></tr><tr><td>77 (0x4D)</td><td>Mark</td></tr><tr><td>83 (0x53)</td><td>Space</td></tr></table></td></tr>
<tr><td>0x8</td><td>SET</td><td>NVS</td><td>Data Bits</td><td>USINT</td><td>Number of data bits in each character. Default value is 8 data bits per character.</td><td>Supported data bits per character are 7 or 8.</td></tr>
<tr><td>0x9</td><td>SET</td><td>NVS</td><td>Stop Bits</td><td>USINT</td><td>Stop bits sent at the end of every character allow the receiving signal hardware to detect the end of a character and to re-synchronize with the character stream. Default value is 1 stop bit.</td><td>Supported stop bit values are 1 or 2.</td></tr>
</table>

<table>
<tr><th colspan="7">0x76 - Serial Port - Instance Attributes (Continued)</th></tr>
<tr><th>Attribute ID</th><th>Access Rule</th><th>NV</th><th>Name</th><th>Data Type</th><th>Description</th><th>Semantics</th></tr>
<tr><td>0xA</td><td>SET</td><td>NVS</td><td>Flow Control</td><td>USINT</td><td>This attribute selects the flow control method used. Default value is 0 (No flow control).</td><td>Supported values are:
<table><tr><th>Value</th><th>Description</th></tr><tr><td>0</td><td>No flow control</td></tr><tr><td>1</td><td>Hardware flow control (RTR/CTS Bidirectional flow control).</td></tr></table></td></tr>
<tr><td>0xB</td><td>SET</td><td>NVS</td><td>Power-up Test Message</td><td>BOOL</td><td>This attribute enables or disables transmission of a power-up test message at startup. Default value is TRUE.</td><td></td></tr>
<tr><td>0xC</td><td>SET</td><td>NVS</td><td>EOM Character List</td><td>UDINT</td><td>This attribute specifies a list of message termination characters. Set this attribute to 0x00000000 to disable message termination functionality. This attribute alters the functionality of the Receive Data attribute as described in the Receive Data attribute semantics.</td><td>Any one of the specified terminating characters terminates data. Up to four termination characters can be specified using the following format: 0xAABBCCDD
Where:
• AA = ASCII code of the first terminating character in hexadecimal representation.
• BB = ASCII code pf the second terminating character in hexadecimal representation.
• CC = ASCII code of the third terminating character in hexadecimal representation.
• DD = ASCII code of the fourth terminating character in hexadecimal representation.
An ASCII code of 0 indicates the end of the list of terminating characters.
The default value 0x0D0A0000 specifies either 0x0D or 0x0A are to be considered message termination characters. To disable terminating character functionality (e.g. to accommodate binary data) set this attribute to 0.</td></tr>
<tr><td>0xC</td><td>SET</td><td>NVS</td><td>Mode</td><td>USINT</td><td>This attribute specifies the operation mode.</td><td>The default value of this attribute is 0. Only the SNAP-SCM-485-422 module utilizes this attribute to specify:
0 - 2-wire mode
1- 4-wire mode</td></tr>
</table>

Common Services

- 0x0E - Get Attribute Single

- 0x10 - Set Attribute Single

Object Specific Services

- 0x4B - SendAndReceive
  - Description

    This instance level service queues data in the transmit FIFO, and retrieves data from the receive FIFO. Data to transmit accompanies the service request. The service response contains data retrieved.
  - Request Service Data Field Parameters

Name	Data Type	Description of Parameters
Maximum Octets To Retrieve	UDINT	Maximum number of octets to retrieve from the Receive FIFO when the Receive Data attribute is read. This value only applies to the current service request. It is not written to the Maximum Number of Octets to Retrieve attribute. This value may not be greater than 200.
Octet Count	UDINT	Number of octets in the Transmit Data field to be written to the transmit FIFO. The octet count can be no larger than 200. See Transmit Data attribute semantics for more information.
Transmit Data	Octet string (0..200)	Octets of data to append data to the transmit FIFO. The maximum number of octets allowed is 200. See Transmit Data attribute for more information.

