

ARTISAN®
TECHNOLOGY GROUP

Your **definitive** source for quality pre-owned equipment.

Artisan Technology Group
(217) 352-9330 | sales@artisantg.com | artisantg.com

Full-service, independent repair center
with experienced engineers and technicians on staff.

We buy your excess, underutilized, and idle equipment
along with credit for buybacks and trade-ins.

Custom engineering
so your equipment works exactly as you specify.

- Critical and expedited services
- In stock / Ready-to-ship
- Leasing / Rentals / Demos
- ITAR-certified secure asset solutions

Expert team | Trust guarantee | 100% satisfaction

All trademarks, brand names, and brands appearing herein are the property of their respective owners.

Find the *AIM APU429-16* at our website: *Click HERE*

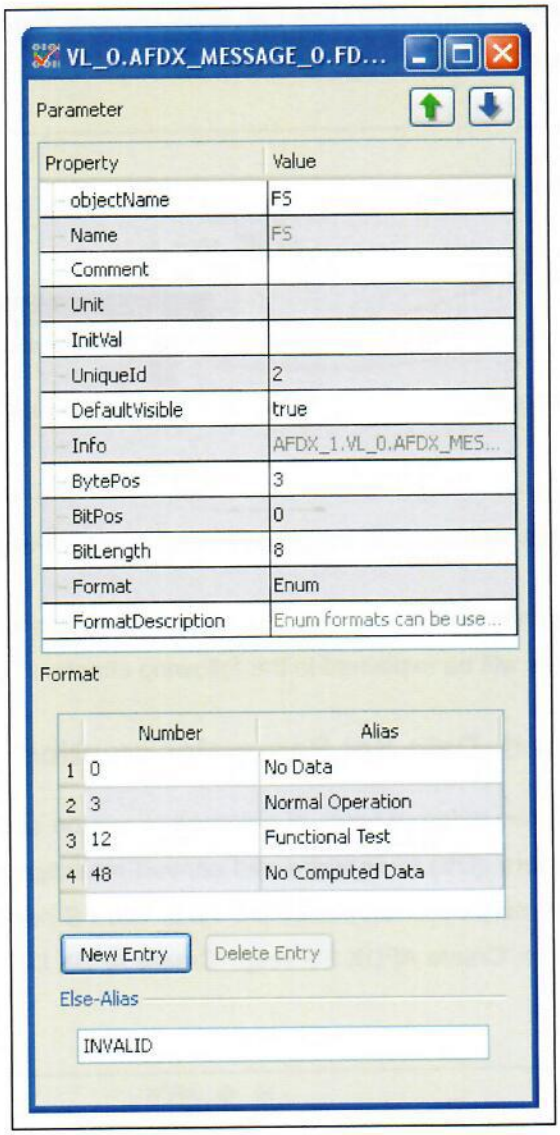


Figure 65: AFDX - FS DBS configuration

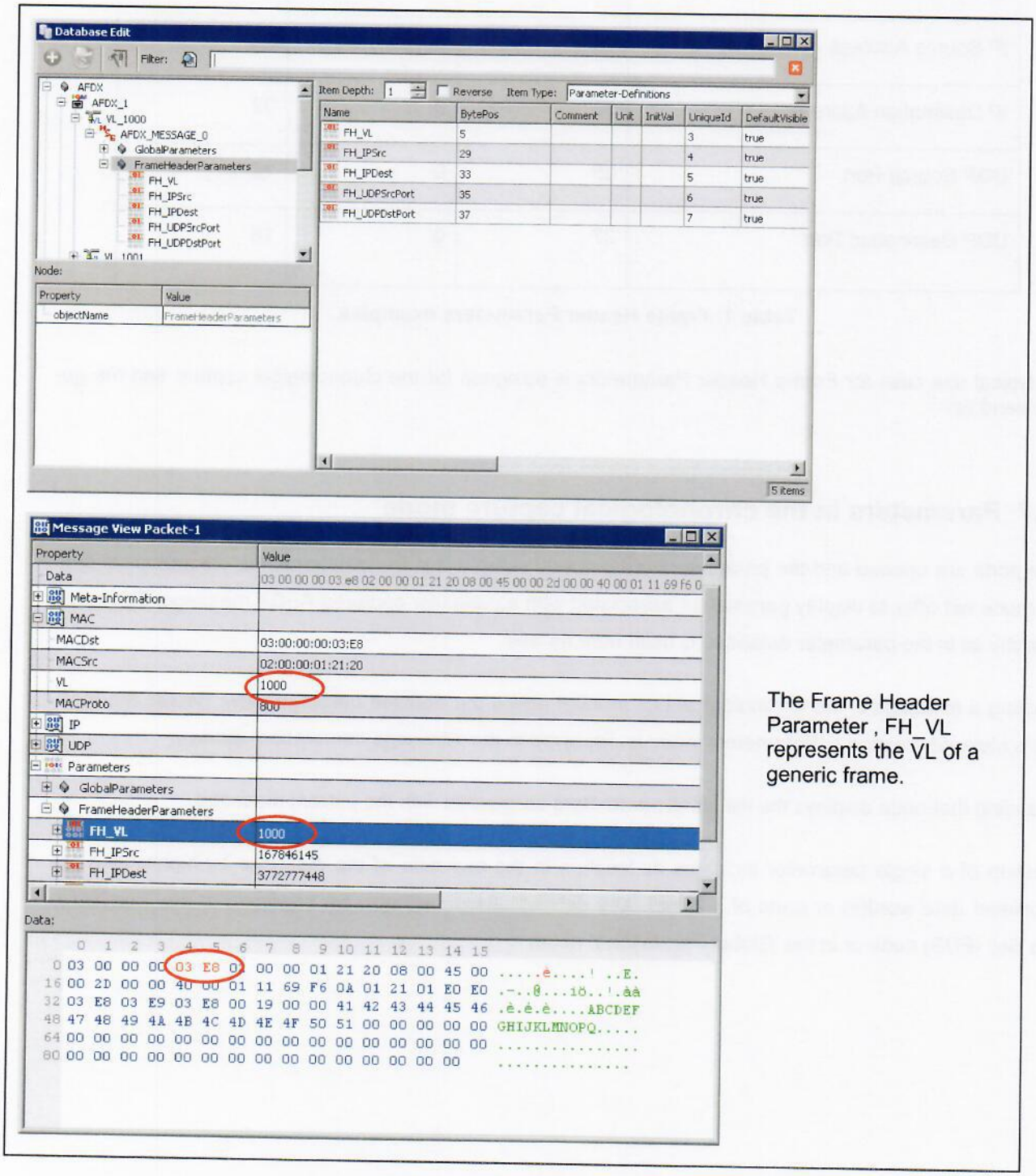
See chapter 3, Edit Parameter Formats of this document for further details on editing Parameters.

2.3.6 Frame Header Parameter creation

Frame Header Parameters can created from the context menu of the 'FrameHeaderParameters' node. The difference to the Parameters described above (Global Parameters and Functional Status) is that the Byte Position 0 starts at the first Byte within the Frame Header and not at the beginning of the Payload. Frame Header Parameters can be used to retrieve values from the first 42 bytes of AFDX frames as parameters. For example a Frame Header Parameter can get values like the Virtual Link or the Addressing of frames. Note that the intention of Frame Header Parameters is to retrieve information of fields arranged in front of the payload. Therefore these parameters are not controllable as they would make frames to transmit inconsistent. However a Frame Header Parameter can act as any other Parameters if it is located fully within the payload, meaning that the Byte Position property is >= 42 and the parameter does not point into the Frame Header because of its BitPos / BitLength property. Because of the direct dependency to the nodes above

