FieldServer
FS-8700-21 LonWorks

Driver Manual
(Supplement to the FieldServer Instruction Manual)

APPLICABILITY & EFFECTIVITY

Effective for all systems manufactured after April 2019.

Driver Revision: 1.36

Document Revision: 1.B

Technical Support

Please call us for any technical support needs related to the FieldServer product.

Sierra Monitor Corporation
1991 Tarob Court
Milpitas, CA 95035

Website: www.sierramonitor.com

U.S. Support Information:

+1 408 964-4443
+1 800 727-4377

Email: support@sierramonitor.com

EMEA Support Information:

+31 33 808 0590

Email: support.emea@sierramonitor.com

TABLE OF CONTENTS

1 DESCRIPTION

The LonWorks driver allows the FieldServer to transfer data to and from devices using LonWorks protocol. The FieldServer can emulate either a Server or Client. The FS-B30, QuickServer and SlotServer Series FieldServers have a built-in LonWorks Interface. A Fieldbus connection is available on the FieldServer. The FS-B30 and the SlotServer can handle up to 4096 Network Variables which can be of the Standard Network Variable Types (SNVT) and/or User-defined Network Variable Types (UNVT), and the QuickServer can handle 250 Network Variables (or a limit of 500 for the enhanced model).

2 DEFINITIONS

2.1 NV Updates/Polls

On LonWorks networks, information is transferred using Network Variable Updates and Network Variable Polls. The Client requests Network Variables from the Server using Network Variable Polls and Network Variable Updates occur when a Client sends Network Variables to a Server.

2.2 Network Management

The FieldServer is able to set its own Domain, Subnet and Node ID at start-up using the FieldServer configuration file. Alternatively, the FieldServer is capable of being commissioned and bound by Network Management Tools such as LonMaker® - see Appendix B.1.1 for further information.

2.2.1 LonWorks – Program ID Calculation (SPID)

Network management tools use the Program ID to identify the different applications running on each device. Two devices on a network may have the same Program ID only if they have identical application/configurations.

The Program ID for the FieldServer consists of a fixed and a variable field (XX in the Program ID below).

90:00:95:47:1E:02:04:XX

The fixed part does not change for different configuration files or for different LonWorks driver versions. The variable part XX, which is also defined as the Model or Revision number, does however change.

The variable part is calculated at start-up when the configuration is loaded on the FieldServer. Certain critical fields in the configuration file are used to calculate this variable part of the Program ID. Thus if the configuration file changes significantly (critical fields) the Program ID will change. Also, as we add new features and capabilities to the LonWorks driver, this variable part of the Program ID may be different between LonWorks driver versions (even though the configuration file is the same).

It is possible though very unlikely that through a change in LonWorks driver and/or configuration file, the program IDs calculates out identical to another combination of driver and configuration. If this is the case with two FieldServers on the same network that run different configurations, one of these devices may fail to commission properly.

The calculated SPID can be found in line 5 of the fserver.xif file uploaded from the FieldServer.

The following configuration can be used to force the SPID:

```
Connections
Adapter      , Protocol     , SPID
LonWorks     , LonWorks     , 80:00:95:48:50:02:04:01
```

2.3 Clients/Servers

The FieldServer functions as a Client when polling or sending updates on the LonWorks network and as a Server when being polled or receiving updates. The FieldServer can be bound to a maximum number of 15 LonWorks Nodes. The FieldServer recommended point limit is 4096 points for the FS-B30.

2.4 Explicit/Implicit

Clients can address Servers using explicit or implicit addressing. Clients using explicit addressing obtain their data transfer parameters directly from the FieldServer configuration file. Implicit addressing is used when a Network Management Tool such as LonMaker® is used to connect a FieldServer to other LonWorks Nodes - the FieldServer is assigned its data transfer (binding) parameters by the Network Management Tool.

NOTE: Updating a configuration file or a profile change may result in a change in the XIF file. Any changes in the XIF will require the FieldServer to be recommissioned.

2.5 XIF Files

At start-up the FieldServer creates an external interface file (XIF) called fserver.xif based on the configuration file. The FieldServer differs from most other LonWorks drivers in that it is configurable and therefore its XIF is not fixed for all applications. The list of points available to the network will vary depending on the other networks connected to the FieldServer, and the requirements of the particular application. The recommended procedure for obtaining the XIF file is to upload it using the FieldServer FS-GUI page. Refer to **Appendix A.4** for more information.

2.6 Device Resource Files

The STANDARD device resource file covers FieldServer data type support. All Standard Network Variable Types (SNVTs) in this file are available to the user. These SNVTs are listed in **Section 6.1.3**. However, due to the sophisticated nature of LonWorks variables, it may not always be possible to typecast the full information supplied in a LonWorks variable to a simple Data Type contained in another protocol. The FieldServer supports Move and Logic functions which would allow the LonWorks variable to be split up and passed to separate simple data types, thus preventing information loss. The FieldServer also supports the sending and receiving of User-defined Network Variable Types (UNVTs) on the LonWorks Network. An UNVT is simply implemented as an array of bytes of user-defined length.

2.7 NV Aliases

The FieldServer currently supports a default of 63 network variable aliases to avoid network variable connection constraints. This number may be increased to a maximum of 4096.

2.8 Configuration Properties

Configuration Properties (CP's) characterize the behavior of a device in the system. Network installation tools realize this attribute and provide database storage to support maintenance operations. If a device fails and needs to be replaced, the configuration property data stored in the database is downloaded into the replacement device to restore the behavior of the replaced device in the system. Configuration properties are implemented on the FieldServer through configuration network variables. User-defined configuration property types are not supported. **Appendix A.5** provides a list of standard configuration property types that are supported.

The information that follows describes how to expand upon the factory defaults provided in the configuration files included with the FieldServer.

3 DRIVER SCOPE OF SUPPLY

3.1 Supplied by Sierra Monitor

Part #	Description
FS-8700-21	Driver Manual

3.2 Provided by the Supplier of 3rd Party Equipment

Part #	Description
N/A	LonMaker® Integration Tool from Echelon Corporation (optional network management software)
N/A	LonWatcher from Distech Controls Inc. (optional network management software)
N/A	Circon System Integrator from Circon Systems Corporation (optional network management software)
PCC10	PCCard LonWorks Adapter (optional hardware)

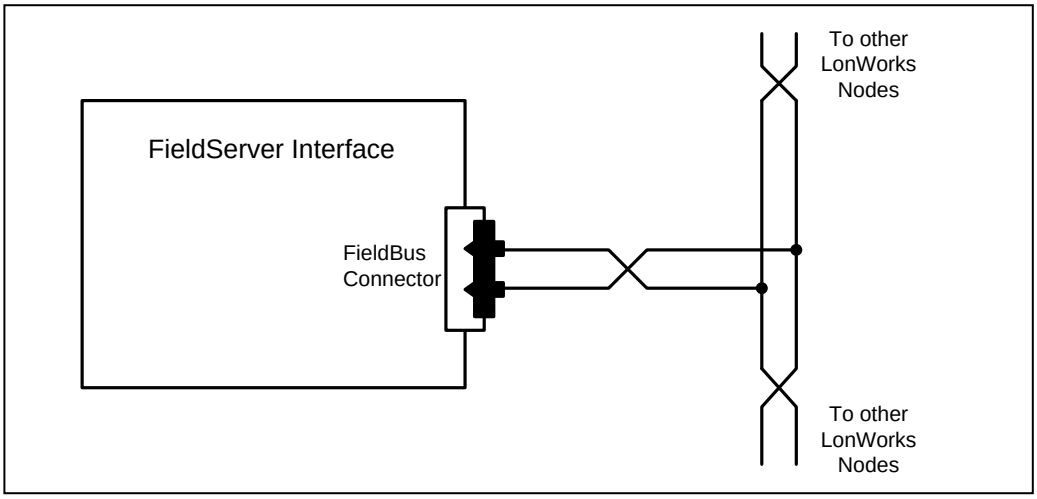
3.3 Additional Files Required

Filename	Auto-Created by driver	Always Loaded by Sierra Monitor	Conditionally loaded	Comments
Config.csv		X		Template loaded if no custom configuration ordered, otherwise Custom configuration loaded.
fserver.xif	X			Generated at power up. Changes if profile in configuration changes.
lon.ini			X	Used for providing default DSN. Refer to **Section 7.1.4**.
lonvars.cfg	X			Maintains commissioning settings. If corrupted, delete it, and reboot FieldServer to create new one.

4 HARDWARE CONNECTIONS

The screw plug supplied with the FieldServer LonWorks interface will connect the FieldServer to the FTT-10 network. The 2-wire twisted-pair connection to the network is polarity insensitive. The FTT-10 network is a free topology network which supports star, loop and/or bus wiring connections. Refer to the FTT-10A Free Topology Transceiver User's Guide from Echelon Corporation for recommended wiring practices if necessary. Additional information on cabling and junction boxes that may be used in twisted pair LonWorks networks are detailed in the following Echelon publication: http://www.echelon.com/support/documentation/bulletin/005-0023-01O_Jbox_wiring.pdf.

The connection diagram below shows how the FS-B30 is connected to a LonWorks network:



NOTE: Configure the remote LonWorks devices according to manufacturer's instructions.

4.1 Service Pin

The service pin is used to commission the FieldServer onto the network using a LonWorks Network Management tool.

4.2 Neuron ID

The driver prints the Neuron ID on the Driver Message Screen as its starts. On all platforms, the driver can store the Neuron ID in a Data Array so that it can be read by a Client node – refer also to **Appendix C.5**.

4.3 Lon LED's

The location of the Lon LED varies between FieldServers. Consult specific FieldServer start-up guides for details. The Lon LED is consistent with the table below for all FieldServers. Note that the LED does not indicate Lon communications, but is a Service LED coupled with the Service Pin which indicates whether the LonWorks interface on the FieldServer is configured or not. It asks for service when flashing and indicates a problem when flashing as described below:

- The LED is solid ON when the neuron is blank or has a hardware failure.
- The LED blinks as 0.5s intervals when the neuron has not been configured with address information (no Lon DCC loaded).
- The LED is OFF when a Lon DCC is loaded.

5 DATA ARRAY PARAMETERS

Data Arrays are "protocol neutral" data buffers for storage of data to be passed between protocols. It is necessary to declare the data format of each of the Data Arrays to facilitate correct storage of the relevant data. More information is available in **Appendix D**.

Section Title		
Data_Arrays		
Column Title	**Function**	**Legal Values**
Data_Array_Name	Provide name for Data Array.	Up to 15 alphanumeric characters
Data_Array_Format	Provide data format. Each Data Array can only take on one format.	Float, Bit, UInt16, SInt16, IInt32, SInt32 (**Appendix D**)
Data_Array_Length	Number of Data Objects. Must be larger than the data storage area required by the Map Descriptors for the data being placed in this array.	1-10, 000

Example

```
//    Data Arrays
Data_Arrays
Data_Array_Name    , Data_Array_Format    , Data_Array_Length
DA_AI_01           , UInt16,              , 200
DA_AO_01           , UInt16               , 200
DA_DI_01           , Bit                  , 200
DA_DO_01           , Bit                  , 200
```

6 IMPLICIT CONFIGURATIONS

6.1 Special Considerations

Implicit addressing is used when a Network Management Tool such as LonMaker® is used to connect a FieldServer to other LonWorks nodes - the FieldServer is assigned its data transfer (binding) parameters by the Network Management Tool.

6.1.1 Map Descriptor Names (Network Variable Names)

When using Implicit configuration, the LonWorks driver provides a table of Network Variables for other Nodes to read. These Nodes access the variables according to Index numbers which are automatically assigned by the driver according to the Map Descriptor definition order in the configuration file. The driver also creates an identification table for configuration nodes to read the name and types of variables available on the FieldServer. Within the LonWorks protocol, this table size is limited to 64K bytes. If the application requires a large number of Server mappings (up to 4096), it will be necessary to limit the variable name length. Each Server mapping requires 4 bytes plus the length of the name from this 64K block.

6.1.2 Implicit LonWorks Functions

Implicit Map Descriptors are distinguished from Explicit Map Descriptors by the function used in the Lon_Function field of the Map Descriptor. Depending on the nature and direction of Data Transfer needed, different implicit functions are available as follows (function field MUST pair up with the Lon_Function field as shown in the table below).

Lon_Function	Description	Function Field
NVUI	Network Variable Update Input.	PASSIVE
NVPO	Network Variable Polled Output.	
NVPIIMC	Network Variable Polled Input Implicitly addressed Continuously sent.	RDBC
NVUOIMC	Network Variable Update Output Implicitly addressed Continuously sent.	WRBC
NVUOIMX	Network Variable Update Output Implicitly addressed and sent on Change.	WRBX
NVUOIMT	Network Variable Update Output Implicitly addressed sent in Throttling Mode.	WRBC (**Appendix A.3**)
CFG_NVUI	Configuration Network Variable Update Input.	PASSIVE

6.1.3 SNVT_Index

For implicitly addressed Client Map Descriptors and all Server side Map Descriptors the driver will assign an index to each one, starting from 0 and incrementing to 4095, in the order they appear in the configuration file.

6.1.4 Address Table Limit

It is important to note the limitation within LonWorks Implicit addressing that the Address Table used for storing the addresses of all active connections made with the FieldServer is limited to 16 Device Addresses including that of the FieldServer. This is a LonWorks limitation, not a FieldServer limitation, and it means that the FieldServer can only make active implicit connections to 15 other devices. If active connections to more than 15 devices are required, then it is necessary to use Explicit Configuration (NMFETCHC is recommended).

Devices that write to passive Map Descriptors in the FieldServer only are not placed in this table and therefore this limitation does not apply. (A Passive Map Descriptor is recognized by the Passive Parameter in the Function Field).

6.1.5 Network Management Capability

When commissioning an implicitly configured FieldServer using the "Traditional" binding method, the configuration parameters below are all that is needed to achieve this purpose. For hints and tips on how to commission the FieldServer using the various Vendor devices refer to **Appendix B**.

It is possible to set the FieldServer Online and Offline using the Network Management Tools. This is a LonWorks function, and is different from the traditional Online and Offline Status discussed in other FieldServer documentation. When the FieldServer is set offline using this LonWorks feature it will not communicate with other LonWorks devices.

6.1.6 Client vs Server Mapping

Distinguishing Client from Server in the LonWorks Environment is more difficult due to the "Peer to Peer" nature of LonWorks design. Clients and Servers can be identified in this environment as follows:

- Clients will write Setpoints and have status written to them, so expect Setpoints and Control Points on the Output side, and Status and Alarms on the Input side.
- Servers will have the opposite structure, with Setpoints and Control Points on the Input Side and Staus and Alarms on the Output side.

Distinguishing Client from Server is important to ensure correct matching with other protocols being used on a FieldServer. For example, it would not be useful to map an Output to a BACnet AI point as both points will be expecting to push data out of the FieldServer.

NOTE: In the tables below, * indicates an optional parameter, with the bold legal value being the default.

6.2 Client Side Mapping

6.2.1 Client Side Connection Parameters

Section Title
Connections

Column Title	Function	Legal Values
Protocol	Specify protocol used	LonWorks

Example

```
//  Client Side Connections

Connections
Adapter         , Protocol
LonWorks        , LonWorks
```

6.2.2 Client Side Node Parameters

Section Title
Nodes

Column Title	Function	Legal Values
Node_Name	Provide name for Node.	Up to 32 alphanumeric characters
Protocol	Specify protocol used.	LonWorks
Startup_Write	Initiate a send on startup. Refer to **Appendix A.3** for more information.	Enabled, **Disabled**

Example

```
//  Client Side Nodes

Nodes
Node_Name    , Protocol
LON_1        , LonWorks
```

6.2.3 Client Side Map Descriptor Parameters

6.2.3.1 FieldServer Related Map Descriptor Parameters

Column Title	Function	Legal Values
Map_Descriptor_Name	Name of this Map Descriptor. (Network Variable Name)	Up to 16 alphanumeric characters
Data_Array_Name	Name of Data Array where data is to be stored or fetched from in the FieldServer.	One of the Data Array names defined in **Section 5**
Data_Array_Offset	Starting location in Data Array.	0 to (Data_Array_Length-1) as specified in **Section 5**
Lon_Function	Detailed Function of Client Map Descriptor.	See **Section 6.1.2** for a list of valid entries for Implicit Connections
Function	Function of Client Map Descriptor.	See **Section 6.1.2** for a list of valid entries based on the Lon_Function

6.2.3.2 Driver Related Map Descriptor Parameters

Column Title	Function	Legal Values
Node_Name	Name of Node to fetch data from or send data to. Needed for Explicit Addressing.	One of the Node names specified in **Section 6.2.2**
SNVT_Type	Standard network variable type of the data.	**Appendix D**
SNVT_Option*	Specifies which record out of a complex SNVT we are after.	**Appendix A.9.1**
SNVT_Units*	Specifies the measurement units (e.g. Metric or English).	**Appendix A.9.2**
UNVT_Byte_Length*	Specifies the length in bytes of an UNVT. This Parameter is compulsory when SNVT_Type is set to UNVT, the parameter is not used in conjunction with SNVT's.	1-255 (**Appendix A.1**)
UNVT_ID	Specifies the Type Number to apply to this particular UNVT. This Parameter is optional when SNVT_Type is set to UNVT, the parameter is not used on conjunction with SNVT's.	**0**, 146-255 (**Appendix A.1**)
NV_Min_Value*	Network Variable updates with values less than specified in this field will be ignored.	**Appendix A.2**
NV_Max_Value*	Network Variable updates with values higher than specified in this field will be ignored.	**Appendix A.2**
NV_Delta_Value*	Network Variable updates where the value's delta change is less than specified in this field will be ignored unless the Max_Scan_Time Condition has been met. Compulsory for Throttling Lon Functions, used on other Lon Functions. It is possible to configure a start-up write option for throttling Map Descriptor functions instead of waiting for the minimum scan time to pass. Refer to **Appendix A.3**.	**Appendix A.2**

NV_Selfdoc_Text*	The Self-Documenting String for this Network Variable.	Normal Text up to a length of 40 characters; defaults to **""** for parameters using string values and **0** for parameters using numeric values

6.2.3.3 Timing Parameters

Column Title	Function	Legal Values
Scan_Interval	Seconds per scan.	0-32000
Min_Scan_Time	Minimum Scan Time in seconds.	**Appendix A.3**
Max_Scan_Time	Maximum Scan Time in seconds.	**Appendix A.3**

6.2.4 Client Side Map Descriptor Example

```
//   Client Side Map Descriptors

Map_Descriptors
Map_Descriptor_Name   , Data_Array_Name   , Data_Array_Offset   , Lon_Function   , Function   , Node_Name   , SNVT_Type
TempOut1              , DA_FO_01          , 0                  , NVUOIMT        , Wrbc       , LON_1       , SNVT_Temp_f
TempOut2              , DA_FO_01          , 1                  , NVUOIMX        , Wrbc       , LON_1       , SNVT_Temp_f
TempOut3              , DA_FO_01          , 2                  , NVUOIMC        , Wrbc       , LON_1       , SNVT_Temp_f
TempIn                , DA_FI_01          , 0                  , NVPIIMT        , Rdbc       , LON_1       , SNVT_Temp_f
```

This field determines the Map Descriptor's function. See **Section 6.1.2** for more information.

Required for Throttling Mode: This field specifies the minimum change needed for a Network Variable's value to invoke an NV Update.

```
, Nv_Delta_Value   , Min_Scan_Time   , Max_Scan_Time   , Scan_Interval
, 10               , 5               , 20              , -
, -                , -               , -               , -
, -                , -               , -               , 20
, -                , -               , -               , 5
```

The Data Array will be sampled and evaluated for delta change once every Min_Scan_Time (seconds).

If found during Min_Scan_Time sampling (Throttling Mode) that Max_Scan_Time has elapsed since the last NV Update, a new NV Update will be transmitted.

For Non-Throttled Client Map Descriptors the Scan_Interval determines the update rate.

7 EXPLICIT CONFIGURATIONS

7.1 Special Considerations

With Explicit configuration, the FieldServer communicates directly with other devices by specifying the remote device address in the FieldServer configuration.

All the Servers and their Clients need to be on the same domain to be able to communicate. If the LonWorks device's Node ID is not known, it may be read using a utility program e.g. NodeUtil or using a LonWorks Analyzer. All the Client Node CSV files need to have their Subnet_ID and Node_ID parameters set to point to their respective Server Subnet and Node ID values. Once the FieldServers are restarted the network should function correctly.

7.1.1 Map Descriptor Names (Network Variable Names)

As a Server, the LonWorks driver provides a table of Network Variables for other Nodes to read. These Nodes access the variables according to Index numbers which are automatically assigned by the driver according to the Map Descriptor definition order in the configuration file. The driver also creates an identification table for configuration nodes to read the name and types of variables available on the FieldServer. Within the LonWorks protocol, this table size is limited to 64K bytes. If the application requires a large number of Server mappings (up to 4096), it will be necessary to limit the variable name length. Each Server mapping requires 4 bytes plus the length of the name from this 64K block.

7.1.2 LonWorks Functions

Explicit Map Descriptors are distinguished from Implicit Map Descriptors by the function used in the Lon_Function field of the Map Descriptor. Depending on the nature and direction of Data Transfer needed different explicit functions are available as shown below.

NOTE:

- Function field MUST pair up with the Lon_Function field as shown in the table below.
- NMFETCHC is the preferred Lon Function for most applications. (**Section 7.1.5**)

Lon_Function	Description	Function Field
NVUI	Network Variable Update Input.	SERVER
NVPO	Network Variable Polled Output.	
NVPIEXC	Network Variable Polled Input Explicitly addressed Continuously sent.	RDBC
NVUOEXC	Network Variable Update Output Explicitly addressed Continuously sent.	WRBC
NVUOEXX	Network Variable Update Output Explicitly addressed and sent on Change.	WRBX
NVUOEXT	Network Variable Update Output Explicitly addressed sent in Throttling Mode.	WRBC
NMFETCHC	Network Management Network Variable Fetch Continuously sent (Explicit Addressing only).	RDBC
CFG_NVUI	Configuration Network Variable Update Input.	SERVER

7.1.3 SNVT_Index

The SNVT_Index field must be specified for all explicitly addressed Client Map Descriptors. These Client Map Descriptors use this field to index their relevant Map Descriptors on the Server side.

7.1.4 Domain Table Setup

To be able to communicate with other Nodes on the LonWorks network, the FieldServer must have its Domain, Subnet and Node ID's set. Explicit configuration requires the user to set these parameters.