  - Success Response Service Data Field Parameters

Name	Data Type	Description of Parameters
Receive FIFO Ready	UDINT	Total number of octets in the receive FIFO.
Transmit FIFO Free	UDINT	Number of new octets that can be accommodated in the transmit FIFO.
Octet Count	UDINT	Number of octets in the Receive Data field. See Receive Data attribute for more information.
Receive Data	Octet string (0..200)	Octets of data retrieved from the receive FIFO. See Receive Data attribute for more details.

0x80 - Communication Watchdog

This object class provides configuration for the communication watchdog. The communication watchdog monitors communication received by the SNAP-PAC device. When no communication is received for the duration of the communication watchdog timeout, communication watchdog values are written to outputs for which the communication watchdog has been enabled. Incoming communication via any supported protocol (OptoMMP, Modbus TCP, EtherNET/IP) will reset the communication watchdog. SNAP-PAC devices implement one instance of this object class, instance ID = 1.

Class Attributes

0x80 - Communication Watchdog - Class Attributes						
Attribute ID	**Access Rule**	**NV**	**Name**	**Data Type**	**Description**	**Semantics**
0x1	GET		Revision	UINT	Revision of this object class.	The revision of this object class. The current value assigned to this attribute is 1.

Instance Attributes

Only one instance (instance ID = 1) is implemented in SNAP-PAC devices.

0x80 - Communication Watchdog - Instance Attributes						
Attribute ID	**Access Rule**	**NV**	**Name**	**Data Type**	**Description**	**Semantics**
0x1	SET	NVS	Timeout	UDINT	Length of communication watchdog timeout in milliseconds. If no incoming communication is detected for the configured amount of time, communication watchdog values will be written to all output points for which the communication watchdog enable attribute has been set. Default value = 0ms.	This attribute may be set to any value >= 0. A value of 0 disables the communication watchdog feature.

Common Services

- 0x0E - Get Attribute Single
- 0x10 - Set Attribute Single

0x81 - PPP Link

The PPP Link object class represents a point to point protocol link over a serial port.

Class Attributes

0x81 - PPP Link - Class Attributes						
Attribute ID	**Access Rule**	**NV**	**Name**	**Data Type**	**Description**	**Semantics**
0x1	GET		Revision	UINT	Revision of this object class specification is 1.	

Instance Attributes

There is only one instance of this object: instance Id 0x200 (512). This instance can also be accessed using instance id 1 if the request is received via the PPP link.

0x81 - PPP Link - Instance Attributes						
Attribute ID	**Access Rule**	**NV**	**Name**	**Data Type**	**Description**	**Semantics**
0x1	GET		Status	UDINT	Status of the PPP link.	Status codes: **Value** / **Description** 0 – Idle 1 – Outgoing Connecting 2 – Outgoing Connected 3 – Outgoing Disconnecting 4 – Listen 5 – Incoming Connecting 6 – Incoming Connected 7 – Incoming Disconnecting 8 – Shutting Down 9 – Disabled

Common Services

- 0x0E - Get Attribute Single
- 0x10 - Set Attribute Single

0xF5 - TCP/IP Object

SNAP-PAC-R1, SNAP-PAC-R2, SNAP-PAC-S1, SNAP-PAC-S2 implement 3 instances of the TCP/IP object. Instance ID 1 accesses the instance associated with the interface over which the request was received.
instance ID 0x100 ==> Ethernet 1 interface
instance ID 0x101 ==> Ethernet 2 interface
instance ID 0x200 ==> PPP interface
SNAP-PAC-EB1 and SNAP-PAC-EB2 implement a single instance of the TCP/IP object (accessed via instance ID 1) associated with the internal Ethernet interface connected to the 3 port Ethernet switch included in those products.

Class Attributes

0xF5 - TCP/IP Object - Class Attributes						
Attribute ID	**Access Rule**	**NV**	**Name**	**Data Type**	**Description**	**Semantics**
0x1	GET		Revision	UINT	Revision of this object	The current value assigned to this attribute is one (01). If updates that require an increase in this value are made, then the value of this attribute increases by 1.
0x2	GET		Max Instance	UINT	Maximum instance number of an object currently created in this class level of the device.	The largest instance number of a created object at this class hierarchy level.
0x3	GET		Number of Instances	UINT	Number of object instances currently created at this class level of the device.	The number of object instances at this class hierarchy level.