There are 3 ways in which the FieldServer DSN (Domain, Subnet, Node address) can be set:

- Set the DSN using a Network manager by commissioning the FieldServer into the Network Manager Project. When doing it this way, it is extremely important that the DSN is not forced in the FieldServer Configuration, as the Configuration address will override the Network Manager Address every time the FieldServer is restarted.
- Set the DSN from the Lon.ini file. Using this option provides a default DSN if the FieldServer does not have a DSN assigned. If a Network Manager or the FieldServer Configuration provides a DSN, then the Lon.ini DSN will be ignored.
- Set the DSN in the FieldServer configuration. This option takes precedence over the 2 other addressing options, and should therefore **not** be used when commissioning a FieldServer into a Network with a Network Manager.

Example 1: Setting the Domain and Subnet from the lon.ini file

The Domain Table and Subnet IDs can be set from a lon.ini file on the FieldServer. The FieldServer's LonWorks Node ID will be set from the System_Station_Address field in the configuration file if specified; otherwise a default value will be used.

lon.ini file contents:

```
:D50:S01:
```

Example 2: Setting the Node, Domain and Subnet in the configuration file

```
FieldServer
System_Station_Address    , Title
[Node_Id]                 , :D[Domain_ID]:S[Subnet_ID]:[Title continued…]
```

The Title field must start with ":D", followed by the Domain_ID in hexadecimal notation, followed by ":S", followed by the Subnet_ID in hexadecimal notation. The domain length is automatically determined by the number of digits in the [Domain_ID] field. Since 2 hexadecimal digits constitute 1 byte, ":D123456:" for example would have a length of 3.

Specifying the Domain and Subnet in the configuration file will override the settings from the lon.ini file.

7.1.5 Network Management

The NVPIEXC command generally used for polling has the limitation that data can only move in one direction (Server to Client). The added functionality of a Write-through was added to the network management fetch command (NMFETCHC) to provide movement of data back to the polled node. Thus, should the data in the Data Array linked to a NMFETCHC Map Descriptor change by another process, the data will be propagated to the polled node by means of a network variable update.

Another useful feature of the NMFETCHC function is that it can perform reads to any other type of Node function (NVUI, CFG_NVUI, NVUOEXX, NVPIIMC, etc.). This means that even Client Map Descriptors can be read this way.

Network management fetch requests received on polled nodes are handled differently from normal NV Poll requests. The response to a NV Poll will contain valid data. Data is valid if the node that is the source of the data is online and if this data is not too old. The data in the response to a network management fetch will not be checked for validity, however, it will simply respond with the current data in the variable.

7.2 Client Side Configuration

7.2.1 Client Side Connection Parameters

Section Title
Connections

Column Title	Function	Legal Values
Adapter	Adapter Name.	LonWorks
Protocol	Specify protocol used.	LonWorks
Default_Service*	Specify messaging service used.	**Ack,** UnAck[1]

Example

```
//    Client Side Connections

Connections
Adapter          , Protocol      , Default_Service
LonWorks         , LonWorks      , Ack
```

[1] Only applies to explicit addressing mode. Use Network Management Tool to set desired service type for implicit messaging. UnAck Service is commonly used to achieve quicker network variable update rates, but message delivery is not guaranteed.

7.2.2 Client Side Node Parameters

Section Title
Nodes

Column Title	Function	Legal Values
Node_Name	Provide name for Node.	Up to 32 alphanumeric characters.
Subnet_ID	Server (destination) Node's Subnet ID of an explicitly addressed message.	1-255
Node_ID	Server (destination) Node's Node ID of an explicitly addressed message.	1-127
Protocol	Specify protocol used.	LonWorks
Startup_Write	Initiate a send on startup. Refer to **Appendix A.3** for more information.	Enabled, **Disabled**
NM_Extended_Commands*	Used to enable Wink install type network management messages on network variable updates from NMFETCHC function Write-Throughs. Certain devices only allow Wink install messages updating network variables.	Enabled, **Disabled**
Readback_Option*	This Client Side parameter enables the user to configure the timing of a read after a write. The Readback operation will apply to all drivers that support Active Reads and Write-Through operations. **Readback_Asynchronously** - When a write occurs, the read will occur when scheduled. **Readback_On_Write** - When a write occurs, set the timer to 0, so Responsible Map Descriptor gets queued in the next cycle. **Readback_Immediately_On_Write** - Prioritize both write and read to happen in a higher priority queue than normal reads. The Readback operation will apply to all drivers that support Active Reads and Write-Through operation.	**Readback_ Asynchronously**, Readback_ Asynchronously, Readback_Immediately _On_Write

Example

```
//    Client Side Nodes

Nodes
Node_Name  , Subnet_ID  , Node_ID  , Protocol    , NM_Extended_Commands  , Readback_Option
LON_1      , 1          , 1        , LonWorks    , Enabled               , Readback_Asynchronously
```

7.2.3 Client Side Map Descriptor Parameters

7.2.3.1 Driver Related Map Descriptor Parameters

Column Title	Function	Legal Values
Node_Name	Name of Node to fetch data from or send data to. Needed for Explicit Addressing.	One of the Node Names specified in **Section 7.2.2**.
SNVT_Index	Server Network Variable Index.	0-4095
SNVT_Type	Standard network variable type of the data.	**Appendix D**
SNVT_Option*	Specifies which record out of a complex SNVT we are after.	**Appendix A.9.1**
SNVT_Units*	Specifies the measurement units when units other than Metric are required.	**Appendix A.9.2**
UNVT_Byte_Length*	Specifies the length in bytes of an UNVT. This Parameter is compulsory when SNVT_Type is set to UNVT, the parameter is not used on conjunction with SNVT's.	1-255 (**Appendix A.1**)
UNVT_ID	Specifies the Type Number to apply to this particular UNVT. This Parameter is optional when SNVT_Type is set to UNVT. The parameter is not used on conjunction with SNVT's.	**0,** 146-255 (**Appendix A.1**)
NV_Min_Value*	Network Variable updates with values less than specified in this field will be ignored.	**Appendix A.2**
NV_Max_Value*	Network Variable updates with values higher than specified in this field will be ignored.	**Appendix A.2**
NV_Delta_Value*	Network Variable updates where the value's delta change is less than specified in this field will be ignored unless the Max_Scan_Time Condition has been met. Compulsory for Throttling Lon Functions, not used on other Lon Functions. It is possible to configure a start-up write option for throttling Map Descriptor functions instead of waiting for the minimum scan time to pass. Refer to **Appendix A.3**.	**Appendix A.2**
NV_Selfdoc_Text*	The Self-Documenting String for this Network Variable.	Normal Text up to a length of 40 characters. Defaults to **""** for parameters using string values and **0** for parameters using numeric values.

7.2.3.2 Timing Parameters

Column Title	Function	Legal Values
Scan_Interval*	Seconds per scan.	0-32000, **2s**
Min_Scan_Time*	Minimum Scan Time in seconds. Compulsory for Throttling Lon Functions only. Not used on other Lon Functions.	**Appendix A.3**
Max_Scan_Time*	Maximum Scan Time in seconds. Compulsory for Throttling Lon Functions only. Not used on other Lon Functions.	**Appendix A.3**

7.2.4 Map Descriptor Examples

7.2.4.1 Explicit Config Using Network Management Polls

The Map Descriptor pollTemp_p shows the use of SNVT_Units for Scaling when polling temperature
The Map Descriptor pollSwitch shows the use of SNVT_Option for Selective data scanning. This only brings in the "State" portion of SNVT_Switch.

Map_Descriptors					
Map_Descriptor_Name	, Data_Array_Name	, Data_Array_Offset	, Lon_Function	, Function	, Node_Name
pollTemp_p	, DA_Explicit	, 1	, NMFETCHC	, Rdbc	, Remote_Node
pollSwitch	, DA_Explicit	, 2	, NMFETCHC	, Rdbc	, Remote_Node

This field determines the Map Descriptor's function. See **Section 7.2.3** for information.

, SNVT_Index	, SNVT_Type	, SNVT_Units	, SNVT_Option	, Scan_Interval
, 0	, SNVT_temp_p	, Deg_F	, -	, 1.0s
, 1	, SNVT_switch	, -	, State	, 1.0s

Explicitly addressed Client Map Descriptors use this field to extract the correct variable from the remote device's external interface.

Explicitly Addressed Client Map Descriptors use this field to point to their relevant Server Map Descriptors.

7.2.4.2 Polling UNVT's

Map_Descriptors				
Map_Descriptor_Name	, Data_Array_Name	, Data_Array_Offset	, Lon_Function	, Function
pollUNVT	, DA_Explicit	, 101	, NMFETCHC	, Rdbc

, Node_Name	, SNVT_Index	, SNVT_Type	, UNVT_Byte_Length	, Scan_Interval
, Remote_Node	, 2	, UNVT	, 31	, 1.0s

UNVT Byte length must be specified for correct operation.

7.2.5 Using Neuron ID to Address Devices (Client Drivers only)

It is possible to poll remote LonWorks devices using their Neuron ID's instead of the traditional Network Manager method. The Neuron ID is the 12 Digit Hex identifier of the associated LonWorks Device. An example configuration follows.

Note that Neuron ID addressing is considered explicit addressing so the station address, domain ID and subnet ID must be set as described in **Section 7.1.4**.

Neuron ID addressing is independent of domains, subnet, nodes and can therefore reach any device connected to the LonWorks network. The FieldServer needs to be commissioned with a Network Management tool for the routers to know it's there, and then neuron based packets will be passed through the router.

```
Bridge
System_Station_Address    , Title
110                       , D01:S01:Neuron ID Example:
```

On the Client side add a new field under Nodes as shown below:

```
Nodes
Node_Name    , Node_ID    , Protocol     , Neuron_ID
Lon_Srv_11   , 11         , LonWorks     , F912AB440100
```

Create a Map Descriptor as Follows:

```
Map_Descriptors
Map_Descriptor_Name      , Data_Array_Name      , Data_Array_Offset      , Lon_Function      , Function
1nviAmp1                 , DA_001               , 0                      , NMFETCHC          , RDBC
```

```
, Node_Name        , SNVT_Index       , SNVT_Type        , Scan_Interval
, Lon_Srv_11       , 000              , SNVT_amp         , 0s
```

7.3 Server Side Configuration

7.3.1 Server Side Connection Parameters

Section Title
Connections

Column Title	Function	Legal Values
Adapter	Adapter Name.	LonWorks
Protocol	Specify protocol used.	LonWorks

Example

```
//    Server Side Connections

Connections
Adapter          , Protocol
LonWorks         , LonWorks
```

7.3.2 Server Side Node Parameters

Section Title
Nodes

Column Title	Function	Legal Values
Node_Name	Provide name for Node.	Up to 32 alphanumeric characters
Protocol	Specify protocol used.	LonWorks
Startup_Write	Initiate a send on startup.	Enabled, **Disabled**

Example

```
//    Server Side Nodes

Nodes
Node_Name    , Protocol
LON_1        , LonWorks
```

7.3.3 Server Side Map Descriptor Parameters

7.3.3.1 FieldServer Related Map Descriptor Parameters

Column Title	Function	Legal Values
Map_Descriptor_Name	Name of this Map Descriptor.	Up to 16 alphanumeric characters
Data_Array_Name	Name of Data Array where data is to be stored or fetched from in the FieldServer.	One of the Data Array names defined in **Section 5**
Data_Array_Offset	Starting location in Data Array.	0 to (Data_Array_Length-1) as specified in **Section 5**
Lon_Function	Detailed Function of Client Map Descriptor.	See **Section 7.1.2** for a list of valid entries
Function	Function of Client Map Descriptor.	

7.3.3.2 Driver Related Map Descriptor Parameters

Column Title	Function	Legal Values
Node_Name	Name of Node to fetch data from or send data to. Needed for Explicit Addressing.	One of the node names specified under Client Node Parameters
SNVT_Index	Server Network Variable Index.	0-4095
SNVT_Type	Standard network variable type of the data.	**Appendix D**
SNVT_Option	Specifies which record out of a complex SNVT we are after.	**Appendix A.9.1**
SNVT_Units	Specifies the measurement units if units other than Metric is required.	**Appendix A.9.2**
UNVT_Byte_Length*	Specifies the length in bytes of an UNVT. This Parameter is compulsory when SNVT_Type is set to UNVT, the parameter is not used on conjunction with SNVT's.	1-255 (**Appendix A.1**)
UNVT_ID	Specifies the Type Number to apply to this particular UNVT. This Parameter is optional when SNVT_Type is set to UNVT, the parameter is not used on conjunction with SNVT's.	**0,** 146-255 (**Appendix A.1**)
NV_Min_Value*	Network Variable updates with values less than specified in this field will be ignored.	**Appendix A.2**
NV_Max_Value*	Network Variable updates with values higher than specified in this field will be ignored.	**Appendix A.2**
NV_Delta_Value*	Network Variable updates where the value's delta change is less than specified in this field will be ignored unless the Max_Scan_Time Condition has been met. Compulsory for Throttling Lon Functions only. Not used on other Lon Functions.	**Appendix A.3**
NV_Selfdoc_Text	The Self-Documenting String for this Network Variable.	Normal Text up to a length of 40 characters. Defaults to **""** for parameters using string values and **0** for parameters using numeric values.
Bad_Value_Validation*	Set to Disabled to allow the out of range value.	**Enabled**, Disabled

7.3.3.3 Timing Parameters

Column Title	Function	Legal Values
Scan_Interval*	Seconds per scan.	0-32000, **2s**
Min_Scan_Time*	Minimum Scan Time in seconds. Compulsory for Throttling Lon Functions only. Not used on other Lon Functions.	**Appendix A.3**
Max_Scan_Time*	Maximum Scan Time in seconds. Compulsory for Throttling Lon Functions only. Not used on other Lon Functions.	**Appendix A.3**

7.3.4 Map Descriptor Example

```
// Server Side Map Descriptors
Map_Descriptors
Map_Descriptor_Name   , Data_Array_Name , Data_Array_Offset , Lon_Function , Function , Node_Name , SNVT_Index
NviTemp               , DA_FI_01        , 0                , NVUI         , Server   , LON_1     , -
NviTemp               , DA_FO_01        , 1                , NVPO         , Server   , LON_1     , -
NviTemp               , -               , -                , CFG_NVUI     , Server   , LON_1     , -
```

Server NVUI can be grouped with Client NV Update types and NMFETCHC. Server NVPO can be grouped with Client NV Poll types and NMFETCHC.

Explicitly Addressed Client Map Descriptors use this field to point to their relevant Server Map Descriptors.

```
, Nv_Delta_Value , SNVT_Type    , Nv_Min_Value , Nv_Max_Value , Nv_Selfdoc_text
, 10             , SNVT_Temp_f  , 0            , -            , "@0|1;WaterTemp"
, -              , SNVT_Temp_f  , 0            , 150          , "&200\x8023"
, -              , SNVT_Temp_f  , -            , 150          , "&200\x8020
```

Required for Throttling Mode: This field specifies the minimum change needed for a Network Variable's value to invoke an NV Update.

If specified for a Floating point or Integer SNVT then NV Updates with values higher than specified here will not be written to its relevant Data Array location.

If specified for a Floating point or Integer SNVT then NV Updates with values lower than specified here will not be written to its relevant Data Array location.

The Self-documenting String for a Network Variable can be up to 40 characters wide.

APPENDIX A USEFUL FEATURES

Appendix A.1 Using UNVT's

UNVT's are User-defined Network Variable Types and are defined in the configuration file by filling the SNVT_Type field with "UNVT". The UNVT's size in bytes should also be added to the UNVT_Byte_Length field and the UNVT's type number in the UNVT_ID field. Valid UNVT_IDs are 0 and the range 146 to 255. The UNVT_ID value will be substituted with the default value of 0 if the field is left empty. Refer to the example in **Section** 7.2.3.

Appendix A.2 Using NV_Min, NV_Max, Etc.

The NV_Min_Value, NV_Max_Value functionality were added to stop the propagation of 'out of bounds' network variables across a LonWorks network. NV values higher than assigned by the NV_Max_Value field or lower than assigned by NV_Min_Value field will not be sent out on the network or copied to their relevant Data Arrays. If one or both of these fields are not assigned values, limit checking will not be performed on that particular field. This functionality applies only to SNVT's of either Floating point or Integer formats. Refer to the examples in **Section** 6.2.4 and Appendix A.6.1.

One or both of these fields can be linked to a configuration network variable update input in order to create an interoperable configuration property. For further information refer to Appendix A.5.

Appendix A.3 Throttling Mode (Send Heartbeat)

When the Lon_Function field specifies a Throttling Mode Client Map Descriptor the NV_Delta_Value, Min_Scan_Time and Max_Scan_Time fields all need to be assigned values.

When operating in Throttling Mode, a Network Variable's value will be sampled at the Min_Scan_Time period. If the value differs by an amount greater than or equal to that specified by the NV_Delta_Value field from the previously transmitted value, the NV update will be transmitted. Otherwise an NV update will be only sent if the time since the last update is greater than or equal to the Max_Scan_Time field value. Refer to the examples in **Sections** 6.2.4.

This functionality applies only to SNVT's of either Floating point or Integer formats.

The driver can be configured to initiate a send on startup rather than waiting for the minimum scan interval to pass. The following example describes the configuration required:

```
Nodes
Node_Name   , Node_ID   , Protocol    , Startup_Write
Lon_Srv_11  , 11        , LonWorks    , Enabled
```

```
Map_Descriptors
Map_Descriptor_Name , Data_Array_Name , Data_Array_Offset , Lon_Function , Function , Node_Name
nvoAmp              , DA_001          , 0                , NVUOEXT      , WRBC     , Lon_Srv_11
```

```
, SNVT_Index  , SNVT_Type  , Min_Scan_Time  , Max_Scan_Time  ,NV_Delta_Value
, 000         , SNVT_amp   , 30s            , 60s            , 30
```

NOTE: Refer to Appendix A.5.1. To configure Send Heartbeat parameters dynamically the NV_Min_Send_Time, NV_Max_Send_Time and NV_Delta_Value should be used with Configuration Network Variables.

Appendix A.4 XIF Generation

After start-up the FieldServer creates a XIF (version 4.0) according to the configuration loaded. This file is called fServer.xif and can be uploaded from the FieldServer using the FieldServer FS-GUI.

To upload the XIF file using the FS-GUI:

- Enter the FieldServer IP Address followed by \fserver.xif into a web browser.
  - For example: 192.168.2.101\fserver.xif

The FieldServer's XIF file is internally generated by the LonWorks driver using the information contained in the FieldServer's configuration file (CONFIG.CSV). It is not possible to download an externally created XIF file using NodeUtil or LonMaker®. Additionally, it is also not possible to download any type of externally created network variables default values file. The FieldServer's configuration can be changed by uploading and editing the CONFIG.CSV file, and therefore the XIF file must be obtained by uploading it from the FieldServer.

NOTE: The XIF file will change whenever the configuration file has been changed, downloaded and the FieldServer restarted.

Appendix A.5 Configuration Properties

Configuration Properties are implemented on the FieldServer using configuration Network Variables. Configuration Network Variables take the form of update inputs in order to be updated or read by a network management tool. Each Configuration Network Variable will have its own Map Descriptor. Since the data carried by a Network Variable is intended as a configuration property for other already declared non-configuration Network Variable(s), no Data Array needs to be assigned to its Map Descriptor.

The following configuration property types are supported:

- Apply gain to Input and Output Network Variable values via SNVT_nv_type.
- Add Offset values to Input and Output Network Variable Values via SNVT_nv_type.
- Enforce Minimum and Maximum Range on Network Variables.
- Set Update Rate on Active Network Variables.
- Set Throttling Mode (Min/MaxSendTime, DeltaValue) properties on Active Network Variables.
- Allow adjusting of scan interval on active LonWorks Map Descriptors (SCPTupdateRate (98)).
- SCPTGain (31) for Modbus Translations via SNVT_muldiv.
- Allow changing of the NV_Delta_Value (SCPTsndDelta (27)).
- SCPTnvType that is mandatory on changeable type interfaces for FieldServer specific LonMark certification of the protocol gateway profile.

Configuration Properties can be declared to belong to the entire Node, to an Object(s) or to a Network Variable(s). This declaration defines the *scope* of the Configuration Property. Configuration Properties that belong to an Object are declared as a part of the Object; Configuration Properties associated with a Network Variable(s) are declared as belonging to the Network Variable(s) etc.

A Configuration Property's Type and Responsible Map Descriptor(s) are declared by the Configuration Network Variable's self-documentation string as follows:

For linking to the entire node:
"&0, , 0\x80, [SCPT Index]"
Example: "&0, , 0\x80, 20"

For linking to object(s):
"&1, [Object Index(s)], 0\x80, [SCPT Index]"
Example: "&1, 0.2.3, 0\x80, 23"

For linking to network variable(s):
"&2, [NV Index(s)], 0\x80, [SCPT Index]"
Example: "&2, 0.1.2.4.7, 0\x80, 20"

Appendix A.5.1 Configuration Examples

Appendix A.5.1.1 Config Network Variable Map Descriptors Applied to Output NV's

Map_Descriptors

Map_Descriptor_Name	Data_Array_Name	Data_Array_Offset	Lon_Function	Function	Node_Name	SNVT_Type	NV_Gain
nciOut01a	-	-	CFG_NVUI	Passive	Lon_Srv_1	SNVT_nv_type	-
nciOut01b	-	-	CFG_NVUI	Passive	Lon_Srv_1	SNVT_count_inc_f	-
nciOut01c	-	-	CFG_NVUI	Passive	Lon_Srv_1	SNVT_count_inc_f	-
nciOut01d	-	-	CFG_NVUI	Passive	Lon_Srv_1	SNVT_count_inc_f	-
nciOut01e	-	-	CFG_NVUI	Passive	Lon_Srv_1	SNVT_time_sec	-
nciOut01f	-	-	CFG_NVUI	Passive	Lon_Srv_1	SNVT_time_sec	-
nciOut01g	-	-	CFG_NVUI	Passive	Lon_Srv_1	SNVT_nv_type	-
nciOut01h	-	-	CFG_NVUI	Passive	Lon_Srv_1	SNVT_count_inc_f	-
nciOut01i	-	-	CFG_NVUI	Passive	Lon_Srv_1	SNVT_count_inc_f	-
nciOut01j	-	-	CFG_NVUI	Passive	Lon_Srv_1	SNVT_nv_type	-
nciOut01k	-	-	CFG_NVUI	Passive	Lon_Srv_1	SNVT_count	-
nciOut01l	-	-	CFG_NVUI	Passive	Lon_Srv_1	SNVT_count	-
nciOut01m	-	-	CFG_NVUI	Passive	Lon_Srv_1	SNVT_count	-
nciOut01n	-	-	CFG_NVUI	Passive	Lon_Srv_1	SNVT_time_sec	-
nciOut01o	-	-	CFG_NVUI	Passive	Lon_Srv_1	SNVT_time_sec	-
nciOut01p	-	-	CFG_NVUI	Passive	Lon_Srv_1	SNVT_nv_type	-
nciOut01q	-	-	CFG_NVUI	Passive	Lon_Srv_1	SNVT_count	-
nciOut01r	-	-	CFG_NVUI	Passive	Lon_Srv_1	SNVT_count	-
nciOut01s	-	-	CFG_NVUI	Passive	Lon_Srv_1	SNVT_muldiv	1.00

NV_Min_Value	NV_Max_Value	NV_Min_Send_Time	NV_Max_Send_Time	NV_Delta_Value	NV_Value_String	Nv_Selfdoc_text
-	-	-	-	-	"0 52 12 4 1 0 0"	"&2, 0, \x80, 254"
-3.40282e+038	-	-	-	-	" "	"&2, 2, 0\x80, 23"
-	3.40282e+038	-	-	-	" "	"&2, 2, 0\x80, 20"
-	-	-	-	1.00	" "	"&2, 2, 0\x80, 27"
-	-	-	10.00	-	" "	"&2, 2, 0\x80, 49"
-	-	1.00	-	-	" "	"&2, 2, 0\x80, 52"
-	-	-	-	-	"0 52 12 4 1 0 0"	"&2, 3, 0\x80, 254"
-3.40282e+038	-	-	-	-	" "	"&2, 3, 0\x80, 23"
-	3.40282e+038	-	-	-	" "	"&2, 3, 0\x80, 20"
-	-	-	-	-	"0 8 6 2 1 0 0 "	"&2, 4, 0\x80, 254"
0	-	-	-	-	" "	"&2, 4, 0\x80, 23"
-	65535	-	-	-	" "	"&2, 4, 0\x80, 20"
-	-	-	-	1.00	" "	"&2, 4, 0\x80, 27"
-	-	-	10.00	-	" "	"&2, 4, 0\x80, 49"
-	-	1.00	-	-	" "	"&2, 4, 0\x80, 52"
-	-	-	-	-	"0 8 6 2 1 0 0 "	"&2, 5, 0\x80, 254"
0	-	-	-	-	" "	"&2, 5, 0\x80, 23"
-	65535	-	-	-	" "	"&2, 5, 0\x80, 20"
-	-	-	-	-	" "	"&1, 1, 0\x80, 31"

Appendix A.5.1.2 Config Network Variable Map Descriptors Applied to Input NV's

Map_Descriptor_Name	Data_Array_Name	Data_Array_Offset	Lon_Function	Function	Node_Name	SNVT_Type	NV_Gain	NV_Min_Value
nciIn01a	-	-	CFG_NVUI	Passive	Lon_Srv_1	SNVT_nv_type	-	-
nciIn01b	-	-	CFG_NVUI	Passive	Lon_Srv_1	SNVT_count_inc_f	-	-3.40282e+038
nciIn01c	-	-	CFG_NVUI	Passive	Lon_Srv_1	SNVT_count_inc_f	-	-
nciIn01d	-	-	CFG_NVUI	Passive	Lon_Srv_1	SNVT_nv_type	-	-
nciIn01e	-	-	CFG_NVUI	Passive	Lon_Srv_1	SNVT_count_inc_f	-	-3.40282e+038
nciIn01f	-	-	CFG_NVUI	Passive	Lon_Srv_1	SNVT_count_inc_f	-	-
nciIn01g	-	-	CFG_NVUI	Passive	Lon_Srv_1	SNVT_time_sec	-	-
nciIn01h	-	-	CFG_NVUI	Passive	Lon_Srv_1	SNVT_nv_type	-	-
nciIn01i	-	-	CFG_NVUI	Passive	Lon_Srv_1	SNVT_count	-	0
nciIn01j	-	-	CFG_NVUI	Passive	Lon_Srv_1	SNVT_count	-	-
nciIn01k	-	-	CFG_NVUI	Passive	Lon_Srv_1	SNVT_nv_type	-	-
nciIn01l	-	-	CFG_NVUI	Passive	Lon_Srv_1	SNVT_count	-	0
nciIn01m	-	-	CFG_NVUI	Passive	Lon_Srv_1	SNVT_count	-	-
nciIn01n	-	-	CFG_NVUI	Passive	Lon_Srv_1	SNVT_time_sec	-	-
nciIn01o	-	-	CFG_NVUI	Passive	Lon_Srv_1	SNVT_muldiv	1.00	-

NV_Max_Value	NV_Min_Send_Time	NV_Max_Send_Time	NV_Delta_Value	NV_Value_String	Nv_Selfdoc_text
-	-	-	-	"0 52 12 4 1 0 0"	"&2, 6, 0\x80, 254"
-	-	-	-	" "	"&2, 6, 0\x80, 23"
3.40282e+038	-	-	-	" "	"&2, 6, 0\x80, 20"
-	-	-	-	"0 52 12 4 1 0 0"	"&2, 7, 0\x80, 254"
-	-	-	-	" "	"&2, 7, 0\x80, 23"
3.40282e+038	-	-	-	" "	"&2, 7, 0\x80, 20"
-	1.00	-	-	" "	"&2, 7, 0\x80, 98"
-	-	-	-	"0 8 6 2 1 0 0 "	"&2, 8, 0\x80, 254"
-	-	-	-	" "	"&2, 8, 0\x80, 23"
65535	-	-	-	" "	"&2, 8, 0\x80, 20"
-	-	-	-	"0 8 6 2 1 0 0 "	"&2, 9, 0\x80, 254"
-	-	-	-	" "	"&2, 9, 0\x80, 23"
65535	-	-	-	" "	"&2, 9, 0\x80, 20"
-	1.00	-	-	" "	"&2, 9, 0\x80, 98"
-	-	-	-	" "	"&1, 2, 0\x80, 31"

Appendix A.6 LonMark Object Creation

For more information on filling out the Node Self-Documentation String as well as the relevant Network Variable Self-Documentation Strings in order to create a LonMark Object, consult the following documentation from the LonMark Interoperability Association at www.lonmark.org:

- Understanding LonMark Self-Documentation
- LonMark Application Layer Interoperability Guidelines

The example below shows how LonMark Objects can be assigned in FieldServer configuration files.

In this example the standard Open Loop Sensor Object (Type 1) is used to create a Water Temperature Sensor that will send out its current status via Network Variable Updates.

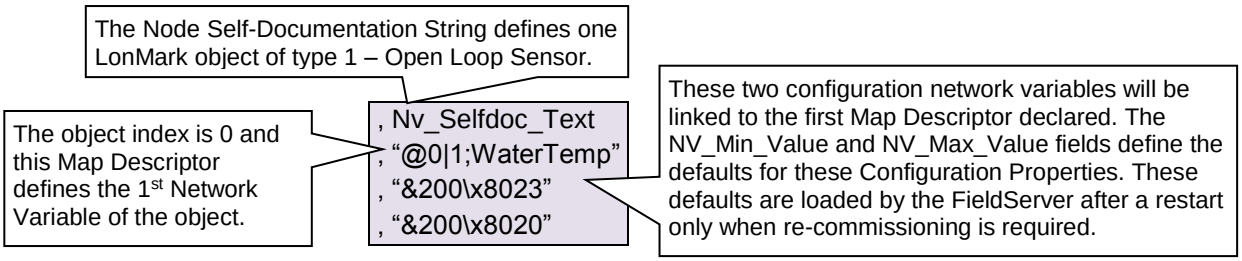
Appendix A.6.1. LonMark Config file Example – Open Loop "Water Temperature" Sensor

```
//    Client Side Map Descriptors

Map_Descriptors
Map_Descriptor_Name   , Data_Array_Name   , Data_Array_Offset   , Lon_Function   , Function   , Node_Name   , SNVT_Index
nvoTemp               , DA_FI_01           , 0                  , NVUOIMC        , Wrbc       , LON_1       , -
NviTempMin            , -                  , -                  , CFG_NVUI       , Server     , LON_1       , -
NviTempMax            , -                  , -                  , CFG_NVUI       , Server     , LON_1       , -
```

```
, SNVT_Type      , Nv_Min_Value    , Nv_Max_Value    , Nv_Selfdoc_Text
, SNVT_Temp_f    , -               , -               , "@0|1;WaterTemp"
, SNVT_Temp_f    , 10              , -               , "&200\x8023"
, SNVT_Temp_f    , -               , 300             , "&200\x8020"
```

The entries under Nv_Selfdoc_Text are described in more detail below:



Appendix A.7 Node Status Operation

When binding network variables on the FieldServer, the node status will be on-line before binding as well as after un-binding network variables. A Node will go off-line only when a bound network variable of that node goes off-line (e.g. is disconnected).

When using a network tool to commission the FieldServer (implicit addressing), the FieldServer LonWorks Node must be placed ONLINE on the LonWorks network before it will pass data with other bound network variables. The following messages may appear on the driver screen when using the FS-GUI which indicates that the FieldServer has not been put into the ONLINE state yet:

"Not sending NV msg for [variable name], Node OFFLINE!"

To set the FieldServer to ONLINE using LonMaker:

Right-click on the FieldServer device and select Manage.
On the Devices Tab, click the ONLINE button.

The FieldServer LonWorks Node is always ONLINE when using explicit addressing.

NOTE: The FieldServer Node_Status function (refer to the FieldServer Configuration Manual) looks at the Node_ID configured in order to monitor Node status. In implicit applications this may not be the real Node_ID since Node_ID is allocated by the network manager.

Appendix A.8 Working with Binary Data

When sending binary data on a LonWorks network, there are two SNVT's that are most commonly used, namely SNVT_State and SNVT_Switch.

SNVT_State is a 16 bit integer that allows the user to send 16 data bits to the remote device. To do this, the 16 data bits need to be packed into an integer word. If this packing is already done when the value is transferred into the FieldServer, then the SNVT need only be linked to the relevant Data Array offset for transfer out of the FieldServer. However, if the FieldServer is picking up 16 separate bits from another network for this variable, then the FieldServer Packed_Bit function can be used to achieve the packing. See the FieldServer Configuration manual for more details on the Packed_Bit function.

SNVT_Switch is a two byte value consisting of an analog value in the left hand byte (VALUE portion), and a binary state in the right hand byte (STATE portion). LonMark guidelines for this SNVT stipulate that the VALUE portion has a range from 0-100%, and that the STATE portion can assume a value of 1 or 0. However, since the raw data type for these two values is byte (a value ranging from 0-255), care needs to be taken when transferring values into this SNVT to make sure the source values are within the legal range. If a value is placed into one of these bytes that is deemed out of range, then a -1 value will be sent on the LonWorks network to indicate out of range. Also note that when sending binary status in a SNVT Switch, some devices expect the VALUE portion to change when the STATUS portion changes, so an analog value that changes in sympathy with the binary state may be necessary.

Appendix A.9 Filtering SNVT Variables

Appendix A.9.1. SNVT_Option Field

This field can be used to isolate a specific record within a complex SNVT. **Appendix D** shows which SNVTs are currently attributed with SNVT_Option capabilities.

Appendix A.9.2. SNVT_Units Field

The measurement units of a SNVT is fixed as far as it is transported across the LonWorks network, but by the setting of the SNVT_Units field it is possible to specify how the data is made available in Data Arrays. For example, SNVT_temp_f is always transported across the LonWorks network as degrees Celsius, but by setting the SNVT_Units field to Deg_F the measurement units of the Map Descriptor's data will be set to degrees Fahrenheit. The following table shows which SNVT's currently have SNVT_Units capabilities.

SNVT_Type	SNVT_Option	SNVT_Units	Length	Measurement Units
SNVT_flow	N/A	Liters/Second	1	Liters per Second
		Cubic-feet-per-minute	1	Cubic-feet-per-minute
		Gallons-per-minute	1	Gallons-per-minute
SNVT_flow_f	N/A	Liters/Second	1	Liters per Second
		Cubic-feet-per-minute	1	Cubic-feet-per-minute
		Gallons-per-minute	1	Gallons-per-minute
SNVT_press_f	N/A	Inch_H20	1	Inches of water
SNVT_press_p	N/A	Inch_H20	1	Inches of water
SNVT_temp	N/A	Raw	1	Raw, unadjusted data
		Deg_C	1	Degrees Celsius (°C)
		Deg_F	1	Degrees Fahrenheit (°F)
SNVT_temp_f	N/A	Deg_C	1	Degrees Celsius (°C)
		Deg_F	1	Degrees Fahrenheit (°F)
SNVT_temp_p	N/A	Raw	1	Raw, unadjusted data
		Deg_C	1	Degrees Celsius (°C)
		Deg_F	1	Degrees Fahrenheit (°F)
SNVT_switch[2]	All	Raw	2	Raw, unadjusted data
	Value	Raw	1	Raw, unadjusted data
		Percent	1	Percent (%)
	State	Raw	1	Raw, unadjusted data
SNVT_temp_setpt	All	Raw	12	Raw, unadjusted data
		Deg_C	6	Degrees Celsius (°C)
		Deg_F	6	Degrees Fahrenheit (°F)
	Occupied_cool	Raw	1	Raw, unadjusted data
		Deg_C	1	Degrees Celsius (°C)
		Deg_F	1	Degrees Fahrenheit (°F)
	Standby_cool	Raw	1	Raw, unadjusted data
		Deg_C	1	Degrees Celsius (°C)
		Deg_F	1	Degrees Fahrenheit (°F)
	Unoccupied_cool	Raw	1	Raw, unadjusted data
		Deg_C	1	Degrees Celsius (°C)
		Deg_F	1	Degrees Fahrenheit (°F)

[2] See also **Appendix A.8**.

SNVT_Type	SNVT_Option	SNVT_Units	Length	Measurement Units
SNVT_temp_setpt	Occupied_heat	Raw	1	Raw, unadjusted data
		Deg_C	1	Degrees Celsius (°C)
		Deg_F	1	Degrees Fahrenheit (°F)
	Standby_heat	Raw	1	Raw, unadjusted data
		Deg_C	1	Degrees Celsius (°C)
		Deg_F	1	Degrees Fahrenheit (°F)
	Unoccupied_ heat	Raw	1	Raw, unadjusted data
		Deg_C	1	Degrees Celsius (°C)
		Deg_F	1	Degrees Fahrenheit (°F)

Appendix A.9.3. Map Descriptor Example

Appendix A.9.1.1 Selecting Multiple Convert Units on SNVT_flow and SNVT_flow_f

```
//    Client Side Map Descriptors

Map_Descriptors
Map_Descriptor_Name     , Data_Array_Name   , Data_Array_Offset   , Lon_Function   , Function
nviFlow1                , DA_001            , 0                  , NVPIEXC         , RDBC
nviFlow1_f              , DA_001            , 1                  , NVPIEXC         , RDBC
nviFlow2                , DA_001            , 2                  , NVPIEXC         , RDBC
nviFlow2_f              , DA_001            , 3                  , NVPIEXC         , RDBC
```

The same Map Descriptor can be used on the Server Side but the Function must be changed to Passive.

```
, Node_Name     , SNVT_Index    , SNVT_Type     , SNVT_Units
, Lon_Srv_11    , 0             , SNVT_flow     , Cubic-feet-per-minute
, Lon_Srv_11    , 1             , SNVT_flow_f   , Cubic-feet-per-minute
, Lon_Srv_11    , 2             , SNVT_flow     , Gallons-per-minute
, Lon_Srv_11    , 3             , SNVT_flow_f   , Gallons-per-minute
```

SNVT_flow and SNVT_flow_f is always transported across the LonWorks network as liters/second, but by setting the SNVT_Units field the units of the Map Descriptor's data can be changed.

Appendix A.10 Accessing Remote Configuration Properties Implemented as Embedded Files

Appendix A.10.1. General Notes

A device's configuration properties (CPs) can be exposed as Network Variables (NVs) that can be viewed in LonMaker, or as embedded files that cannot be viewed in LonMaker.

Appendix A.10.2. Determining if NVs Are Used to Expose CPs

- View the XIF file with a text viewer.
- Search for lines starting with VAR.
- A first indication that it uses NVs is the name of the VAR that starts with nci (network configuration input).
- A second determination that NVs are used is the 2nd line below VAR (the self-documentation string line) that contains a '&' character after the " (e.g. "&1,0,0).
- Finally, at the end of the file there should be a NVVAL section that will give default values for all Configuration Properties exposed as Network Variables.

Appendix A.10.2.1. Accessing CPs Exposed as NVs

CPs exposed as NVs can be accessed either with:

- Implicit Addressing – where a network management tool such as LonMaker is simply used to bind to the network variable.
- Explicit Addressing – where it is accessed using the SNVT_Index.

Example: If the CP exposed as a NV is the 5th entry in the XIF file from the top, the SNVT_Index will be 4.

Looking at the complete VAR line of the CP to be accessed:
VAR nciDevMinVer 4 0 0 0 -- the number just after the name is always the SNVT_Index, in this case 4.

Appendix A.10.3. Determining if Embedded Files Are Used to Expose CPs

- View the XIF file with a text viewer.
- Search for lines starting with VAR.
- A first indication that embedded files are used is names such as nvoFileDirectory, and a second determination is the SNVT_Type number in the 3rd of 4th lines below the VAR line. This line will always have a number followed by a star and an additional number (e.g. 73 * 1).
- The first number is the SNVT_Type and these numbers indicate that embedded files are used:
  73 * (SNVT_file_req)
  74 * (SNVT_file_status)
  114 * (SNVT_address)
- Finally, at the end of the file there will be FILE sections to confirm embedded files are used.

Appendix A.10.3.1. Accessing CPs Exposed as Files[3]

Currently, the FieldServer driver can only access CPs exposed as files where the SNVT_address of the files are exposed, such as done by all the LonPoint modules. Therefore, the existence of the 114 * line in the XIF indicates that the device exposes the address of the files which the FieldServer driver will use to access the files.

STEP 1:
Add a map descriptor to the config file that will read the address of the embedded files.

Example from the DI-10V3.XIF file:
VAR Node_Directory **4** 0 0 0
0 1 63 1 0 1 0 1 0 0 0 0 0
"@0|8;Address offset of the node file directory. For tool-use" only 114 * 1

```
Map_Descriptors
Map_Descriptor_Name    , Data_Array_Name    , Data_Array_Offset    , Lon_Function
nviFDAddress           , File_Address       , 0                    , NMFETCHC

, Function    , Node_Name     , SNVT_Index    , SNVT_Type
, RDBC        , Lon_Remote    , 004           , SNVT_address
```

Note this value from the VAR line.

Note this value from the SNVT Type line.

In the following excerpt taken from the Echelon SNVTs document, 114 refers to SNVT_Address which is the Type of SNVT:

SNVT_address (114)

Neuron Chip Address

SNVT Index	Measurement	Type Category	Type Size
114	Neuron Chip Address	Unsigned Long	2 bytes
Valid Type Range	**Type Resolution**	**Units**	**Invalid Value**
16,384 .. 64,767	1	16-bit address value	64,768 (0xFD00)
Raw Range	**Scale Factors**	**File Name**	**Default Value**
16,384 .. 64,767 (0 .. 0xFCFF)	1, 0, 0 $S = a*10^{b}*(R+c)$	N/A	N/A

[3] **Warning:** This method uses direct memory write LonWorks messages that could corrupt a remote device's configuration property files if the incorrect memory address is determined by the outlined steps. Do not allow write-throughs to happen until the configuration property's value is correctly read and displayed in the FieldServer Data Array. Verify correct configuration property and values by adding the device to a network management tool and changing the values with the tool making sure the corresponding FieldServer Data Array values match before using the device in explicit messaging mode.

STEP 2:

Add a map descriptor that will read the first 16 bytes at the address provided by the 1st map descriptor. These 16 bytes are the file directory table of the device and indicate the types and addresses of the files that need to be accessed to read or write the Configuration Properties:

Map_Descriptors				
Map_Descriptor_Name	, Data_Array_Name	, Data_Array_Offset	, DA_Address	, Lon_Function
Poller 1	, CP_Info	, 0	, File_Address	, NM_MEMORY

The first 16 bytes of the file directory table will be stored in this data array.

NOTE: This is the data array from the 1st mapdesc that will give this mapdesc the address of the file directory table.

, Function	, Node_Name	, SNVT_Type	, UNVT_ID	, UNVT_Byte_Length
, RDBC	, Lon_Remote	, Not SNVT	, -	, 16

STEP 3:

Create a moves function to extract the address of the file where the Configuration Properties are located.

This table is for Version 1.1 File Table Format - the 1st byte of CP_Info shows the version:
0x11 = Version 1.1

Version 1.1 uses 1 byte for the file-type entries.

Moves					
Source_Data_Array	, Source_Offset	, Target_Data_Array	, Target_Offset	, Length	, Function
CP_Info	, 10	, CPs_Address	, 0	, 1	, Join_Int16

NOTE: The CP_Info data array is populated by the 2nd mapdesc from the previous step and contains the first 16 bytes of the file directory table on the device.

This function will take the 2 bytes at offset 10 from the CP_Info data array and combine them into a word address and store that value in the CPs_Address data array at offset 0.

In practice, here is how the CP_Info data array will look like for a version 1.1 file directory table:

Data_Array			
Data_Array_Name	, Data_Array_Length	, Data format	, Bytes Per Item
CP_Info	, 16	, Byte	, 1

Bytes:	11	03	09	5a	02	cd	31	04	fa	01	d7	16	00	31	01	d6
Offset:	0	1	2	3	4	5	6	7	8	9	10	11	12	12	14	15

These bytes can be interpreted as follows for version 1.1 and version 2.0:

Offset (v1.1)	Offset (v2.0)	Description	Example
0	0	File directory table format version.	11 -> version 1.1 20 -> version 2.0
1	1	Number of files on the device.	03 -> 3 files
2-3	2-3	Filesize of 1st file in bytes.	095a -> 2,394 bytes
4	4-5	* File type of 1st file. 2 = template file (XIF)	02 -> 2 (v1.1) 00 02 -> 2 (v2.0)
5-6	6-7	Address of 1st file.	cd31
7-8	8-9	Filesize of 2nd file on the device.	04fa -> 1,274 bytes
9	10-11	* File type of 2nd file. 1 = configuration properties	01 -> 1 (v1.1) 00 01 -> (v2.0)
10-11	12-13	Address of 2nd file.	d716

* The first file will always be the template (XIF) file and the 2nd file will always be the configuration properties file according to the LonMark Interoperability Guidelines document, section 4-6 and 4-13.

The version of the file director format can be determined beforehand from the device's XIF file by searching for the 1st FILE section, example:

FILE template 0 2
"1.1;"
Here it shows version 1.1, so Source_Offset 10 can be used.
For version 2.0 it would show "2.0;" and Source_Offset 12 would be used.