Instance Attributes

<table>
<tr><th colspan="7">0xF5 - TCP/IP Object - Instance Attributes</th></tr>
<tr><th>Attribute ID</th><th>Access Rule</th><th>NV</th><th>Name</th><th>Data Type</th><th>Description</th><th>Semantics</th></tr>
<tr><td>0x1</td><td>GET</td><td></td><td>Status</td><td>DWORD</td><td>Interface Status</td><td>The Status attribute is a bitmap that indicates the status of the TCP/IP network interface.
<table>
<tr><th>Bit(s)</th><th>Called</th><th>Definition</th><th>Value</th></tr>
<tr><td>0-3</td><td>Interface Configuration Status</td><td>Indicates the status of the Interface Configuration attribute.</td><td>Possible values are:
0 = The Interface Configuration attribute has not been configured.
1 = The Interface Configuration attribute contains valid configuration obtained from BOOTP or non-volatile storage.</td></tr>
<tr><td>4-31</td><td>Reserved/ Unused</td><td></td><td>0</td></tr>
</table></td></tr>
<tr><td>0x2</td><td>GET</td><td></td><td>Configuration Capability</td><td>DWORD</td><td>Interface configuration capability</td><td>The Configuration Capability attribute is a bitmap that indicates support for optional network configuration capability.
<table>
<tr><th>Bit(s)</th><th>Called</th><th>Definition</th><th>Value</th></tr>
<tr><td>0</td><td>BOOTP Client</td><td>1 (TRUE) Shall indicate the device is capable of obtaining its network configuration via BOOTP.</td><td>1 for ETHERNET 1 on SNAP-PAC-R/SNAP-PAC-S, 1 for SNAP-PAC-EB, 0 otherwise.</td></tr>
<tr><td>1</td><td>DNS Client</td><td>1 (TRUE) Shall indicate the device is capable of resolving host names by querying a DNS server.</td><td>0</td></tr>
<tr><td>2</td><td>DHCP Client</td><td>1 (TRUE) Shall indicate the device is capable of obtaining its network configuration via DHCP.</td><td>0</td></tr>
<tr><td>3</td><td>DHCP-DNS Update</td><td>1 (TRUE) Shall indicate the device is capable of sending its host name in the DHCP request as documented in Internet draft <draft-ietf-dhc-dhcp-dns-12.txt></td><td>0</td></tr>
</table></td></tr>
</table>

<table>
<tr><th colspan="7">0xF5 - TCP/IP Object - Instance Attributes (Continued)</th></tr>
<tr><th>Attribute ID</th><th>Access Rule</th><th>NV</th><th>Name</th><th>Data Type</th><th>Description</th><th>Semantics</th></tr>
<tr><td>0x3</td><td>SET</td><td>N</td><td>Configuration Control</td><td>DWORD</td><td>Interface configuration control</td><td>The Configuration Control attribute is a bitmap used to control network configuration options.
<table>
<tr><th>Bit(s)</th><th>Called</th><th>Definition</th><th>Value</th></tr>
<tr><td>0-3</td><td>Startup Configuration</td><td>Startup Configuration Determines how device shall obtain its initial configuration at start up.</td><td>Possible values are:
0 - The device shall use the interface configuration values previously stored.
1 - The device shall obtain its configuration values via BOOTP.
Valid values are:
1 or 0 for ETHERNET 1 on SNAP-PAC-R/SNAP-PAC-S or instance 0x1 on SNAP-PAC-EB.
0 for all other interfaces.</td></tr>
<tr><td>4</td><td>DNS Enable</td><td>If 1 (TRUE) device shall resolve all host names by querying a DNS server.</td><td>0</td></tr>
<tr><td>5-31</td><td>Reserved</td><td>Reserved for future use.</td><td>0</td></tr>
</table></td></tr>
<tr><td>0x4</td><td>GET</td><td></td><td>Physical Link Object</td><td>STRUCT OF</td><td>Path to Physical Link Object</td><td></td></tr>
<tr><td></td><td></td><td></td><td>Path size</td><td>UINT</td><td>Size of Path</td><td>Number of 16 bit words in Path</td></tr>
<tr><td></td><td></td><td></td><td>Path</td><td>PADDED EPATH</td><td>Size of Path</td><td>Logical segments identifying the physical link object. One class segment and one instance segment. Total EPATH length is less than or equal to 12 bytes.</td></tr>
<tr><td>0x5</td><td>GET</td><td></td><td>Interface Configuration</td><td>STRUCT OF</td><td>TCP/IP Network interface configuration</td><td></td></tr>
<tr><td></td><td></td><td></td><td>IP Address</td><td>UDINT</td><td>The interface's IP address</td><td>A value of 0 indicates no IP address has been configured. Otherwise the IP address shall be set to a valid Class A B or C address and shall not be set to the loopback address (127.0.0.1).</td></tr>
<tr><td></td><td></td><td></td><td>Network Mask</td><td>UDINT</td><td>The interface's network mask</td><td>A value of 0 indicates no network mask has been configured.</td></tr>
<tr><td></td><td></td><td></td><td>Gateway Address</td><td>UDINT</td><td>The Default gateway address</td><td>A value of 0 indicates no IP address has been configured. Otherwise the IP address shall be set to a valid Class A B or C address and shall not be set to the loopback address (127.0.0.1).</td></tr>
</table>