STEP 4:
Create map descriptors for the configuration properties of the device.
At this stage in the example, the CPs_Address data array will now contain an address value of d716 which is the start address of the first configuration property on the device.

View the device's XIF file again and search for the FILE template section which will be the first FILE section encountered.

FILE template 02:

```
"1.1;"
"1,0,3\x84,62,1;"        // UCPTdeviceMajorVersion (unsigned int)
"1,0,3\xA4,63,1;"        // UCPTdeviceMinorVersion (unsigned int)
"1,0,3\x84,64,1;"        // UCPTobjectMajorVersion (unsigned int)
"1,0,3\xA4,65,1;"        // UCPTobjectMinorVersion (unsigned int)
"1,0,0\x82,17,31;"       // SCPT_location (SNVT_str_asc) Label, Location
"1,0,0\x82,22,7;"        // SCPT_max_snd_t (SNVT_elapsed_tm) Send Time, Max
"1,0,3\x82,2,7;"         // UCPTgoUncnfgT (SNVT_elapsed_tm)
"1,0,3\x82,3,7;"         // UCPTgoUncnfgPrwupT (SNVT_elapsed_tm)
"1,1,3\x85,38,3;"        // UCPTobjectType (struct ObjectType)
"1,1,3\x84,64,1;"        // UCPTobjectMajorVersion (unsigned int)
"1,1,3\xA4,65,1;"        // UCPTobjectMinorVersion (unsigned int)
"1,1,0\x81,17,31;"       // SCPT_location (SNVT_str_asc) Label, Location
"1,1,3\x81,4,7;"         // UCPTdebounceT (SNVT_elapsed_tm)
"1,1,3\x81,16,1;"        // UCPTdioType (struct DioType)
"1,1,3\x81,5,1;"         // UCPTinvertInput (boolean)
"1,1,3\x81,17,7;"        // UCPTonDelay (SNVT_elapsed_tm)
"1,1,3\x81,18,7;"        // UCPToffDelay (SNVT_elapsed_tm)
```

Each line starting from the 3rd line eg. "1,0,3\x84 … " defines a configuration property on the device. All lines that contain an x84 or x85 or xA4 are constant configuration properties which cannot be accessed and are not part of the addressing offsets into the file, so these lines should be ignored.

The following description is a quick way to determine the address offset and type of each configuration property for the most common declaration, but there may be variations so the LonMark Interoperability Guidelines document section 4-6 can be studied further for more complex definitions of configuration properties.

The format of each entry is basically:
"<header>,<select>,<flag\attributes>,<config_index>,<length>;"
eg. " 1 , 0 , 0\x82 , 17 , 31 ;"

The comments after each entry can be used to ascertain the type of CP.
For the example above the flag is 0 which always points to a SCPT type CP. Any non-zero flag value points to a UCPT type CP.

So, from this example, flag is 0, and config_index is 17, so it is SCPT 17 that should be used to determine the type of the CP.

From the Echelon SCPT list:

SCPTlocation (17)

Location Label

This configuration property sets descriptive physical location information for the associated functional block or device. It provides a more detailed description of the device that can be provided by the Neuron Chip's 6-byte location string.

A SCPTlocation configuration property that applies to the Node Object functional block is used to identify the subsystem containing the device. This allows network recovery tools to recover subsystem information from a device. The subsystem may be a simple location name, or may be a hierarchical subsystem name. If a hierarchical subsystem name is specified, the subsystem hierarchy components must be separated by periods (“.”). For example, a device may have a Node Object SCPTlocation value of “Bldg 1.Floor 2.Rm 29”, representing the Bldg 1/Floor 2/Rm 29 subsystem. Periods must not otherwise be used in a SCPTlocation value that applies to a Node Object functional block. Other characters that cannot be used in a subsystem name are the backslash (“\”), colon (“:”), forward slash (“/”), or double-quote characters. For very large networks, subsystem numbers may be used instead of subsystem names, for example: “1.2.29”. This allows deeply nested hierarchies to fit within the 31 character limit for SCPTlocation.

SCPT Index	Measurement	Type Category	Type Size
17	SNVT_str_asc	Structure	31 bytes

Structure Definition

```
typedef struct {
      unsigned char       ascii[31];
} SNVT_str_asc;
```

The ascii field contains a nul-terminated string of up to 30 characters. The default value is nul string (all zeroes).

The Measurement shows SNVT_str_asc should be used as the data-type to store the value from this configuration property. The comment in the XIF file confirms this:
// SCPT_location (SNVT_str_asc) Label, Location
Finally, the length is 31 bytes.

At this stage the following is determined:
The first accessible Configuration Property on this device is at address d716 (hex) offset 0 and it is 31 bytes long.

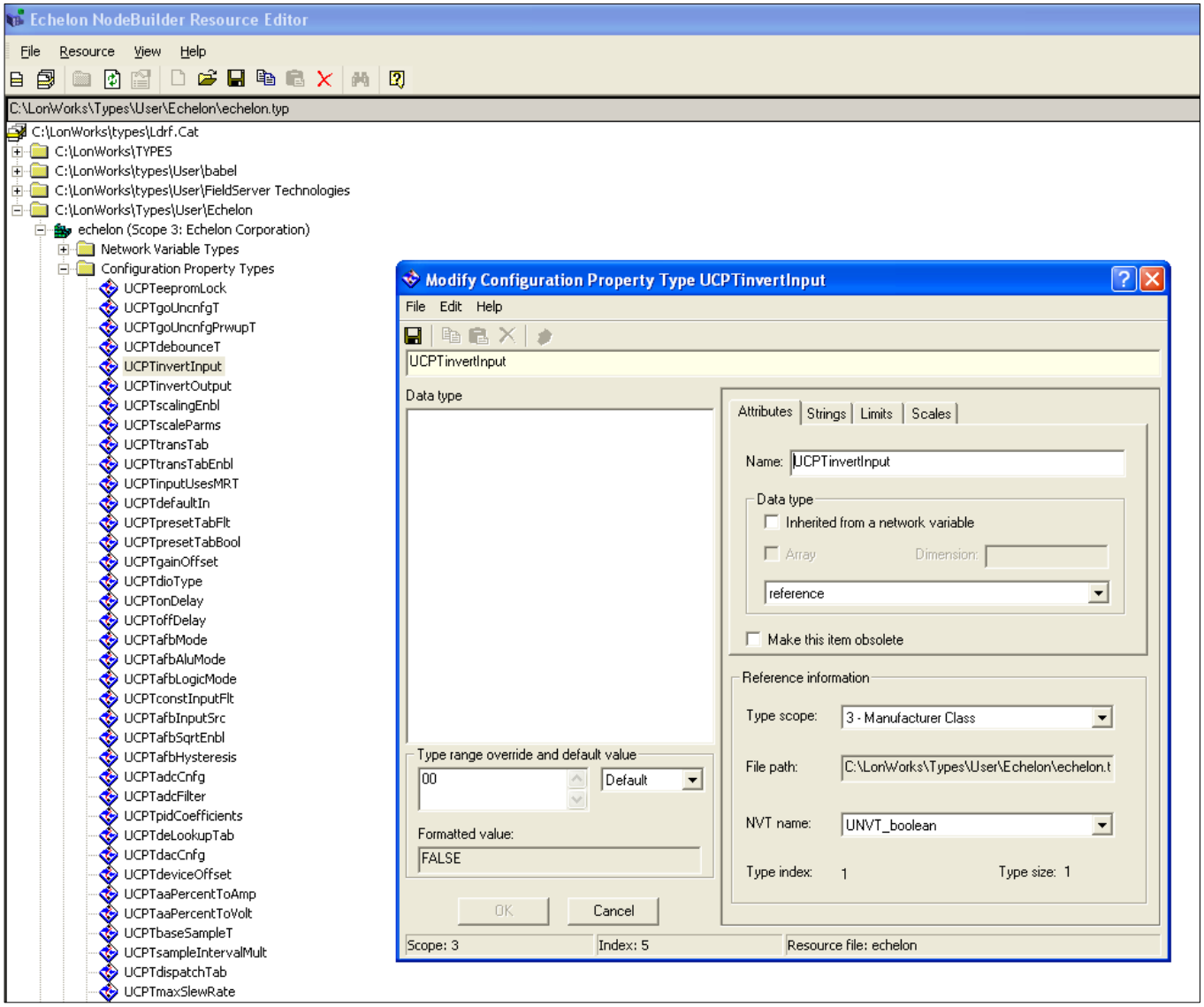
To create a mapdesc to read and write to this CP, it is defined:

Map_Descriptors					
Map_Descriptor_Name	,Data_Array_Name	,Data_Array_Offset	,DA_Address	,Address_Offset	,Lon_Function
CP_CMD001	, DA001	, 0	,CPs_Address	, 000	,NM_MEMORY

,Function	,Scan_Interval	,Node_Name	,SNVT_Type	,UNVT_ID	, UNVT_Byte_Length
, RDBC	, 0s	, Lon_Remote	, SNVT_str_asc	, -	, -

A UCPT example:
"1,1,3\x81,5,1;" // UCPTinvertInput (boolean)
the format is:
"<header>,<select>,<flag\attributes>,<config_index>,<length>;"
flag is non-zero so this is UCPT that is defined in resource file scope 3 index 5 and it is 1 byte long.

The resource editor shows the Echelon scope 3 file at index 5:



In this example, it isn't necessary to check the resource files. This is a simple example; the comment was enough to indicate the use of an UNVT type of length 1, other UCPTs may be structures with elements that need to be interpreted by the front-end to decide which offsets to read or write to inside the data bytes.

To create a mapdesc to read and write to this CP, it is defined:

Map_Descriptors					
Map_Descriptor_Name	, Data_Array_Name	, Data_Array_Offset	, DA_Address	, Address_Offset	, Lon_Function
CP_CMD008	, DA008	, 0	, CPs_Address	, 091	, NM_MEMORY

, Function	, Scan_Interval	, Node_Name	, SNVT_Type	, UNVT_ID	, UNVT_Byte_Length
, RDBC	, 0s	, Lon_Remote	, Not SNVT	, -	, 01

NOTE: The SNVT_Type is set to Not SNVT to indicate this is an UNVT and the UNVT_Id is set to 1 which is the first UNVT. The next UNVT will be 02. And finally the type size of 1 as per the example is set as the UNVT_Byte_Length.

APPENDIX B VENDOR INFORMATION

Appendix B.1 Echelon LonMaker

Appendix B.1.1 Using LonMaker to Commission the FieldServer

- Ensure that the correct firmware and latest configuration is loaded on the FieldServer (Each change in the FieldServer requires re-commissioning of the FieldServer in LonMaker).
- Ensure that the FieldServer and the LonMaker machine are on the same network.
- Open the existing Network in LonMaker, or create a new Network.
- Click on “Create New Network” and follow the network wizard, making the following selections:
  - Network Interface: Choose Network Attached
  - Management Mode: Choose Onnet unless working offline
  - Register Plug-ins required: None
- Once Visio is open with the Network showing, drag a new device onto the drawing from the toolbox.
- Follow the Device Network, making the following selections:
  - Enter Device Name: Choose commission device
  - Specify Device Template: Choose upload from device
  - Specify Device Channel: Choose Auto Detect
  - Specify Device Properties: Leave as is (Ping is optional)
  - Identify Device: Choose service pin
  - Device Application Image: Leave unchecked
  - Initial State: Leave as is
- Press the service pin on the FieldServer when asked to do so, and the FieldServer will be commissioned.
- Drag a new function block onto the drawing from the toolbox. Give the function block a name and ensure that it is allocated to the FieldServer device.
- Once the function block is on the drawing, drag input and output variables onto the function block. LonMaker will display the variables available for binding. Click on the required variables (or use the select all option), and they will be commissioned onto the function block.
- Connect these variables to other devices by dragging connections from the toolbox and connecting the variables.

Appendix B.1.2. Removing Previous FieldServer Device Templates in LonMaker

If using a previously used LonMaker to commission a FieldServer or ProtoCessor onto a LonWorks network, LonMaker saves the properties of that device as a device template. It is important to remove all previous device templates from LonMaker when moving onto the next FieldServer product as some parameters might transfer to the new device.

To remove the device templates, open the LonMaker workspace as normal. Then choose the LonMaker toolbar and choose the Device Templates option. This will bring up a list of all device templates. Highlight and then remove all device templates that do not contain “Echelon” or “LNS”. This will ensure a fresh commission of the FieldServer product.

Appendix B.2 Checking LonWorks® Network Using Echelon's NodeUtility.exe

- Run Nodeutil.exe.
- Press Service Pin on all LonWorks® Devices.
- Press 'F' to find Nodes.
- Press 'L' to List Nodes, Confirm that all Nodes can be seen.
- Press 'G' (Goto) each Node and:
- Dump Node Domain Table ('D', Enter).
    - Confirm Domain ID is correct (usually , "F").
    - Record Node.
    - Confirm that Subnet is "1".
    - Confirm that Size is "1".
- Press 'L' (List) Network Variables.
- Record SNVT Type and Index.

Appendix B.3 Neuron Systems

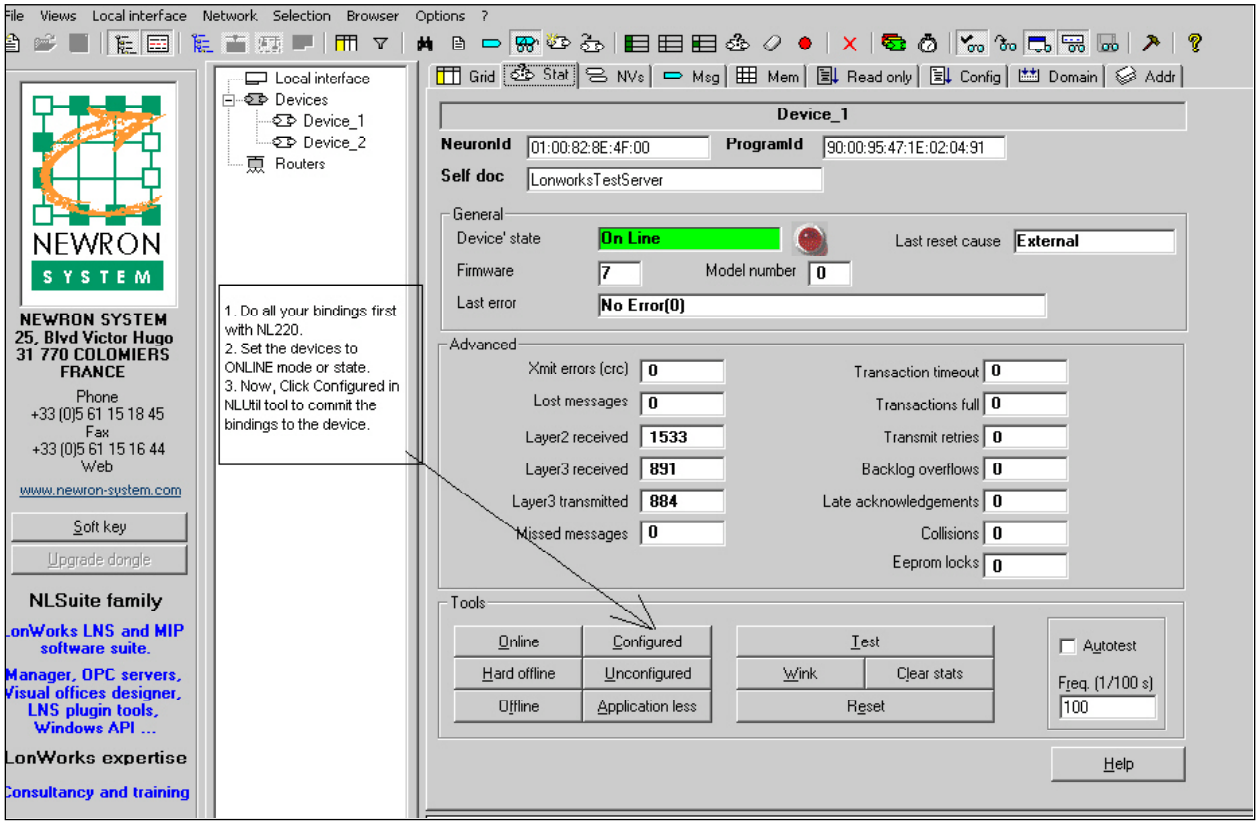
Newron Systems may produce the following Error message “bindings lost after power-cycle”. This can be fixed by placing the FieldServer in the Configured state which will cause the bindings to be committed to the Neuron and they will remain after a power cycle.

Procedure:

1) Make the bindings in NL220.

2) Place devices in ONLINE mode or state.

3) Refer to the following Screencap for assistance in placing the FieldServer in the Configured state.



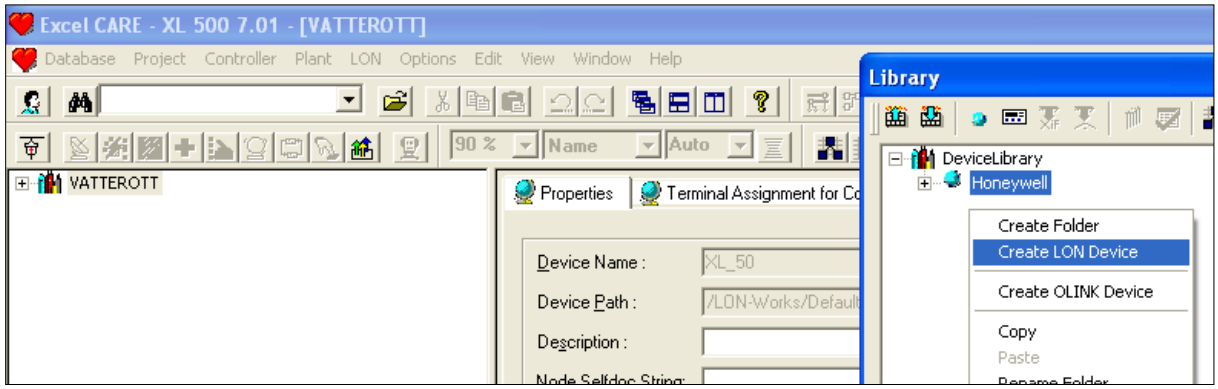
Appendix B.4 Honeywell Care

This system uses unacknowledged service as default.

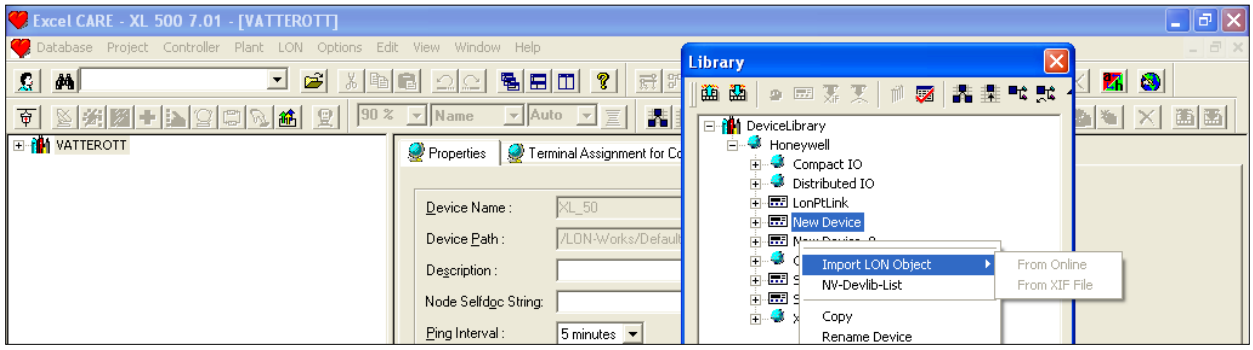
Appendix B.5 CARE and LON PT link Integration

The procedure that follows describes a successful integration from a LonPtLink using Honeywell Care ver 7.0:

- Set the PT Link configuration file as follows for LonWorks: Implicit, Update, Update.
- Open the device library in Care and create a new Lon device as shown in the example below.

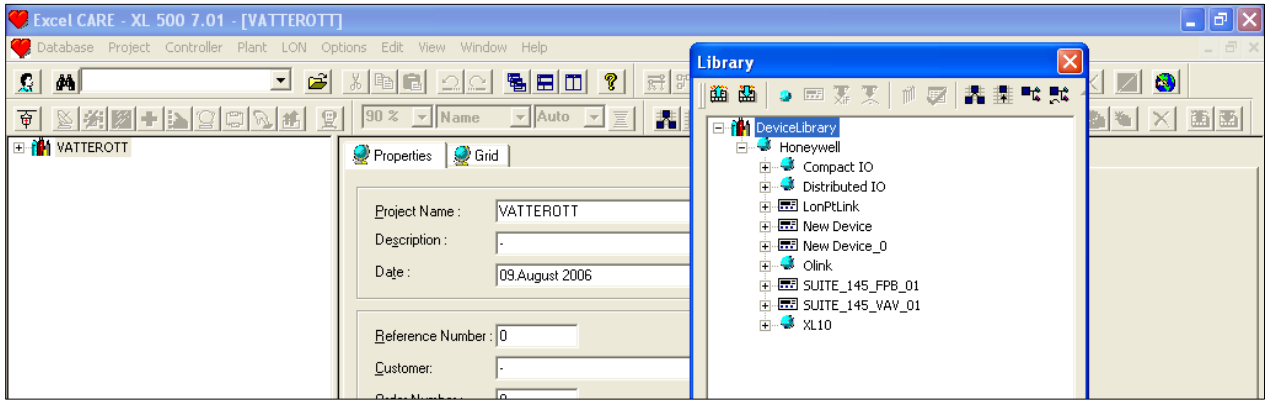


- Connect to the Lon side of the PT Link using the Lon network interface cable and set Care to Onnet.
- Right click on the new Lon device created in the device library and select Import Lon Object>From Online. This will upload the PT Link configuration into the new device.



NOTE: All Controllers that will be connected to the PT Link must be powered up and communicating to the PT Link before uploading the new device.