0xF5 - TCP/IP Object - Instance Attributes (Continued)						
Attribute ID	**Access Rule**	**NV**	**Name**	**Data Type**	**Description**	**Semantics**
			Name Server	UDINT	Primary name server	A value of 0 indicates no IP address has been configured. Otherwise the IP address shall be set to a valid Class A B or C address and shall not be set to the loopback address (127.0.0.1).
			Name Server 2	UDINT	Secondary name server	A value of 0 indicates no IP address has been configured. Otherwise the IP address shall be set to a valid Class A B or C address and shall not be set to the loopback address (127.0.0.1).
			Domain Name	STRING	Default domain name	ASCII characters. Maximum length is 48 characters Shall be padded to an even number of characters (pad not included in length). A length of 0 shall indicate no Domain Name is configured.
0x6	GET		Host Name	STRING	Host Name	ASCII characters. Maximum length is 64 characters. Shall be padded on an even number of characters. (Pad not included in length.) A length of 0 shall indicate no host name is configured.

Common Services

- 0x01 - Get Attribute All
  - Get Attribute All Response

    See table 5-3.10 from The CIP Networks Library Volume 2: EtherNet/IP Adaptation of CIP, section 5-3.3.2
- 0x0E - Get Attribute Single
- 0x10 - Set Attribute Single

0xF6 - Ethernet Link

SNAP-PAC-R1, SNAP-PAC-R2, SNAP-PAC-S1, SNAP-PAC-S2 implement three instances of the Ethernet Link object. Instance ID 1 accesses the instance associated with the interface over which the request was received.

Instance Id	Description
0x1	Instance associated with the interface over which the request was received.
0x100	External interface labeled ETHERNET 1.
0x101	External interface labeled ETHERNET 2.

SNAP-PAC-EB1 and SNAP-PAC-EB2 contain an embedded 3 port Ethernet switch. Two ports are connected to the RJ-45 connectors on the top cover of the brain, and the third port is connected internally to the processor. 3 Ethernet Link object instances are implemented, one for each of the 3 ports on the Ethernet switch.

Instance Id	Description
0x1	Represents internal connection between Ethernet switch and device processor.
0x100	External interface labeled ETHERNET 1.
0x101	External interface labeled ETHERNET 2.

Class Attributes

0xF6 - Ethernet Link - Class Attributes						
Attribute ID	**Access Rule**	**NV**	**Name**	**Data Type**	**Description**	**Semantics**
0x1	GET		Revision	UINT	Revision of this object	The current value assigned to this attribute is one (02). If updates that require an increase in this value are made, then the value of this attribute increases by 1.
0x2	GET		Max Instance	UINT	Maximum instance number of an object currently created in this class level of the device.	The largest instance number of a created object at this class hierarchy level. Value = 0x101.
0x3	GET		Number of Instances	UINT	Number of object instances currently created at this class level of the device	The number of object instances at this class hierarchy level. Value = 3.