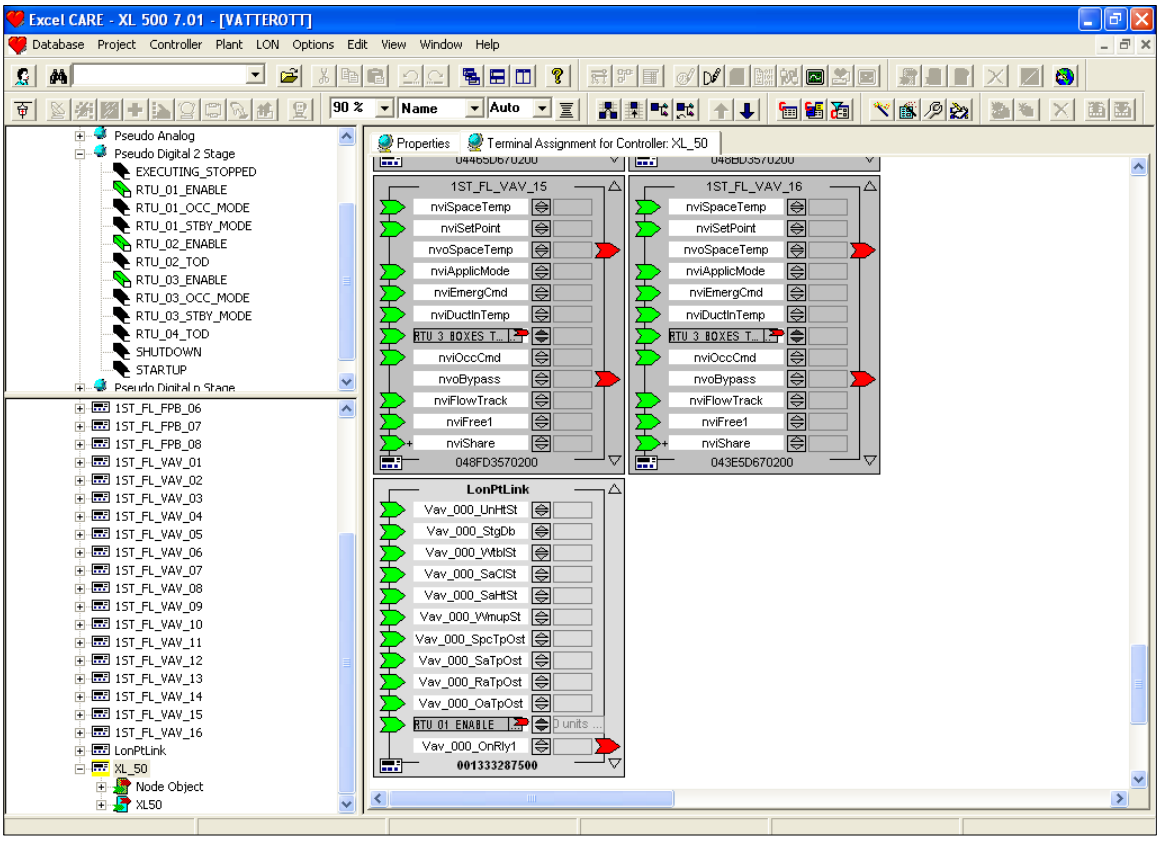
- Rename the new device e.g. LonPtLink, and drag it into the default channel.



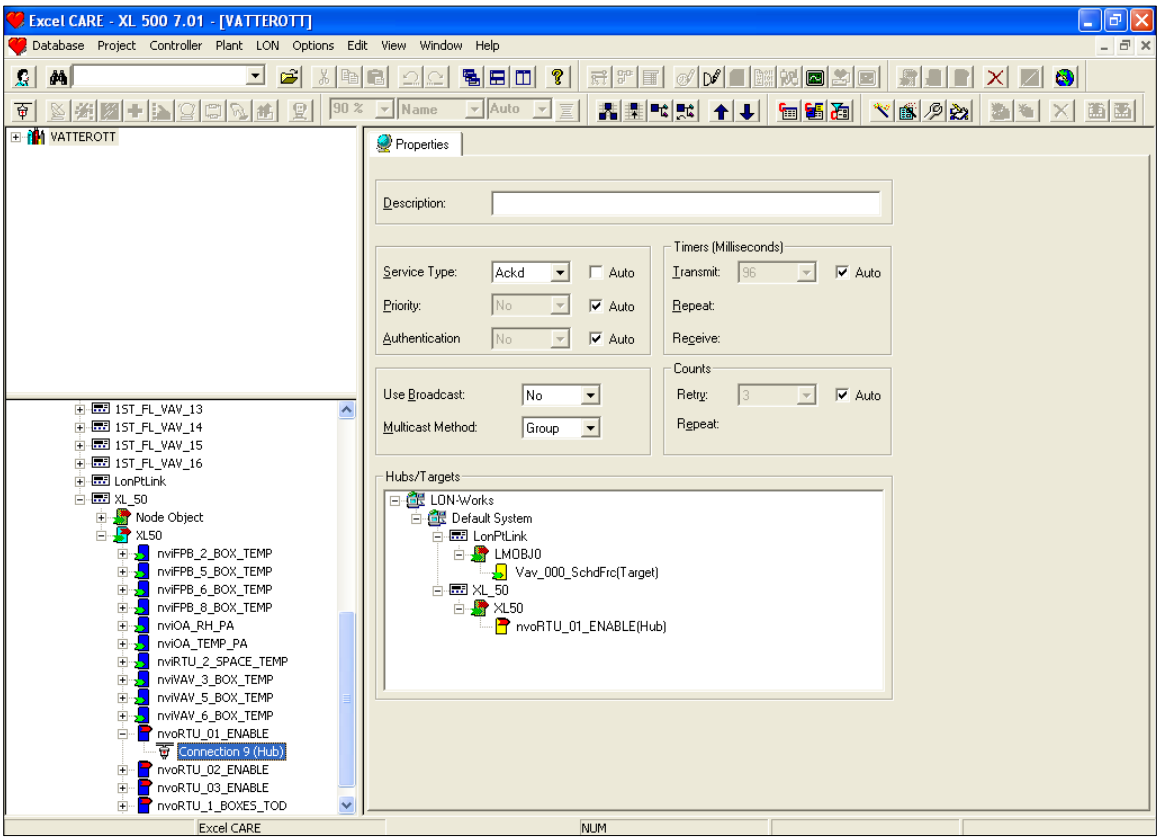
- Create the pseudo points to be bound to the PT Link. Leave them as non Lon Points for now and setup any engineering units.
- Open the Controller that is being binded from in the network tree and go to the Terminal Assignments tab.
- Drag the newly created LonPtLink device into the Terminal Assignment window.
- Now drag the pseudo points from the XL5000 controller and drop them onto the appropriate NV'S.

The example below shows the pseudo digital 2 stat RTU_01_ENABLE in the XL50 bound to the NVI schedule force (shcdfrc) on the PT Link. Using the autobind in Care will automatically configure the required snvt and lon nv.

NOTE: The schdfrc nv is the occupancy point. The pseudo point has to be set to a logical 1 for occupied and a logical 0 for unoccupied.



- Open up the nv points bound from the controller under the controller in the network tree and set the nvo points to Ackd bindings in the service type as shown below.
    - Care automatically defaults to unacknowledged bindings



- Now go onnet in Care and commission the controllers.

NOTE: If installing on a SymmetrE front end, create an XIF file for LNS to use to commission the device on synchronization. Use FS GUI or NodeUtil182 (available for free download at www.echelon.com). Ensure that the XIF is not created until everything has been configured and commissioned properly.

Appendix B.6 Circon_SI

NOTE: Circon_SI does not recognize whitespace; %; -.

NOTE: The maximum number of Network Variables under Circon_SI is limited to 255 per node.

- Set the Connection_Type parameter in the configuration file per the following example. This will ensure that *all* illegal characters are replaced with underscores, and that the number of Network Variables per Node is limited to 255 in the configuration.

Example

```
//    Client Side Connections

Connections
Adapter         , Protocol      , Connection_Type
LonWorks        , LonWorks      , Circon_SI
```

Appendix B.7 Tridium JACE

This system uses unacknowledged service as default.

NOTE: Tridium calls Acknowledged service "critical bindings".

Appendix B.7.1. Correcting SNVT Indexing Problem in Tridium Niagara

The procedure below was done in Niagara AX-Workbench Rev 3.4.X. It may not work in prior revs or with R2 because the menus are different.

If after a dynamic device is created and the nv (Network Variable) index of a monitored device does not match the XIF file of the device being monitored, the XIF can be imported by first converting the XIF to an LNML file. There is a menu item under tools in AX Workbench to do this.

- Obtain a copy of the correct XIF file.
- Choose the option for LON XML conversion under Tools.
- In the LON XML converter, browse to the XIF file.
- Choose a working directory for the new LNML file and convert the file.
- In LON Device manager, double click the device with the problem nv index.
- In the LNML file select box, browse to the newly created LNML file.
- Commission the device.
- Right click on the device, select Actions and then Upload.
- Compare the nv index with the XIF file index. If the numbers match, data should be correctly displayed after deleting and re-generating the proxy points.

Appendix B.8 Trane Rover

Use Trane Rover Service Pack 4 or later when commissioning the FieldServer. This does not ship from Trane by default and may have to be requested from the local Trane representative.

When using Pre-Service Pack 4 software, the FieldServer will appear in Rover, but none of the FieldServer variables will be available for binding even though they are present. Service Pack 4 has been tested to ensure that the variables show and the bindings work well.

Appendix B.9 Trane Tracer SC

The Trane Tracer SC is unable to deal with devices on remote subnets cleanly and all precautions needs to be taken to ensure that the FieldServer presents itself to the SC in an unconfigured manner so that the SC can allocate a Domain, SubNet and Node to the FieldServer on the same subnet.

The FieldServer LonWorks Configuration must be set up for implicit addressing by not specifying the Domain, Subnet and Node_ID in the Title section of the Configuration file, to ensure that the Trane Tracer SC tool will successfully find the FieldServer on the LonWorks Network. To this end ensure the following:

- The FieldServer needs to be configured as LonWorks Implicit on the Server Side of the configuration file.
- The DSN information needs to be completely removed from the Title section of the configuration file.

APPENDIX C TROUBLESHOOTING

Appendix C.1 Debugging a LonWorks connection: Hints and tips

- To test connections, set the configuration tool into a mode that will display service requests, and then push the button on the FieldServer. A request should appear.
- In order for variables to appear as outputs, they need to be declared as write variables in the FieldServer. Declaring the Map Descriptor functions as WRBC, or WRBX can do this. See **Section 6.1.2** for more information.
- If the domain and Subnet ID are defined in the configuration file of the FieldServer, these values will be used when power is cycled to the FieldServer. If the values set here are different to what the Network Manager has set up for the FieldServer for clients using implicit addressing, the FieldServer will stop communicating. In this case, change these settings or remove them from the configuration file completely and recommission the FieldServer.

Appendix C.2 Error Messages

Message	Description
"LON:#03 FYI. Standard driver build. This is normal.	This message is informational and can be safely ignored.

Appendix C.2.1. Error Message Lonlive.c 121

If the following error message is returned:

DRIVER-> LON : Error, timeout while waiting for Lon I/F to reset!-------- Configuration Error ------

It is possible that the FieldServer hardware in use may not support LonWorks, or the firmware loaded contains LonWorks but is the wrong firmware for the application. Remove LonWorks from the configuration or get new firmware/hardware.

Appendix C.3 Communication Errors using Neuron_ID Polling

In order to allow communication across routers when using Neuron_ID based polling, routers need to be set up to forward NID addressed messages to/from the connected channels.

Appendix C.4 Monitoring Node Status

When monitoring Node Status with a configuration using implicit addressing, it is important that at least one of the Map Descriptors has active functions (Wrbc, Rdbc). This will cause traffic and when the traffic stops, this will indicate that the Node is offline.

Appendix C.5 Exposing Operation Statistics

If this driver is appropriately configured, it can expose operation statistics in a Data Array which can be monitored by a remote device to check that the driver is performing without error.

The lines from the example below can be cut and pasted into a configuration file to expose these stats.

```
Data_Arrays,

Data_Array_Name    , Data_Format    , Data_Array_Length
Lonworks-stats     , UINT32         , 1000
```

Data Array Offset	Description
1	Increments when Neuron ID is read. Used for scripting.
2	Overwritten when Neuron ID is read. Contains 1st byte of Neuron ID.
3	Overwritten when Neuron ID is read. Contains 2nd byte of Neuron ID.
4	Overwritten when Neuron ID is read. Contains 3rd byte of Neuron ID.
5	Overwritten when Neuron ID is read. Contains 4th byte of Neuron ID.
6	Overwritten when Neuron ID is read. Contains 5th byte of Neuron ID.
7	Overwritten when Neuron ID is read. Contains 6th byte of Neuron ID.
8	Not Used.
9	Not Used.
10	Overwritten when Neuron ID is read. Contains 1st nibble of 1st byte of Neuron ID as an ASCII char. e.g. if 1st byte=0x12 then the value stored here is 0x32 - the ASCII char for the digit 2.
11	Overwritten when Neuron ID is read. Contains 2nd nibble of 1st byte of Neuron ID as an ASCII char. e.g. if 1st byte=0x12 then the value stored here is 0x31 - the ASCII char for the digit 1.
12	Overwritten when Neuron ID is read. Contains 1st nibble of 2nd byte of Neuron ID as an ASCII char.
13	Overwritten when Neuron ID is read. Contains 2nd nibble of 2nd byte of Neuron ID as an ASCII char.
14	Overwritten when Neuron ID is read. Contains 1st nibble of 3rd byte of Neuron ID as an ASCII char.
15	Overwritten when Neuron ID is read. Contains 2nd nibble of 3rd byte of Neuron ID as ASCII char.
16	Overwritten when Neuron ID is read. Contains 1st nibble of 4th byte of Neuron ID as an ASCII char.
17	Overwritten when Neuron ID is read. Contains 2nd nibble of 4th byte of Neuron ID as an ASCII char.
18	Overwritten when Neuron ID is read. Contains 1st nibble of 5th byte of Neuron ID as an ASCII char.
19	Overwritten when Neuron ID is read. Contains 2nd nibble of 5th byte of Neuron ID as an ASCII char.
20	Overwritten when Neuron ID is read. Contains 1st nibble of 6th byte of Neuron ID as an ASCII char.
21	Overwritten when Neuron ID is read. Contains 2nd nibble of 6th byte of Neuron ID as an ASCII char.
30	Domain Table Subnet ID
31	Domain Table Node ID
32	Domain Table Domain ID - length in bytes
33	Domain Table Domain ID - Byte 1
34	Domain Table Domain ID - Byte 2
35	Domain Table Domain ID - Byte 3
36	Domain Table Domain ID - Byte 4
37	Domain Table Domain ID - Byte 5
38	Domain Table Domain ID - Byte 6

Appendix C.6 “Illegal” Characters in Network Variable Names

Spaces: Spaces may not be used in Network Variable Names. The driver automatically replaces all spaces with underscores at startup.

Brackets: LonMaker truncates a name from where a bracket is used e.g. a name like nviTest[1]Input shows as nviTest. Brackets are therefore not able to be used in Network Variable Names.

Appendix C.7 Service types currently supported for network variable messages

The default service type for network variable update messages is fixed to acknowledged service. This can be changed to unacknowledged service (**Section 7.2**). Request-response service is used for network variable poll and network management fetch messages.

Appendix C.8 Combining Explicit and Implicit

To ensure undisturbed communication between the network management tool and the FieldServer during the commissioning, binding or monitoring processes, the timeouts of the explicitly addressed variables (Map Descriptors) might need to be set to values lower than the message timeouts used by the network management tool (LonMaker for Windows).

LonWorks FieldServer Performance

No.	Description	Min	Typ	Max	Units
1	FieldServer startup time with 4096 network variables	-	6.5	-	Minutes
2	Time to commission FieldServer with 4096 network variables	-	-	40	Minutes
3	Time to complete a network variable update between two FieldServers (acknowledged service)	-	40	-	Milli-seconds
4	Time to complete a network poll message between two FieldServers (request/response service)	-	125	-	Milli-seconds
5	Time to complete a network management variable fetch message between two FieldServers (request/response service)	-	125	-	Milli-seconds

APPENDIX D REFERENCE

Appendix D.1 LonMark Profiles

The following FieldServer Platforms are LonMark XIF version 3.4 certified:

FS-B35XX-05

FS-SlotServer

FS-QuickServer-Lon

ProtoCessor-Lon

The following Profiles are certified on LonMark.org.

GATEWAYS
Gateways
FS-B3510 Serial/Ethernet/LonWorks® Multiport Gateway
ProtoCarrier/ProtoCessor FPC-CD2 Daughter Card Gateway for OEMs
ProtoCessor FPC-F04 Embedded Gateway Module for OEMs
ProtoNode LER External Protocol Gateway for OEMs
QuickServer FS-QS-1011 Gateway for LonWorks®
HVAC Gateways
Liebert SiteLink-12 LonWorks® Adapter
Industrial Gateways
SlotServer LonWorks® Open Interface
HVAC
Boiler Controller
Cleaver Brooks Adapter for LonWorks®
Discharge Air Controller
FieldServer Data Aire DAP Adapter for LonWorks®
INDUSTRIAL
Generator Set
FieldServer Caterpillar EMCP II Adapter
FieldServer Kohler 550 Adapter

Appendix D.2 Performance Tests for LonMark Certified blocks

Appendix D.2.1. Startup Times, 10 Input Blocks and 10 Output Blocks Totalling 80 NVs

Device	Startup Time without a config	1st Startup Time with config	Restart Startup Time with config	Commission Time (load XIF first)	Commission Time (upload from dev.)
FFPLon/QuickServer	48 seconds	193 seconds	135 seconds		
FS-B30	57 seconds	63 seconds	62 seconds		

Appendix D.2.2. Startup Times, 20 Input Blocks and 20 Output Blocks Totalling 160 NVs

Device	Startup Time without a config	1st Startup Time with config	Restart Startup Time with config	Commission Time (load XIF first)	Commission Time (upload from dev.)
FFPLon/QuickServer	48 seconds	460 seconds	306 seconds	260 seconds	802 seconds
FS-B30	57 seconds	63 seconds	62 seconds		

Appendix D.3 SNVT_Type

The following section describes each SNVT as implemented on the FieldServer LonWorks driver.

Data Length – The number of Data Array elements required when specifying a Data Array name under a Map Descriptor.

Suggested Data Array Formats – This field suggests FieldServer Data Array formats in ascending order of storage space required while ensuring data integrity. It is not recommended to store a byte value in a bit Data Array since only the values 0 and 1 will be retained from the byte value resulting in a loss of resolution. If a byte value is stored in a Floating point Data Array, 3 bytes will go to waste. A SNVT with more than one data item may require a float Data Array format to prevent loss of resolution. Since FieldServer Map Descriptors can only work with one type of Data Array format for all the data items in a SNVT, it is sometimes necessary to use a Float Data Array element for all data items of which some may only have required a Byte. Fortunately, all SNVT's are short in length and RAM is adequately provided for on the FieldServer. If a loss in resolution can be tolerated, any Data Array format may be used.

The range of FieldServer Data Arrays formats are:

Data Array Format	Description	Values Range
BIT	1-bit	0, 1
BYTE	8-bit Byte	0.. 255
SINT16	16-bit Signed Integer	-32, 768... 32, 767
UINT16	16-bit Unsigned Integer	0.. 65, 535
SINT32	32-bit Signed Integer	-2, 147, 483, 647... 2, 147, 483, 647
UINT32	32-bit Unsigned Integer	0.. 4, 294, 967, 295
FLOAT	32-bit Float	-3.40282E38... 3.40282E38

Values Range – These are the only values that will be allowed for the SNVT, e.g. a value of 101 will not be processed in a read or write on SNVT_Switch's Value data item.

Invalid Value – The Invalid value is used to force the data item's value should a value outside the value range be encountered in a read or write. Should the invalid value be N/A (Not Applicable), the data item's value will be bounded by the indicated values range, e.g. an incoming read value of 101 will be stored as 100 for SNVT_Switch's Value data item. Note that in some cases both the type and raw values are specified. Raw values are not of the same type of the SNVT's measurement type, i.e. they are not scaled.

Data Item – SNVT's containing more than one data item have a name specified for each item. This name can be used in the SNVT_Option field of a Map Descriptor to isolate only this one item.

SNVT_abs_humid	**Data Length**			1
	Suggested Data Array Formats			Float
	Data Item 1			
	Measurement	**Units**	**Values Range**	**Invalid Value**
	Absolute humidity	gram/kilogram	0.. 655.34	Raw: 65,535 (0xFFFF) Type: 655.35
SNVT_address	**Data Length**			1
	Suggested Data Array Formats			Uint16, Float
	Data Item 1			
	Measurement	**Units**	**Values Range**	**Invalid Value**
	Neuron Chip Address	16-bit address value	16,384 .. 64,767	N/A
SNVT_alarm	**Data Length**			25
	Suggested Data Array Formats			Uint16, Float
	Data Item 1-6			Alm_Location00,..Alm_Location05
	Measurement	**Units**	**Values Range**	**Invalid Value**
	Location array element	8-bit unsigned byte	0 .. 255	N/A

	Data Item 7			Object_ID
	Object ID	object index	0 .. 65,535	N/A
	Data Item 8			Alarm_type
	alarm_type_t	N/A	0 .. 254	255 (0xFF) (AL_NUL)
	Data Item 9			Priority_level
	Priority_level_t	N/A	0 .. 11	255 (0xFF) (PR_NUL)
	Data Item 10			Index_to_SNVT
	Index to NV	index of NV causing alarm	0 .. 65,535	N/A
	Data Item 11-14			Alm_Value00, ..Alm_Value03
	Value array element	specific to NVT	0 .. 255	N/A
	Data Item 15			Year
	Year	year	0 .. 3000	N/A
	Data Item 16			Month
	Month	month of year	0 ..12	N/A
	Data Item 17			Day
	Day	day of month	0 ..31	N/A
	Data Item 18			Hour
	Hour	Hour of day	0 ..23	N/A
	Data Item 19			Minute
	Minute	Minutes	0 ..59	N/A
	Data Item 20			Second
	Second	Seconds	0 ..59	N/A
	Data Item 21			Millisecond
	Millisecond	Milliseconds	0 ..999	N/A
	Data Item 22-25			Alm_limit00, ..Alm_limit03
	Alarm limit array element	specific to NVT	0 .. 255	N/A
SNVT_alarm_2	**Data Length**			27
	Suggested Data Array Formats			Uint32
	Data Item 1			Alarm_type
	Measurement	**Units**	**Values Range**	**Invalid Value**
	alarm_type_t	N/A	0 .. 32	255 (0xFF) (AL_NUL)
	Data Item 2			Priority_level
	Priority_level_t	N/A	0 .. 11	255 (0xFF) (PR_NUL)
	Data Item 3			Alarm_time
	time	seconds	0..4,294,967,294	4294967295(0xFFFFFFFF)
	Data Item 4			milliseconds
	Millisecond	Milliseconds	0 .. 999	-1 (0xFFFF)
	Data Item 5			Sequence_number
	Number	Milliseconds	0 .. 255	N/A
	Data Item 6-27			description
	ASCII char string	N/A	0 .. 255	N/A
SNVT_amp	**Data Length**			1
	Suggested Data Array Formats			Float
	Data Item 1			
	Measurement	**Units**	**Values Range**	**Invalid Value**
	Electric current	Amperes	-3,276.8..3,276.7	N/A
SNVT_amp_ac	**Data Length**			1
	Suggested Data Array Formats			Uint16, Float
	Data Item 1			
	Measurement	**Units**	**Values Range**	**Invalid Value**
	Alt. electric current	Amperes	0 .. 65,534	65,535 (0xFFFF)
SNVT_amp_f	**Data Length**			1
	Suggested Data Array Formats			Float
	Data Item 1			
	Measurement	**Units**	**Values Range**	**Invalid Value**
	Electric current	Amperes	-3.40282E38.. 3.40282E38	N/A

SNVT_amp_mil	**Data Length**			1
	Suggested Data Array Formats			Float
	Data Item 1			
	Measurement	**Units**	**Values Range**	**Invalid Value**
	Electric current	Milli-amperes	-3,276.8.. 3,276.7	N/A
SNVT_angle	**Data Length**			1
	Suggested Data Array Formats			Uint16, Float
	Data Item 1			
	Measurement	**Units**	**Values Range**	**Invalid Value**
	Phase/Rotation	Radians	0 .. 65.535	N/A
SNVT_angle_deg	**Data Length**			1
	Suggested Data Array Formats			Float
	Data Item 1			
	Measurement	**Units**	**Values Range**	**Invalid Value**
	Angular distance	degrees	-359.98 .. 360.00	Type: 655.34 Raw: 32,767 (0x7FFF)
SNVT_angle_f	**Data Length**			1
	Suggested Data Array Formats			Float
	Data Item 1			
	Measurement	**Units**	**Values Range**	**Invalid Value**
	Phase/Rotation	Radians	-3.40282E38.. 3.40282E38	N/A
SNVT_angle_vel	**Data Length**			1
	Suggested Data Array Formats			Float
	Data Item 1			
	Measurement	**Units**	**Values Range**	**Invalid Value**
	Angular velocity	Radians/second	-3,276.8..3,276.7	N/A
SNVT_angle_vel_f	**Data Length**			1
	Suggested Data Array Formats			Float
	Data Item 1			
	Measurement	**Units**	**Values Range**	**Invalid Value**
	Angular Velocity	Radians/Second	-3.40282E38.. 3.40282E38	N/A
SNVT_area	**Data Length**			1
	Suggested Data Array Formats			Float
	Data Item 1			
	Measurement	**Units**	**Values Range**	**Invalid Value**
	Area	Square meters (m^2)	0 .. 13.1068	Type: 13.107 Raw: 0xFFFF (65,535)
SNVT_btu_f	**Data Length**			1
	Suggested Data Array Formats			Float
	Data Item 1			
	Measurement	**Units**	**Values Range**	**Invalid Value**
	Thermal Energy	British Thermal Units	0 .. 3.40282E38	N/A
SNVT_btu_kilo	**Data Length**			1
	Suggested Data Array Formats			Uint16, Float
	Data Item 1			
	Measurement	**Units**	**Values Range**	**Invalid Value**
	Thermal Energy	Kilo-British Thermal Units	0 .. 65,535	N/A
SNVT_btu_mega	**Data Length**			1
	Suggested Data Array Formats			Uint16, Float
	Data Item 1			
	Measurement	**Units**	**Values Range**	**Invalid Value**
	Thermal Energy	Mega-British Thermal Units	0 .. 65,535	N/A
SNVT_char_ascii	**Data Length**			1
	Suggested Data Array Formats			Byte, Uint16, Float
	Data Item 1			
	Measurement	**Units**	**Values Range**	**Invalid Value**
	Character	8-bit ASCII Char	0 .. 255	N/A

SNVT_chlr_status	**Data Length**			3
	Suggested Data Array Formats			Byte, Uint16, Float
	Data Item 1			chlr_run_mode
	Measurement	**Units**	**Values Range**	**Invalid Value**
	Byte, Uint16, Float	8-bit ASCII Char	0 .. 255	N/A
	Data Item 2			chlr_op_mode
	hvac_t	N/A	0 .. 17	255 (0xFF) (HVAC_NUL)
	Data Item 3			chlr_state
		N/A	0 .. 255	N/A
SNVT_color	**Data Length**			3
	Suggested Data Array Formats			Float
	Data Item 1			L_star
	Measurement	**Units**	**Values Range**	**Invalid Value**
	Lightness	Lightness	0.0 .. 100.0	N/A
	Data Item 2			a_star
	Redness/Greenness	Redness/Greenness	-200.0 .. 200.0	N/A
	Data Item 3			b_star
	Yellowness/Blueness	Yellowness/Blueness	-200.0 .. 200.0	N/A
SNVT_config_src	**Data Length**			1
	Suggested Data Array Formats			Byte, Uint16, Float
	Data Item 1			
	Measurement	**Units**	**Values Range**	**Invalid Value**
	Config_source_t	N/A	0 .. 1	255 (0xFF) (CFG_NUL)
SNVT_count	**Data Length**			1
	Suggested Data Array Formats			Uint16, Float
	Data Item 1			
	Measurement	**Units**	**Values Range**	**Invalid Value**
	Event Count	Count	0 .. 65,535	N/A
SNVT_count_f	**Data Length**			1
	Suggested Data Array Formats			Float
	Data Item 1			
	Measurement	**Units**	**Values Range**	**Invalid Value**
	Event Count	Count	0 .. 3.40282E38	N/A
SNVT_count_inc	**Data Length**			1
	Suggested Data Array Formats			Sint16, Float
	Data Item 1			
	Measurement	**Units**	**Values Range**	**Invalid Value**
	Incremental Count	Count	-32,768 .. 32,767	N/A
SNVT_count_inc_f	**Data Length**			1
	Suggested Data Array Formats			Float
	Data Item 1			
	Measurement	**Units**	**Values Range**	**Invalid Value**
	Incremental Count	Count	-3.40282E38.. 3.40282E38	N/A
SNVT_ctrl_req	**Data Length**			3
	Suggested Data Array Formats			Uint16, Float
	Data Item 1			ctl_req_rcvr_id
	Measurement	**Units**	**Values Range**	**Invalid Value**
	Receiver ID	ID number	1 .. 65,535	0
	Data Item 2			ctl_req_sndr_id
	Sender ID	ID number	1 .. 65,534	65,535 (0xFFFF)
	Data Item 3			ctl_req_sndr_pri
	Sender priority	Priority value	0 .. 200	N/A

SNVT_ctrl_resp	**Data Length**			5
	Suggested Data Array Formats			Uint16, Float
	Data Item 1			ctl_rsp_status
	Measurement	**Units**	**Values Range**	**Invalid Value**
	control_resp_t	N/A	0 .. 5	255 (0xFF) (CTRLR_NUL)
	Data Item 2			ctl_rsp_snd_id
	Sender ID	ID number	1 .. 65,534	65,535 (0xFFFF)
	Data Item 3			ctl_rsp_snd_rlower
	Sender Range Lower ID	ID number	1 .. 65,534	65,535 (0xFFFF)
	Data Item 4			ctl_rsp_snd_rupper
	ctl_rsp_snd_rupper	ID number	1 .. 65,534	65,535 (0xFFFF)
	Data Item 5			ctl_rsp_cntrl_id
	Controller ID	ID number	1 .. 65,534	65,535 (0xFFFF)
SNVT_currency	**Data Length**			3
	Suggested Data Array Formats			Sint32
	Data Item 1			Currency
	Measurement	**Units**	**Values Range**	**Invalid Value**
	Currency_t	N/A	0 .. 56	255 (0xFF) (CU_NUL)
	Data Item 2			Power_of_10
	Magnitude	Power of 10	-128 .. 127	N/A
	Data Item 3			Currency_Value
	Value	Currency Value	-2,147,483,648 .. 2,147,483,647	N/A
SNVT_date_day	**Data Length**			1
	Suggested Data Array Formats			Byte, Uint16, Float
	Data Item 1			
	Measurement	**Units**	**Values Range**	**Invalid Value**
	days_of_week_t	N/A	0 .. 6	255 (0xFF)
SNVT_defr_mode	**Data Length**			1
	Suggested Data Array Formats			Byte, Uint16, Float
	Data Item 1			
	Measurement	**Units**	**Values Range**	**Invalid Value**
	defrost_t	N/A	0 .. 2	255 (0xFF) (DFM_MODE_NUL)
SNVT_defr_state	**Data Length**			1
	Suggested Data Array Formats			Byte, Uint16, Float
	Data Item 1			
	Measurement	**Units**	**Values Range**	**Invalid Value**
	defrost_state_t	N/A	0 .. 4	255 (0xFF) (DFS_NUL)
SNVT_defr_term	**Data Length**			1
	Suggested Data Array Formats			Byte, Uint16, Float
	Data Item 1			
	Measurement	**Units**	**Values Range**	**Invalid Value**
	defrost_term_t	N/A	0 .. 100	255 (0xFF) (DFT_NUL)
SNVT_density	**Data Length**			1
	Suggested Data Array Formats			Float
	Data Item 1			
	Measurement	**Units**	**Values Range**	**Invalid Value**
	Density	Kg/m³: kilograms per cubic meter	0 .. 32,767.5	N/A
SNVT_density_f	**Data Length**			1
	Suggested Data Array Formats			Float
	Data Item 1			
	Measurement	**Units**	**Values Range**	**Invalid Value**
	Density	Kilograms per Cubic Meter (kg/m³)	0 .. 3.40282E38	N/A
SNVT_dev_c_mode	**Data Length**			1
	Suggested Data Array Formats			Byte, Uint16, Float
	Data Item 1			
	Measurement	**Units**	**Values Range**	**Invalid Value**
	device_c_mode_t	N/A	0 .. 29	255 (0xFF) (DCM_NUL)

SNVT_earth_pos	**Data Length**			6
	Suggested Data Array Formats			Float
	Data Item 1			epos_dir_bits
	Measurement	**Units**	**Values Range**	**Invalid Value**
	Latitude/Longitude	Direction	0 .. 255	N/A
	Data Item 2			epos_lat_degs
	Latitude degrees	Degrees	1 .. 90	255 (0xFF)
	Data Item 3			epos_lat_mins
	Latitude Minutes	Minutes	1 .. 59.999	Raw: 65,535 (0xFFFF) Type: 65.535
	Data Item 4			epos_long_degs
	Longitude Degrees	Degrees	1 ..180	255 (0xFF)
	Data Item 5			epos_long_mins
	Longitude Minutes	Minutes	1 .. 59.999	Raw: 65,535 (0xFFFF) Type: 65.535
	Data Item 6			epos_htabv_sea
	Height Above Sea Level	Meters (m)	-3.40282E38.. 3.40282E38	N/A
SNVT_elapsed_tm	**Data Length**			5
	Suggested Data Array Formats			Uint16, Float
	Data Item 1			etm_day
	Measurement	**Units**	**Values Range**	**Invalid Value**
	Days	Days	0 .. 65,534	65,535 (0xFFFF)
	Data Item 2			etm_hour
	Hours	Hours	0 .. 23	N/A
	Data Item 3			etm_minute
	Minutes	Minutes	0 .. 59	N/A
	Data Item 4			etm_second
	Seconds	Seconds	0 .. 59	N/A
	Data Item 5			etm_millisecond
	Milliseconds	Milliseconds	0 .. 999	N/A
SNVT_elec_kwh	**Data Length**			1
	Suggested Data Array Formats			Uint16, Float
	Data Item 1			
	Measurement	**Units**	**Values Range**	**Invalid Value**
	Electrical energy	Kilowatt-hours	0 .. 65,535	N/A
SNVT_elec_kwh_l	**Data Length**			1
	Suggested Data Array Formats			Sint32
	Data Item 1			
	Measurement	**Units**	**Values Range**	**Invalid Value**
	Electricity	Kilowatt-hour	-214,748,364.8 ..214,748,364.6	-214,748,364.7 (0x7FFFFFFF)
SNVT_elec_whr	**Data Length**			1
	Suggested Data Array Formats			Float
	Data Item 1			
	Measurement	**Units**	**Values Range**	**Invalid Value**
	Electric energy	Watt-hours	0 .. 6,553.5	N/A
SNVT_elec_whr_f	**Data Length**			1
	Suggested Data Array Formats			Float
	Data Item 1			
	Measurement	**Units**	**Values Range**	**Invalid Value**
	Electric Energy	Watt-hour	0 .. 3.40282E38	N/A
SNVT_enthalpy	**Data Length**			1
	Suggested Data Array Formats			Float
	Data Item 1			
	Measurement	**Units**	**Values Range**	**Invalid Value**
	Enthalpy	KiloJoules per Kilogram kJ/kg	-327.68 .. 327.66	Raw: 32,767 (0x7FFF) Type: 327.67

SNVT_evap_state	**Data Length**			1
	Suggested Data Array Formats			Byte, Uint16, Float
	Data Item 1			
	Measurement	**Units**	**Values Range**	**Invalid Value**
	evap_t	N/A	0 .. 2	255 (0xFF) (EVAP_NUL)
SNVT_ex_control	**Data Length**			10
	Suggested Data Array Formats			Byte, Uint16, Float
	Data Item 1			ex_ctrl_status
	Measurement	**Units**	**Values Range**	**Invalid Value**
	ex_control_t	N/A	0 .. 2	255 (0xFF) (EX_CONTROL_NUL)
	Data Item 2-7			ex_ctrl_domid_0, ..ex_ctrl_domid_5
	Domain ID	Byte	0 .. 255	N/A
	Data Item 8			ex_ctrl_domid_len
	Domain length	Number of bytes	0 .. 6	N/A
	Data Item 9			ex_ctrl_subnet
	Subnet	Subnet number	1 .. 255	N/A
	Data Item 10			ex_ctrl_node
	Node	Node number	1 .. 127	N/A
SNVT_file_pos	**Data Length**			2
	Suggested Data Array Formats			Uint32
	Data Item 1			ops_rw_ptr
	Measurement	**Units**	**Values Range**	**Invalid Value**
	Read/Write Pointer	File Byte Address	0..2,147,483,647	N/A
	Data Item 2			ops_rw_len
	Read/Write Length	Number of Bytes	0 .. 65,535	N/A
SNVT_file_req (only raw data support)	**Data Length**			12
	Suggested Data Array Formats			Byte
SNVT_file_status (only raw data support)	**Data Length**			27
	Suggested Data Array Formats			Byte
SNVT_fire_indcte	**Data Length**			1
	Suggested Data Array Formats			Byte, Uint16, Float
	Data Item 1			
	Measurement	**Units**	**Values Range**	**Invalid Value**
	fire_indicator_t	N/A	0 .. 8	255 (0xFF) (FN_NUL)
SNVT_fire_init	**Data Length**			1
	Suggested Data Array Formats			Byte, Uint16, Float
	Data Item 1			
	Measurement	**Units**	**Values Range**	**Invalid Value**
	fire_initiator_t	N/A	0 .. 16	255 (0xFF) (FI_NUL)
SNVT_fire_test	**Data Length**			1
	Suggested Data Array Formats			Byte, Uint16, Float
	Data Item 1			
	Measurement	**Units**	**Values Range**	**Invalid Value**
	fire_test_t	N/A	0 .. 3	255 (0xFF) (FT_NUL)
SNVT_flow	**Data Length**			1
	Suggested Data Array Formats			Uint16, Float
	Data Item 1			
	Measurement	**Units**	**Values Range**	**Invalid Value**
	Flow volume	Liters/Second	0 .. 65,534	65,535 (0xFFFF)
SNVT_flow_f	**Data Length**			1
	Suggested Data Array Formats			Float
	Data Item 1			
	Measurement	**Units**	**Values Range**	**Invalid Value**
	Flow Volume	Liters/Second	-3.40282E38.. 3.40282E38	N/A
SNVT_flow_mil	**Data Length**			1
	Suggested Data Array Formats			Uint16, Float
	Data Item 1			
	Measurement	**Units**	**Values Range**	**Invalid Value**
	Flow volume	Milliliters/Second ml/s	0 .. 65,535	N/A

SNVT_flow_p	**Data Length**			1
	Suggested Data Array Formats			Float
	Data Item 1			
	Measurement	**Units**	**Values Range**	**Invalid Value**
	Flow volume	Cubic meters per hr	0 .. 655.34	Raw: 65,535 (0xFFFF) Type: 655.35
SNVT_freq_f	**Data Length**			1
	Suggested Data Array Formats			Float
	Data Item 1			
	Measurement	**Units**	**Values Range**	**Invalid Value**
	Frequency	Hertz	0 .. 3.40282E38	N/A
SNVT_freq_hz	**Data Length**			1
	Suggested Data Array Formats			Float
	Data Item 1			
	Measurement	**Units**	**Values Range**	**Invalid Value**
	Frequency	Hertz	0 .. 6,553.5	N/A
SNVT_freq_kilohz	**Data Length**			1
	Suggested Data Array Formats			Float
	Data Item 1			
	Measurement	**Units**	**Values Range**	**Invalid Value**
	Frequency	Kilohertz	0 .. 6,553.5	N/A
SNVT_freq_milhz	**Data Length**			1
	Suggested Data Array Formats			Float
	Data Item 1			
	Measurement	**Units**	**Values Range**	**Invalid Value**
	Frequency	Hertz	0 .. 6.5535	N/A
SNVT_gfci_status	**Data Length**			1
	Suggested Data Array Formats			Byte, Uint16, Float
	Data Item 1			
	Measurement	**Units**	**Values Range**	**Invalid Value**
	gfci_status_t	N/A	0 .. 5	255 (0xFF) (GFCI_NUL)
SNVT_grammage	**Data Length**			1
	Suggested Data Array Formats			Float
	Data Item 1			
	Measurement	**Units**	**Values Range**	**Invalid Value**
	Grammage	Grams/Square-meter	0 .. 6,553.5	N/A
SNVT_grammage_f	**Data Length**			1
	Suggested Data Array Formats			Float
	Data Item 1			
	Measurement	**Units**	**Values Range**	**Invalid Value**
	Grammage	Grams/Square-meter	0 .. 3.40282E38	N/A
SNVT_hvac_emerg	**Data Length**			1
	Suggested Data Array Formats			Byte, Uint16, Float
	Data Item 1			
	Measurement	**Units**	**Values Range**	**Invalid Value**
	emerg_t	N/A	0 .. 5	255 (0xFF) (EMERG_NUL)
SNVT_hvac_mode	**Data Length**			1
	Suggested Data Array Formats			Byte, Uint16, Float
	Data Item 1			
	Measurement	**Units**	**Values Range**	**Invalid Value**
	Hvac_t	N/A	0 .. 17	255 (0xFF) (HVAC_NUL)
SNVT_hvac_overid	**Data Length**			3
	Suggested Data Array Formats			Float
	Data Item 1			hvac_ord_state
	Measurement	**Units**	**Values Range**	**Invalid Value**
	hvac_overid_t	N/A	0 .. 48	255 (0xFF) (HVO_NUL)
	Data Item 2			hvac_ord_percent
	Percent	Percent of Full Scale	-163.840.. 163.830	Type: 163.835 Raw: 32,767 (0x7FFF)
	Data Item 3			hvac_ord_flow
	Flow	Liters per Second	0 .. 65,534	65,535 (0xFFF)

SNVT_hvac_status	**Data Length**			7
	Suggested Data Array Formats			Float
	Data Item 1			hvac_sts_mode
	Measurement	**Units**	**Values Range**	**Invalid Value**
	hvac_t	N/A	0 .. 17	255 (0xFF)(HV_NUL)
	Data Item 2			hvac_heat_out_pri
	Primary Heat Output	Percent of Full Scale	-163.840 .. 163.830	Type: 163.835 Raw: 32,767 (0x7FFF)
	Data Item 3			hvac_heat_out_sec
	Secondary Heat Output	Percent of Full Scale	-163.840 .. 163.830	Type: 163.835 Raw: 32,767 (0x7FFF)
	Data Item 4			hvac_cool_out
	Cooling Output	Percent of Full Scale	-163.840 .. 163.830	Type: 163.835 Raw: 32,767 (0x7FFF)
	Data Item 5			hvac_econ_out
	Economizer Output	Percent of Full Scale	-163.840 .. 163.830	Type: 163.835 Raw: 32,767 (0x7FFF)
	Data Item 6			hvac_fan_out
	Fan Output	Percent of Full Scale	-163.840 .. 163.830	Type: 163.835 Raw: 32,767 (0x7FFF)
	Data Item 7			hvac_in_alarm
	In Alarm State	Alarm value	0 .. 1	N/A
SNVT_hvac_type	**Data Length**			1
	Suggested Data Array Formats			Byte, Uint16, Float
	Data Item 1			
	Measurement	**Units**	**Values Range**	**Invalid Value**
	Hvac_hvt_t	N/A	0 .. 9	255 (0xFF)(HVT_NUL)
SNVT_length	**Data Length**			1
	Suggested Data Array Formats			Float
	Data Item 1			
	Measurement	**Units**	**Values Range**	**Invalid Value**
	Length	Meters (m)	0 .. 6,553.5	N/A
SNVT_length_f	**Data Length**			1
	Suggested Data Array Formats			Float
	Data Item 1			
	Measurement	**Units**	**Values Range**	**Invalid Value**
	Length	Meters	0 .. 3.40282E38	N/A
SNVT_length_kilo	**Data Length**			1
	Suggested Data Array Formats			Float
	Data Item 1			
	Measurement	**Units**	**Values Range**	**Invalid Value**
	Length	Kilometers (km)	0 .. 6,553.5	N/A
SNVT_length_micr	**Data Length**			1
	Suggested Data Array Formats			Float
	Data Item 1			
	Measurement	**Units**	**Values Range**	**Invalid Value**
	Length	Micrometers, Microns	0 .. 6,553.5	N/A
SNVT_length_mil	**Data Length**			1
	Suggested Data Array Formats			Float
	Data Item 1			
	Measurement	**Units**	**Values Range**	**Invalid Value**
	Length	Millimeters (mm)	0 .. 6,553.5	N/A
SNVT_lev_cont	**Data Length**			1
	Suggested Data Array Formats			Float
	Data Item 1			
	Measurement	**Units**	**Values Range**	**Invalid Value**
	Continuous Level	Percent of Full Level	0 .. 100.0	N/A