Instance Attributes

<table>
<tr><th colspan="7">0xF6 - Ethernet Link - Instance Attributes</th></tr>
<tr><th>Attribute ID</th><th>Access Rule</th><th>NV</th><th>Name</th><th>Data Type</th><th>Description</th><th>Semantics</th></tr>
<tr><td>0x1</td><td>GET</td><td></td><td>Interface Speed</td><td>UDINT</td><td>Interface speed currently in use.</td><td>Speed in Mbps (e.g. 10, 100).</td></tr>
<tr><td>0x2</td><td>GET</td><td></td><td>Interface Flags</td><td>DWORD</td><td>Interface status flags.</td><td>Interface status flags
<table>
<tr><th>Bit(s)</th><th>Called</th><th>Definition</th></tr>
<tr><td>0</td><td>Link Status</td><td>Indicates whether or not the Ethernet 802.3 communications interface is connected to an active network. 0 indicates an inactive link; 1 indicates an active link. The determination of link status is implementation specific. In some cases devices can tell whether the link is active via hardware/driver support. In other cases, the device may only be able to tell whether the link is active by the presence of incoming packets.</td></tr>
<tr><td>1</td><td>Half/Full Duplex</td><td>Indicates the duplex mode currently in use. 0 indicates the interface is running half duplex; 1 indicates full duplex. Note that if the Link Status flag is 0, then the value of the Half/Full Duplex flag is indeterminate.</td></tr>
<tr><td>2-4</td><td>Negotiation Status</td><td>Indicates the status of link auto-negotiation</td></tr>
<tr><td>5</td><td>Manual Setting Requires Reset</td><td>0 indicates the interface can activate changes to link parameters (auto-negotiate, duplex mode, interface speed) automatically. 1 indicates the device requires a Reset service be issued to its Identity Object in order for the changes to take effect.</td></tr>
<tr><td>6</td><td>Local Hardware Fault</td><td>0 indicates the interface detects no local hardware fault; 1 indicates a local hardware fault is detected. The meaning of this is product-specific. Examples are an AUI/MII interface detects no transceiver attached or a radio modem detects no antennae attached. In contrast to the soft, possible self-correcting nature of the Link Status being inactive, this is assumed a hard-fault requiring user intervention.</td></tr>
<tr><td>7-31</td><td>Reserved</td><td>Shall be set to zero.</td></tr>
</table></td></tr>
<tr><td>0x3</td><td>GET</td><td></td><td>Physical Address</td><td>ARRAY OF 6 USINTS</td><td>MAC layer address associated with this interface.</td><td></td></tr>
</table>

<table>
<tr><th colspan="7">0xF6 - Ethernet Link - Instance Attributes (Continued)</th></tr>
<tr><th>Attribute ID</th><th>Access Rule</th><th>NV</th><th>Name</th><th>Data Type</th><th>Description</th><th>Semantics</th></tr>
<tr><td>0x7</td><td>GET</td><td></td><td>Interface Type</td><td>USINT</td><td>Type of interface. (Implemented on SNAP-PAC-EB1/SNAP-PAC-EB2 only)</td><td>On SNAP-PAC-EB1/SNAP-PAC-EB2 values are:
<table>
<tr><th>Interface Instance ID</th><th>Interface Type</th><th>Description</th></tr>
<tr><td>1</td><td>1</td><td>Internal interface</td></tr>
<tr><td>0x100, 0x101</td><td>2</td><td>Twisted pair</td></tr>
</table></td></tr>
<tr><td>0xA</td><td>GET</td><td></td><td>Interface Label</td><td>SHORT_STRING</td><td>Human readable identification.</td><td>On SNAP-PAC-EB1/SNAP-PAC-EB2 values are:
<table>
<tr><th>Interface Instance ID</th><th colspan="2">Interface Label</th></tr>
<tr><td rowspan="2">1</td><th>Product</th><th>Attribute Value</th></tr>
<tr><td>SNAP-PAC-EB1/ SNAP-PAC-EB2</td><td>“internal”</td></tr>
<tr><td></td><td>All others</td><td>The value for the interface over which the request was received.</td></tr>
<tr><td>0x100</td><td>“ETHERNET 1”</td><td></td></tr>
<tr><td>0x101</td><td>“ETHERNET 2”</td><td></td></tr>
</table></td></tr>
</table>

Common Services

- 0x0E - Get Attribute Single