SNVT_lev_cont_f	**Data Length**			1
	Suggested Data Array Formats			Float
	Data Item 1			
	Measurement	**Units**	**Values Range**	**Invalid Value**
	Continuous Level	Percent of Full Scale	0 .. 100	N/A
SNVT_lev_disc	**Data Length**			1
	Suggested Data Array Formats			Byte, Uint16, Float
	Data Item 1			
	Measurement	**Units**	**Values Range**	**Invalid Value**
	Discrete_levels_t	N/A	0 .. 4	255 (0xFF)(ST_NUL)
SNVT_lev_percent	**Data Length**			1
	Suggested Data Array Formats			Float
	Data Item 1			
	Measurement	**Units**	**Values Range**	**Invalid Value**
	Percentage Level	% of Full Scale, or Parts-per Million (ppm)	-163.840.. 163.830	Type: 163.835 Raw: 32,767 (0x7FFF)
SNVT_lux	**Data Length**			1
	Suggested Data Array Formats			Uint16, Float
	Data Item 1			
	Measurement	**Units**	**Values Range**	**Invalid Value**
	Illumination	Lux	0 .. 65,535	N/A
SNVT_magcard (only raw data support)	**Data Length**			20
	Suggested Data Array Formats			Byte
SNVT_mass	**Data Length**			1
	Suggested Data Array Formats			Float
	Data Item 1			
	Measurement	**Units**	**Values Range**	**Invalid Value**
	Mass	Grams	0 .. 6,553.5	N/A
SNVT_mass_f	**Data Length**			1
	Suggested Data Array Formats			Float
	Data Item 1			
	Measurement	**Units**	**Values Range**	**Invalid Value**
	Mass	Grams	0 .. 3.40282E38	N/A
SNVT_mass_flow	**Data Length**			1
	Suggested Data Array Formats			Float, Uint16
	Data Item 1			
	Measurement	**Units**	**Values Range**	**Invalid Value**
	Mass	kg/h	-32768..32767	N/A
SNVT_mass_flow_f	**Data Length**			1
	Suggested Data Array Formats			Float
	Data Item 1			
	Measurement	**Units**	**Values Range**	**Invalid Value**
	Mass	kg/h	-3.40282E38.. 3.40282E38	N/A
SNVT_mass_kilo	**Data Length**			1
	Suggested Data Array Formats			Float
	Data Item 1			
	Measurement	**Units**	**Values Range**	**Invalid Value**
	Mass	Kilograms (kg)	0 .. 6,553.5	N/A
SNVT_mass_mega	**Data Length**			1
	Suggested Data Array Formats			Float
	Data Item 1			
	Measurement	**Units**	**Values Range**	**Invalid Value**
	Mass	Metric Tons; Tonne	0 .. 6,553.5	N/A
SNVT_mass_mil	**Data Length**			1
	Suggested Data Array Formats			Float
	Data Item 1			
	Measurement	**Units**	**Values Range**	**Invalid Value**
	Mass	Milligrams (mg)	0 .. 6,553.5	N/A

SNVT_motor_state	**Data Length**			1
	Suggested Data Array Formats			Byte, Uint16, Float
	Data Item 1			
	Measurement	**Units**	**Values Range**	**Invalid Value**
	motor_state_t	N/A	0 .. 7	255 (0xFF) (MOTOR_NUL)
SNVT_mul_div	**Data Length**			2
	Suggested Data Array Formats			Uint16, Float
	Data Item 1			Multiplier
	Measurement	**Units**	**Values Range**	**Invalid Value**
	Multiplier	N/A	0 .. 65,535	N/A
	Data Item 2			Divisor
	Divisor	N/A	1 .. 65,535	0
SNVT_multiplier	**Data Length**			1
	Suggested Data Array Formats			Float
	Data Item 1			
	Measurement	**Units**	**Values Range**	**Invalid Value**
	Multiplier	N/A	0 .. 32.7675	N/A
SNVT_nv_type	**Data Length**			15
	Suggested Data Array Formats			Byte, Uint16, Float
	Data Item 1-8			nv_type_pid0,..nv_type_pid7
	Measurement	**Units**	**Values Range**	**Invalid Value**
	Program ID	N/A	0 .. 255	N/A
	Data Item 9			nv_type_scope
	Scope	N/A	0 .. 6	N/A
	Data Item 10			nv_type_index
	Type index	N/A	1 .. 65,535	0
	Data Item 11			nv_type_cat
	nv_type_category_t	N/A	1 .. 14	255 (0xFF)(NVT_CAT_NUL)
	Data Item 12			nv_type_length
	Type length	Bytes	1 .. 31	0
	Data Item 13			nv_type_scale_a
	Scaling multiplier	N/A	-32,768 .. 32,766	32,767
SNVT_nv_type	**Data Item 14**			nv_type_scale_b
	Scaling exponent	N/A	-32,768 .. 32,766	32,767
	Data Item 15			nv_type_scale_c
	Scaling offset	N/A	-32,768 .. 32,766	32,767
SNVT_obj_request	**Data Length**			2
	Suggested Data Array Formats			Uint16, Float
	Data Item 1			obj_request_id
	Measurement	**Units**	**Values Range**	**Invalid Value**
	obj_request_id	File Byte Address	0..2,147,483,647	N/A
	Data Item 2			obj_request
	object_request_t	N/A	0 .. 17	255 (0xFF)(RQ_NUL)
SNVT_obj_status	**Data Length**			2
	Suggested Data Array Formats			Uint32
	Data Item 1			obj_status_id
	Measurement	**Units**	**Values Range**	**Invalid Value**
	Functional Block ID	N/A	0 .. 65,535	N/A
	Data Item 2			obj_status_bits
	object_status_bits	Bits	0..4,294,967,295	N/A
SNVT_occupancy	**Data Length**			1
	Suggested Data Array Formats			Byte, Uint16, Float
	Data Item 1			
	Measurement	**Units**	**Values Range**	**Invalid Value**
	Occup_t	N/A	0 .. 3	255 (0xFF) (OC_NUL)
SNVT_override	**Data Length**			1
	Suggested Data Array Formats			Byte, Uint16, Float
	Data Item 1			
	Measurement	**Units**	**Values Range**	**Invalid Value**
	override_t	N/A	0 .. 2	255 (0xFF) (OV_NUL)

SNVT_pH	**Data Length**			1
	Suggested Data Array Formats			Float
	Data Item 1			
	Measurement	**Units**	**Values Range**	**Invalid Value**
	Acidity	pH	-32.768 .. 32.767	N/A
SNVT_pH_f	**Data Length**			1
	Suggested Data Array Formats			Float
	Data Item 1			
	Measurement	**Units**	**Values Range**	**Invalid Value**
	Acidity	pH	-3.40282E38.. 3.40282E38	N/A
SNVT_pos_ctrl	**Data Length**			9
	Suggested Data Array Formats			Float
	Data Item 1			pos_ctrl_rcvr_id
	Measurement	**Units**	**Values Range**	**Invalid Value**
	Receiver ID	ID number	1 .. 65,535	0
	Data Item 2			pos_ctrl_cntrl_id
	Controller ID	ID number	1 .. 65,534	65,535 (0xFFFF)
	Data Item 3			pos_ctrl_cntrl_prio
	Controller priority	Priority value	0 .. 200	N/A
	Data Item 4			pos_ctrl_function
	cam_func_t	N/A	0 .. 2	255 (0xFF)(CMF_NUL)
	Data Item 5			pos_ctrl_action
	cam_act_t	N/A	0 .. 2	255 (0xFF)(CMA_NUL)
	Data Item 6			pos_ctrl_number
	Action number	Action number	1 .. 255	0
	Data Item 7			pos_ctrl_pan
	Pan position	Degrees	-359.98 .. 360.00	Raw: 32,767 (0x7FFF) Type: 655.34
	Data Item 8			pos_ctrl_tilt
	Tilt position	Degrees	-359.98 .. 360.00	Raw: 32,767 (0x7FFF) Type: 655.34
	Data Item 9			pos_ctrl_zoom
	Zoom position	Percent of full-scale or ppm	-163.840.. 163.830	Raw: 32,767 (0x7FFF) Type: 163.835
SNVT_power	**Data Length**			1
	Suggested Data Array Formats			Float
	Data Item 1			
	Measurement	**Units**	**Values Range**	**Invalid Value**
	Power	Watts (W)	0 .. 6,553.5	N/A
SNVT_power_f	**Data Length**			1
	Suggested Data Array Formats			Float
	Data Item 1			
	Measurement	**Units**	**Values Range**	**Invalid Value**
	Power	Watts	-3.40282E38.. 3.40282E38	N/A
SNVT_power_kilo	**Data Length**			1
	Suggested Data Array Formats			Float
	Data Item 1			
	Measurement	**Units**	**Values Range**	**Invalid Value**
	Power	KiloWatts	0 .. 6,553.5	N/A
SNVT_ppm	**Data Length**			1
	Suggested Data Array Formats			Uint16, Float
	Data Item 1			
	Measurement	**Units**	**Values Range**	**Invalid Value**
	Concentration	Parts per Million (ppm)	0 .. 65,535	N/A
SNVT_ppm_f	**Data Length**			1
	Suggested Data Array Formats			Uint16, Float
	Data Item 1			
	Measurement	**Units**	**Values Range**	**Invalid Value**
	Concentration	Parts per Million	0 .. 3.40282E38	N/A

SNVT_preset	**Data Length**			11
	Suggested Data Array Formats			Uint16, Float
	Data Item 1			preset_learn
	Measurement	**Units**	**Values Range**	**Invalid Value**
	learn_mode_t	N/A	0 .. 3	255 (0xFF) (LN_NUL)
	Data Item 2			preset_selector
	Selector	N/A	0 .. 65,535	N/A
	Data Item 3-6			preset_value00..preset_value03
	Value	Specific to SNVT	0 .. 255	N/A
	Data Item 7			preset_day
	Days	Days	0 .. 65,534	65,535 (0xFFFF)
	Data Item 8			preset_hour
	Hours	Hour	0 .. 23	N/A
	Data Item 9			preset_minute
	Minutes	Minutes	0 .. 59	N/A
	Data Item 10			preset_second
	Seconds	Seconds	0 .. 59	N/A
SNVT_preset	**Data Item 11**			preset_millisec
	Milliseconds	Milliseconds	0 .. 999	N/A
SNVT_press	**Data Length**			1
	Suggested Data Array Formats			Float
	Data Item 1			
	Measurement	**Units**	**Values Range**	**Invalid Value**
	Pressure (gauge)	KiloPascals	-3,276.8 .. 3,276.6	Type: 3,276.7 Raw: 32,767 (0x7FFF)
SNVT_press_f	**Data Length**			1
	Suggested Data Array Formats			Float
	Data Item 1			
	Measurement	**Units**	**Values Range**	**Invalid Value**
	Pressure (gauge)	Pascals	-3.40282E38 .. 3.40282E38	N/A
SNVT_press_p	**Data Length**			1
	Suggested Data Array Formats			Sint16, Float
	Data Item 1			
	Measurement	**Units**	**Values Range**	**Invalid Value**
	Pressure (gauge)	Pascals	-32,768 .. 32,766	32,767 (0x7FFF)
SNVT_privacyzone	**Data Length**			3
	Suggested Data Array Formats			Uint16, Float
	Data Item 1			priv_zn_action
	Measurement	**Units**	**Values Range**	**Invalid Value**
	privacyzone_t	N/A	0 .. 5	255 (0xFF) (PZ_NUL)
	Data Item 2			priv_zn_number
	Zone number	Privacy zone number	1 .. 255	0
	Data Item 3			priv_zn_camera_id
	Camera ID	ID number	1 .. 65,535	0
SNVT_ptz	**Data Length**			6
	Suggested Data Array Formats			Float
	Data Item 1			ptz_pan_dir
	Measurement	**Units**	**Values Range**	**Invalid Value**
	ptz_pan_dir	Direction	0 .. 255	N/A
	Data Item 2			ptz_pan_speed
	Pan speed	Percent of full level	0.0 .. 100.0	N/A
	Data Item 3			ptz_tilt_dir
	tilt_dir_t	N/A	0 .. 2	255 (0xFF) (TILT_NUL)
	Data Item 4			ptz_tilt_speed
	Tilt speed	Percent of full level	0.0 .. 100.0	N/A
	Data Item 5			ptz_zoom
	zoom_t	N/A	0 .. 2	255 (0xFF) (ZOOM_NUL)
	Data Item 6			ptz_zoom_speed
	ptz_zoom_speed	Meters (m)	-3.40282E38 .. 3.40282E38	N/A

SNVT_pump_ sensor	**Data Length**			13
	Suggested Data Array Formats			Float
	Data Item 1			pump_sn_rot_speed
	Measurement	**Units**	**Values Range**	**Invalid Value**
	Frequency	Hertz	0 .. 6,553.4	N/A
	Data Item 2			pump_sn_body_temp
	Body temperature	Degrees Celsius	-274.0 .. 6,279.4	N/A
	Data Item 3			pump_sn_mot_ext_temp
	Motor external temp	Degrees Celsius	-274.0 .. 6,279.4	N/A
	Data Item 4			pump_sn_mot_int_temp
	Motor internal temp	Degrees Celsius	-274.0 .. 6,279.4	N/A
	Data Item 5			pump_sn_mot_overload
	Boolean	Bit	0, 1	255 (0xFF) (BOOL_NUL)
	Data Item 6			pump_sn_oil_low
	Boolean	Bit	0, 1	255 (0xFF) (BOOL_NUL)
	Data Item 7			pump_sn_ph_imbal
	Boolean	Bit	0, 1	255 (0xFF) (BOOL_NUL)
	Data Item 8			pump_sn_curr_use
	Electric current use	Ampere	-3,276.8..3,276.6	N/A
	Data Item 9			pump_sn_pwr_usage
	Power usage	Kilowatts	0 .. 6,553.4	N/A
	Data Item 10			pump_sn_temp_ctrl
	unit_temp_t	N/A	0 .. 3	255 (0xFF) (TEMP_NUL)
	Data Item 11			pump_sn_em_brake_act
	Boolean	Bit	0, 1	255 (0xFF) (BOOL_NUL)
	Data Item 12			pump_sn_fric_brk_act
	Boolean	Bit	0, 1	255 (0xFF) (BOOL_NUL)
	Data Item 13			pump_sn_gas_brk_act
	Boolean	Bit	0, 1	255 (0xFF) (BOOL_NUL)
SNVT_pumpset_mn	**Data Length**			8
	Suggested Data Array Formats			Byte, Uint16, Float
	Data Item 1			pset_main_pump
	Measurement	**Units**	**Values Range**	**Invalid Value**
	motor_state_t	N/A	0 .. 7	255 (0xFF) (MOTOR_NUL)
	Data Item 2			pset_booster_pump
	motor_state_t	N/A	0 .. 7	255 (0xFF) (MOTOR_NUL)
	Data Item 3			pset_prio_level
	priority_level_t	N/A	0 .. 11	255 (0xFF) (PR_NUL)
	Data Item 4			pset_process_ready
	Boolean_t	Bit	0, 1	255 (0xFF) (BOOL_NUL)
	Data Item 5			pset_emerg_stop
	Boolean_t	Bit	0, 1	255 (0xFF) (BOOL_NUL)
	Data Item 6			pset_main_pumpd_ena
	Boolean_t	Bit	0, 1	255 (0xFF) (BOOL_NUL)
	Data Item 7			pset_boost_pumpd_ena
	Boolean_t	Bit	0, 1	255 (0xFF) (BOOL_NUL)
	Data Item 8			pset_maint_req
	Boolean_t	Bit	0, 1	255 (0xFF) (BOOL_NUL)
SNVT_pumpset_sn	**Data Length**			14
	Suggested Data Array Formats			Float
	Data Item 1			pset_sn_dil_flow
	Measurement	**Units**	**Values Range**	**Invalid Value**
	Total Dilution Flow Volume	Milliliters per second	0 .. 65,534	N/A
	Data Item 2			pset_sn_exh_temp
	Exhaust Temperature	Degrees Celsius	-274.0 .. 6,279.4	N/A
	Data Item 3			pset_sn_exh_press
	Exhaust Pressure	Kilo-pascals	-3,276.8..3,276.6	N/A
	Data Item 4			pset_sn_seal_press
	Shaft seal purge pressure	Kilo-pascals	-3,276.8..3,276.6	N/A

	Data Item 5			pset_sn_inlet_vacuum
	Inlet vacuum pressure	Kilo-pascals	-3.40282E38 .. 3.40282E38	N/A
	Data Item 6			pset_sn_supply_volts
	Supply voltage	Volts	-3,276.8..3,276.6	N/A
	Data Item 7			pset_sn_cool_flow
	Total coolant flow	Milliliters per second	0 .. 65,534	N/A
	Data Item 8			pset_sn_cool_fl_low
	Boolean	Bit	0, 1	255 (0xFF) (BOOL_NUL)
	Data Item 9			pset_sn_dil_active
	Boolean	Bit	0, 1	255 (0xFF) (BOOL_NUL)
	Data Item 10			pset_sn_bal_dil_act
	Boolean	Bit	0, 1	255 (0xFF) (BOOL_NUL)
	Data Item 11			pset_sn_inl_p_dil_act
	Boolean	Bit	0, 1	255 (0xFF) (BOOL_NUL)
	Data Item 12			pset_sn_exh_dil_act
	Boolean	Bit	0, 1	255 (0xFF) (BOOL_NUL)
	Data Item 13			pset_sn_dil_fl_oor
	Boolean	Bit	0, 1	255 (0xFF) (BOOL_NUL)
	Data Item 14			pset_sn_pwr_sup_on
	Boolean	Bit	0, 1	255 (0xFF) (BOOL_NUL)
SNVT_pwr_fact	**Data Length**			1
	Suggested Data Array Formats			Float
	Data Item 1			
	Measurement	**Units**	**Values Range**	**Invalid Value**
	Power Factor	Multiplier	-1.00000..1.00000	N/A
SNVT_pwr_fact_f	**Data Length**			1
	Suggested Data Array Formats			Float
	Data Item 1			
	Measurement	**Units**	**Values Range**	**Invalid Value**
	Power Factor	N/A	-1 .. 1	N/A
SNVT_reg_val	**Data Length**			3
	Suggested Data Array Formats			Sint32
	Data Item 1			reg_val_value
	Measurement	**Units**	**Values Range**	**Invalid Value**
	Raw value	Defined by unit field	-2,147,483,638 .. 2,147,483,647	N/A
	Data Item 2			reg_val_unit
	reg_val_unit_t	N/A	0 .. 43	255 (0xFF) (RVU_NUL)
	Data Item 3			reg_val_decimals
	Decimal Place	Digits after Decimal	0 ..7	N/A
SNVT_reg_val_ts	**Data Length**			9
	Suggested Data Array Formats			Sint32
	Data Item 1			reg_val_ts_raw
	Measurement	**Units**	**Values Range**	**Invalid Value**
	Raw Value		-2,147,483,648 .. 2,147,483,647	N/A
	Data Item 2			reg_val_ts_unit
	reg_val_unit_t	N/A	0 .. 43	255 (0xFF) (RVU_NUL)
	Data Item 3			reg_val_ts_bits
	See SNVT tables	See SNVT tables	0 .. 255	N/A
	Data Item 4			reg_val_ts_year
	Year	Year	0 .. 300	N/A
	Data Item 5			reg_val_ts_month
	Month	Month of Year	0 .. 12	N/A
	Data Item 6			reg_val_ts_day
	Day	Day of Month	0 .. 31	N/A
	Data Item 7			reg_val_ts_hour
	Hour	Hour of day	0 .. 23	N/A
	Data Item 8			reg_val_ts_min
	Minutes	Minute of Hour	0 .. 59	N/A

	Data Item 9			reg_val_ts_secs
	Minutes	Minute of Hour	0 .. 59	N/A
SNVT_res	**Data Length**			1
	Suggested Data Array Formats			Float
	Data Item 1			
	Measurement	**Units**	**Values Range**	**Invalid Value**
	Electrical Resistance	Ohms	0 .. 6,553.5	N/A
SNVT_res_f	**Data Length**			1
	Suggested Data Array Formats			Float
	Data Item 1			
	Measurement	**Units**	**Values Range**	**Invalid Value**
	Electrical Resistance	Ohms	0 .. 3.40282E38	N/A
SNVT_res_kilo	**Data Length**			1
	Suggested Data Array Formats			Float
	Data Item 1			
	Measurement	**Units**	**Values Range**	**Invalid Value**
	Electrical Resistance	KiloOhms	0 .. 6,553.5	N/A
SNVT_rpm	**Data Length**			1
	Suggested Data Array Formats			Uint16, Float
	Data Item 1			
	Measurement	**Units**	**Values Range**	**Invalid Value**
	Angular Velocity	Revolutions per Minute (RPM)	0 .. 65,534	65,535 (0xFFFF)
SNVT_sblnd_state	**Data Length**			5
	Suggested Data Array Formats			Float
	Data Item 1			function
	Measurement	**Units**	**Values Range**	**Invalid Value**
	setting_t	N/A	0 .. 5	255 (SET_NUL)
	Data Item 2			setting
	Scene setting Level	Percent of Full Level	0 .. 100	N/A
	Data Item 3			rotation
	Rotation Angle	Degrees	-359.98 .. 360.00	32,767 (0x7FFF)
	Data Item 4			cmd_source
	sblnd_cmd_source_t	N/A	0 .. 127	-1 (0xFF)
	Data Item 5			error_code
	sblnd_error_t	N/A	0 .. 14	-1 (0xFF)
SNVT_scene	**Data Length**			2
	Suggested Data Array Formats			Byte, Uint16, Float
	Data Item 1			Function
	Measurement	**Units**	**Values Range**	**Invalid Value**
	Scene_t	N/A	0 .. 23	255 (0xFF) (SC_NUL)
	Data Item 2			Scene_Number
	Scene Number	Scene Number	1 .. 255	N/A
SNVT_scene_cfg	**Data Length**			7
	Suggested Data Array Formats			Float
	Data Item 1			scn_cfg_func
	Measurement	**Units**	**Values Range**	**Invalid Value**
	scene_config_t	N/A	0 .. 4	255 (0xFF) (SCF_NUL)
	Data Item 2			scn_cfg_scene
	Scene Number	Scene Number	1 .. 255	N/A
	Data Item 3			scn_cfg_setting
	Scene Setting Level	Percent of Full Level	0 .. 100.0	N/A
	Data Item 4			scn_cfg_rotation
	Scene Rotation Angle	Degrees	-359.98 .. 360.00	Raw: 32,767 (0x7FFF) Type: 655.34
	Data Item 5			scn_cfg_fade_tim
	Scene Fade Time	Seconds	0.0 to 6,553.4	Raw: 65,535 (0xFFFF) Type: 6553.5

	Data Item 6			scn_cfg_dly_tim
	Scene Delay Time	Seconds	0.0 to 6,553.4	Raw: 65,535 (0xFFFF) Type: 6553.5
	Data Item 7			scn_cfg_scn_proi
	Scene Priority	Priority Value	0 .. 255	N/A
SNVT_setting	**Data Length**			3
	Suggested Data Array Formats			Float
	Data Item 1			setting_function
	Measurement	**Units**	**Values Range**	**Invalid Value**
	setting_t	N/A	0 .. 5	255 (0xFF)(SET_NUL)
	Data Item 2			setting_setting
	Scene Setting Level	Percent of Full Level	0 .. 100.0	N/A
	Data Item 3			setting_rotation
	Rotation Angle	Degrees	-359.98 .. 360.00	Raw: (0x7FFF) 32,767 Type: 655.34
SNVT_smo_obscur	**Data Length**			1
	Suggested Data Array Formats			Float
	Data Item 1			
	Measurement	**Units**	**Values Range**	**Invalid Value**
	Smoke Obscuration	Percent Obscuration	0.000 .. 5.000	N/A
SNVT_sound_db	**Data Length**			1
	Suggested Data Array Formats			Float
	Data Item 1			
	Measurement	**Units**	**Values Range**	**Invalid Value**
	Sound Level	Decibels	-327.68 .. 327.67	N/A
SNVT_sound_db_f	**Data Length**			1
	Suggested Data Array Formats			Float
	Data Item 1			
	Measurement	**Units**	**Values Range**	**Invalid Value**
	Sound Level	Decibels from Sound Pressure Level	-3.40282E38 .. 3.40282E38	N/A
SNVT_speed	**Data Length**			1
	Suggested Data Array Formats			Float
	Data Item 1			
	Measurement	**Units**	**Values Range**	**Invalid Value**
	Linear Velocity	Meters per Second	0 .. 6,553.5	N/A
SNVT_speed_f	**Data Length**			1
	Suggested Data Array Formats			Float
	Data Item 1			
	Measurement	**Units**	**Values Range**	**Invalid Value**
	Speed	Meters per Second	-3.40282E38 .. 3.40282E38	N/A
SNVT_speed_mil	**Data Length**			1
	Suggested Data Array Formats			Float
	Data Item 1			
	Measurement	**Units**	**Values Range**	**Invalid Value**
	Linear Velocity	Millimeters per Second	0 .. 65,535	N/A
SNVT_state	**Data Length**			1
	Suggested Data Array Formats			Uint16, Float
	Data Item 1			
	Measurement	**Units**	**Values Range**	**Invalid Value**
	16 state bits	Bits	0 .. 65,535	N/A
SNVT_state_64	**Data Length**			4
	Suggested Data Array Formats			Packed_Bit, Uint16, Float
	Data Item 1			state_64_word0
	Measurement	**Units**	**Values Range**	**Invalid Value**
	Boolean	Bit	0, 1 per bit (16 bits in word)	N/A

	Data Item 2			state_64_word1
	Boolean	Bit	0, 1 per bit (16 bits in word)	N/A
	Data Item 3			state_64_word2
	Boolean	Bit	0, 1 per bit (16 bits in word)	N/A
	Data Item 4			state_64_word3
	Boolean	Bit	0, 1 per bit (16 bits in word)	N/A
SNVT_str_asc	**Data Length**			31
	Suggested Data Array Formats			Byte, Uint16, Float
	Data Item 1 - 31			ascii00, ascii01, .. ascii30
	Measurement	**Units**	**Values Range**	**Invalid Value**
	Character String	N/A	0 .. 255	N/A
SNVT_str_int	**Data Length**			16
	Suggested Data Array Formats			Uint16, Float
	Data Item 1			char_set
	Measurement	**Units**	**Values Range**	**Invalid Value**
	Locale Code	N/A	0 .. 255	N/A
	Data Item 2 - 16			wide_char00, .. wide_char14
	Wide-character string with NULL terminator	N/A	0 .. 65,535	N/A
SNVT_switch	**Data Length**			2
	Suggested Data Array Formats			Float
	Data Item 1			Value
	Measurement	**Units**	**Values Range**	**Invalid Value**
	Value	Percent of full scale	0 .. 100.0	N/A
	Data Item 2			State
	State	N/A	0 .. 1	-1
SNVT_telcom	**Data Length**			1
	Suggested Data Array Formats			Byte, Uint16, Float
	Data Item 1			
	Measurement	**Units**	**Values Range**	**Invalid Value**
	telcom_states_t	N/A	0 .. 20	255 (0xFF) (TEL_NUL)
SNVT_temp	**Data Length**			1
	Suggested Data Array Formats			Float
	Data Item 1			
	Measurement	**Units**	**Values Range**	**Invalid Value**
	Temperature	Degrees Celsius	-274.0 .. 6,279.5	N/A
SNVT_temp_diff_p	**Data Length**			1
	Suggested Data Array Formats			Float
	Data Item 1			
	Measurement	**Units**	**Values Range**	**Invalid Value**
	Temp. difference	Degrees Celsius	-327.68 .. 327.66	Raw: 32,767 (0x7FFF) Type: 655.34
SNVT_temp_f	**Data Length**			1
	Suggested Data Array Formats			Float
	Data Item 1			
	Measurement	**Units**	**Values Range**	**Invalid Value**
	Incremental Count	Degrees Celsius	-3.40282E38 .. 3.40282E38	N/A
SNVT_temp_p	**Data Length**			1
	Suggested Data Array Formats			Float
	Data Item 1			
	Measurement	**Units**	**Values Range**	**Invalid Value**
	Temperature	Degrees Celsius	-273.17 .. 327.66	Type: 327.67 Raw: 32,767 (0x7FFF)

SNVT_temp_ror	**Data Length**			1
	Suggested Data Array Formats			Float
	Data Item 1			
	Measurement	**Units**	**Values Range**	**Invalid Value**
	Temperature Rate of Change/Rise	Degrees Celsius per Minute (˚C/min)	-16,384.0.. 16,383.0	Raw: 32,767 (0x7FFF) Type: 16,383.5
SNVT_temp_setpt	**Data Length**			6
	Suggested Data Array Formats			Float
	Data Item 1			Occupied_cool
	Measurement	**Units**	**Values Range**	**Invalid Value**
	Occupied Cooling Setpoint	Degrees Celsius	-273.17 .. 327.66	Type: 327.67 Raw: 32,767 (0x7FFF)
	Data Item 2			Standby_cool
	Standby Cooling Setpoint	Degrees Celsius	-273.17 .. 327.66	Type: 327.67 Raw: 32,767 (0x7FFF)
	Data Item 3			Unoccupied_cool
	Unoccupied Cooling Setpoint	Degrees Celsius	-273.17 .. 327.66	Type: 327.67 Raw: 32,767 (0x7FFF)
	Data Item 4			Occupied_heat
	Occupied Heating Setpoint	Degrees Celsius	-273.17 .. 327.66	Type: 327.67 Raw: 32,767 (0x7FFF)
	Data Item 5			Standby_heat
	Standby Heating Setpoint	Degrees Celsius	-273.17 .. 327.66	Type: 327.67 Raw: 32,767 (0x7FFF)
	Data Item 6			Unoccupied_heat
	Unoccupied Heating Setpoint	Degrees Celsius	-273.17 .. 327.66	Type: 327.67 Raw: 32,767 (0x7FFF)
SNVT_therm_mode	**Data Length**			1
	Suggested Data Array Formats			Byte, Uint16, Float
	Data Item 1			
	Measurement	**Units**	**Values Range**	**Invalid Value**
	therm_mode_t	N/A	0 .. 2	255 (0xFF) (THERM_NUL)
SNVT_time_f	**Data Length**			1
	Suggested Data Array Formats			Float
	Data Item 1			
	Measurement	**Units**	**Values Range**	**Invalid Value**
	Elapsed Time	Seconds	-3.40282E38.. 3.40282E38	N/A
SNVT_time_hour	**Data Length**			1
	Suggested Data Array Formats			Uint16, Float
	Data Item 1			
	Measurement	**Units**	**Values Range**	**Invalid Value**
	Elapsed Time	Hours	0 .. 65,535	N/A
SNVT_time_min	**Data Length**			1
	Suggested Data Array Formats			Float
	Data Item 1			
	Measurement	**Units**	**Values Range**	**Invalid Value**
	Elapsed Time	Minutes	0 .. 65,535	N/A
SNVT_time_sec	**Data Length**			1
	Suggested Data Array Formats			Float
	Data Item 1			
	Measurement	**Units**	**Values Range**	**Invalid Value**
	Elapsed time	Seconds	0 .. 6,553.4	65,535 (0xFFFF)
SNVT_time_stamp	**Data Length**			6
	Suggested Data Array Formats			Uint16, Float
	Data Item 1			ts_year
	Measurement	**Units**	**Values Range**	**Invalid Value**
	Year	Year	0 .. 3000	65,535 (0xFFFF)
	Data Item 2			ts_month
	Month	Month of Year	0 .. 12	N/A

	Data Item 3			ts_day
	Day	Day of Month	0 .. 31	N/A
	Data Item 4			ts_hour
	Hours	Hours of Day	0 .. 23	N/A
	Data Item 5			ts_minute
	Minutes	Minute of Hour	0 .. 59	N/A
	Data Item 6			ts_second
	Second	Second of Minute	0 .. 59	N/A
SNVT_time_zone (only raw data support)	**Data Length**			15
	Suggested Data Array Formats			Byte
SNVT_tod_event	**Data Length**			3
	Suggested Data Array Formats			Uint16, Float
	Data Item 1			tod_cur_state
	Measurement	**Units**	**Values Range**	**Invalid Value**
	occup_t	N/A	0 .. 3	255 (0xFF) (OC_NUL)
	Data Item 2			tod_nxt_state
	occup_t	N/A	0 .. 3	255 (0xFF) (OC_NUL)
	Data Item 3			tod_tm_to_nxt
	Time to Next State	Minute of Hour	0 .. 65,535	N/A
SNVT_trans_table	**Data Length**			8
	Suggested Data Array Formats			Float
	Data Item 1-7			tr_point00, ..tr_point06
	Measurement	**Units**	**Values Range**	**Invalid Value**
	Axis-points Array	N/A	-3.40282E38.. 3.40282E38	N/A
	Data Item 8			interp_methods
	interp_t	N/A	0 .. 65,535	N/A
SNVT_turbidity	**Data Length**			1
	Suggested Data Array Formats			Float
	Data Item 1			
	Measurement	**Units**	**Values Range**	**Invalid Value**
	Turbidity	Nephelometric Turbidity Units (NTU)	0 .. 65.534	65,535 (0xFFFF)
SNVT_turbidity_f	**Data Length**			1
	Suggested Data Array Formats			Float
	Data Item 1			
	Measurement	**Units**	**Values Range**	**Invalid Value**
	Turbidity	Nephelometric Turbidity Units (NTU)	0 .. 3.40282E38	N/A
SNVT_valve_mode	**Data Length**			1
	Suggested Data Array Formats			Byte, Uint16, Float
	Data Item 1			
	Measurement	**Units**	**Values Range**	**Invalid Value**
	valve_mode_t	N/A	0 .. 7	255 (0xFF) (VALVE_NUL)
SNVT_vol	**Data Length**			1
	Suggested Data Array Formats			Float
	Data Item 1			
	Measurement	**Units**	**Values Range**	**Invalid Value**
	Volume	Liters	0 .. 6,553.5	N/A
SNVT_vol_f	**Data Length**			1
	Suggested Data Array Formats			Float
	Data Item 1			
	Measurement	**Units**	**Values Range**	**Invalid Value**
	Volume	Liters	0 .. 3.40282E38	N/A
SNVT_vol_kilo	**Data Length**			1
	Suggested Data Array Formats			Float
	Data Item 1			
	Measurement	**Units**	**Values Range**	**Invalid Value**
	Volume	Kiloliters (kl)	0 .. 6,553.5	N/A

SNVT_vol_mil	**Data Length**			1
	Suggested Data Array Formats			Float
	Data Item 1			
	Measurement	**Units**	**Values Range**	**Invalid Value**
	Volume	Milliliters	0 .. 6,553.5	N/A
SNVT_volt	**Data Length**			1
	Suggested Data Array Formats			Float
	Data Item 1			
	Measurement	**Units**	**Values Range**	**Invalid Value**
	Electric Voltage	Volts	-3,276.8..3,276.7	N/A
SNVT_volt_ac	**Data Length**			1
	Suggested Data Array Formats			Uint16, Float
	Data Item 1			
	Measurement	**Units**	**Values Range**	**Invalid Value**
	Electric Voltage	Volts, Alt. Current (VAC)	0 .. 65,534	65,535 (0xFFFF)
SNVT_volt_dbmv	**Data Length**			1
	Suggested Data Array Formats			Float
	Data Item 1			
	Measurement	**Units**	**Values Range**	**Invalid Value**
	Electric Voltage	Decibels-milliVolts, Direct Current	-3,276.8.. 3,276.7	N/A
SNVT_volt_f	**Data Length**			1
	Suggested Data Array Formats			Float
	Data Item 1			
	Measurement	**Units**	**Values Range**	**Invalid Value**
	Electric Voltage	Volts	-3.40282E38 .. 3.40282E38	N/A
SNVT_volt_kilo	**Data Length**			1
	Suggested Data Array Formats			Float
	Data Item 1			
	Measurement	**Units**	**Values Range**	**Invalid Value**
	Electric Voltage	Kilovolts	-3,276.8.. 3,276.7	N/A
SNVT_volt_mil	**Data Length**			1
	Suggested Data Array Formats			Float
	Data Item 1			
	Measurement	**Units**	**Values Range**	**Invalid Value**
	Electric Voltage	Millivolts	-3,276.8.. 3,276.7	N/A
SNVT_zerospan	**Data Length**			2
	Suggested Data Array Formats			Float
	Data Item 1			zero
	Measurement	**Units**	**Values Range**	**Invalid Value**
	Zero-term	Percent (%) or Parts-per-Million (ppm)	-163.840.. 163.835	N/A
	Data Item 2			span
	Span-factor	Percent (%) or Parts-per-Million (ppm)	0 .. 32.7675	N/A

Appendix D.4 SNVT_Type by Number

SNVT Type Number	SNVT Type Name	SNVT Type Number	SNVT Type Name
1	SNVT_amp	83	SNVT_state
2	SNVT_amp_mil	84	SNVT_time_stamp
3	SNVT_angle	85	SNVT_zerospan
4	SNVT_angle_vel	86	SNVT_magcard
5	SNVT_btu_kilo	87	SNVT_elapsed_tm
6	SNVT_btu_mega	88	SNVT_alarm
7	SNVT_char_ascii	89	SNVT_currency
8	SNVT_count	90	SNVT_file_pos
9	SNVT_count_inc	91	SNVT_muldiv
10	SNVT_date_cal	92	SNVT_obj_request
11	SNVT_date_day	93	SNVT_obj_status
12	SNVT_date_time	94	SNVT_preset
13	SNVT_elec_kwh	95	SNVT_switch
14	SNVT_elec_whr	96	SNVT_trans_table
15	SNVT_flow	97	SNVT_override
16	SNVT_flow_mil	98	SNVT_pwr_fact
17	SNVT_length	99	SNVT_pwr_fact_f
18	SNVT_length_kilo	100	SNVT_density
19	SNVT_length_micr	101	SNVT_density_f
20	SNVT_length_mil	102	SNVT_rpm
21	SNVT_lev_cont	103	SNVT_hvac_emerg
22	SNVT_lev_disc	104	SNVT_angle_deg
23	SNVT_mass	105	SNVT_temp_p
24	SNVT_mass_kilo	106	SNVT_temp_setpt
25	SNVT_mass_mega	107	SNVT_time_sec
26	SNVT_mass_mil	108	SNVT_hvac_mode
27	SNVT_power	109	SNVT_occupancy
28	SNVT_power_kilo	110	SNVT_area
29	SNVT_ppm	111	SNVT_hvac_overid
30	SNVT_press	112	SNVT_hvac_status
31	SNVT_res	113	SNVT_press_p
32	SNVT_res_kilo	114	SNVT_address
33	SNVT_sound_db	115	SNVT_scene
34	SNVT_speed	116	SNVT_scene_cfg
35	SNVT_speed_mil	117	SNVT_setting
36	SNVT_str_asc	118	SNVT_evap_state
37	SNVT_str_int	119	SNVT_therm_mode
38	SNVT_telcom	120	SNVT_defr_mode
39	SNVT_temp	121	SNVT_defr_term
40	SNVT_time_passed	122	SNVT_defr_state
41	SNVT_vol	123	SNVT_time_min
42	SNVT_vol_kilo	124	SNVT_time_hour
43	SNVT_vol_mil	125	SNVT_ph
44	SNVT_volt	126	SNVT_ph_f
45	SNVT_volt_dbmv	127	SNVT_chlr_status
46	SNVT_volt_kilo	128	SNVT_tod_event
47	SNVT_volt_mil	129	SNVT_smo_obscur
48	SNVT_amp_f	130	SNVT_fire_test
49	SNVT_angle_f	131	SNVT_temp_ror
50	SNVT_angle_vel_f	132	SNVT_fire_init
51	SNVT_count_f	133	SNVT_fire_indcte
52	SNVT_count_inc_f	134	SNVT_time_zone
53	SNVT_flow_f	135	SNVT_earth_pos
54	SNVT_length_f	136	SNVT_reg_val
55	SNVT_lev_cont_f	137	SNVT_reg_val_ts
56	SNVT_mass_f	138	SNVT_volt_ac
57	SNVT_power_f	139	SNVT_amp_ac
58	SNVT_ppm_f	143	SNVT_turbidity

SNVT Type Number	SNVT Type Name	SNVT Type Number	SNVT Type Name
59	SNVT_press_f	144	SNVT_turbidity_f
60	SNVT_res_f	145	SNVT_hvac_type
61	SNVT_sound_db_f	146	SNVT_elec_kwh_l
62	SNVT_speed_f	147	SNVT_temp_diff_p
63	SNVT_temp_f	148	SNVT_ctrl_req
64	SNVT_time_f	149	SNVT_ctrl_resp
65	SNVT_vol_f	150	SNVT_ptz
66	SNVT_volt_f	151	SNVT_privacyzone
67	SNVT_btu_f	152	SNVT_pos_ctrl
68	SNVT_elec_whr_f	153	SNVT_enthalpy
69	SNVT_config_src	154	SNVT_gfci_status
70	SNVT_color	155	SNVT_motor_state
71	SNVT_grammage	156	SNVT_pumpset_mn
72	SNVT_grammage_f	157	SNVT_ex_control
73	SNVT_file_req	158	SNVT_pumpset_sn
74	SNVT_file_status	159	SNVT_pump_sensor
75	SNVT_freq_f	160	SNVT_abs_humid
76	SNVT_freq_hz	161	SNVT_flow_p
77	SNVT_freq_kilohz	162	SNVT_dev_c_mode
78	SNVT_freq_milhz	163	SNVT_valve_mode
79	SNVT_lux	164	SNVT_alarm_2
80	SNVT_ISO_7811	165	SNVT_state_64
81	SNVT_lev_percent	166	SNVT_nv_type
82	SNVT_multiplier		

Appendix D.5 How to Interpret an XIF file

An XIF file can be viewed with a viewer program such as Notepad. The file contents start with a header section. The first three lines are for informational purposes, followed by a 4th blank line:

File: fserver.xif generated by LonDriver Revision 1.30(e), XIF Version 4.0

Copyright (c) 2000-2012 by Sierra Monitor Corporation

All Rights Reserved. Run on Tue Jun 18 10:47:15 2013

The fifth line contains the program ID:

90:00:95:47:1E:02:04:D7

Lines 6-10 contain encoded information on the capabilities of the LonWorks device. Refer to the LonMark Device Interface File Reference Guide for more information.

Line 11 contains a single asterisk indicating the end of the device information.

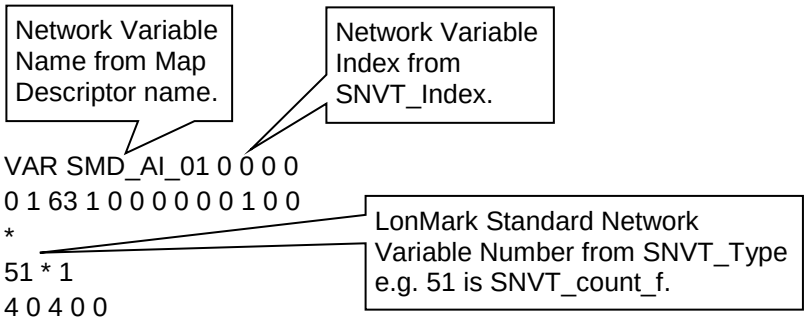
*

Line 12 contains the device self-documentation string, which is the same as the FieldServer title, specified in the configuration file:

":D48:S01:DCC085 X30.CSV v4.10c

The next section contains the Network Variable definitions. We will only discuss the encoded information of importance to the FieldServer. Refer to the LonMark Device Interface File Reference Guide for more information.

An example of a Standard Network Variable Type (SNVT) Definition:

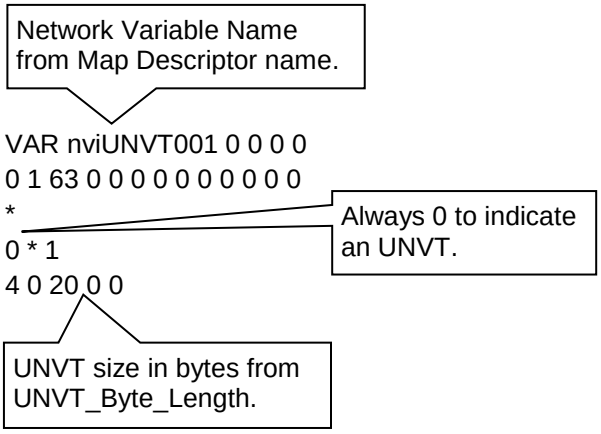


This is the Network Variable's entry in the configuration file:

```
Map_Descriptors
Map_Descriptor_Name     ,Data_Array_Name    ,Data_Array_Offset    ,Lon_Function    ,Function
SMD_AI_01               ,DA_AI_01           ,0                    ,NVPO            ,Server
```

```
, Node_Name     , SNVT_Index    ,SNVT_Type
, Lon_srv_1     ,0              ,SNVT_count_f
```

An example of a User Defined Network Variable Type (UNVT) Definition:



This is the Network Variable's entry in the configuration file:

```
Map_Descriptors
Map_Descriptor_Name , Data_Array_Name , Data_Array_Offset , Lon_Function , Function
nviUNVT001          , DA_BYTE         , 0                 , NVUI         , SERVER
```

```
, Node_Name , SNVT_Type , UNVT_Byte_Length
, Lon_1     , Not SNVT  , 20
```