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

each Frame Header Parameter AFDX Message / Virtual Link it is recommended that these nodes are wild carded for the corresponding property. Otherwise the parameter will not be displayed on the resource side. For example a Frame Header Parameter representing the Virtual Link of a AFDX Frame is defined as BytePos = 5 and BitLength = 16 (BitPos = 0) therefore the Virtual Link should be wild carded to retrieve all VL Frame Header Parameters.



The Frame Header Parameter , FH_VL represents the VL of a generic frame.

Figure 66: AFDX – Frame Header Parameters

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

The most common Frame Header Parameters are collected in the following table.

Parameter Representation	BytePos	BitPos	BitLength
VL	5	0	16
IP Source Address	29	0	32
IP Destination Address	33	0	32
UDP Source Port	35	0	16
UDP Destination Port	37	0	16

Table 1: Frame Header Parameters examples

The typical use case for Frame Header Parameters is designed for the chronological capture and the generic send list.

2.3.7 Parameters in the chronological capture mode

If the ports are opened and the parameters are properly setup within the database, the chronological capture mode will offer to display parameters associated with a particular captured AFDX message. The same hierarchy as in the parameter database is used here as well.

Selecting a message in the chronological bus monitor opens the detailed message view. Beside the message's Meta information a 'Parameters' node is displayed in the 'Message Information' window.

Expanding that node displays the list off all parameters associated with the current message.

Selection of a single parameter indicates its location in the raw data of the message by highlighting the associated data word(s) or parts of it in red. Any defined aliases will also be displayed in the 'Functional Data Set' (FDS) node or in the 'Global Parameters' node:

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

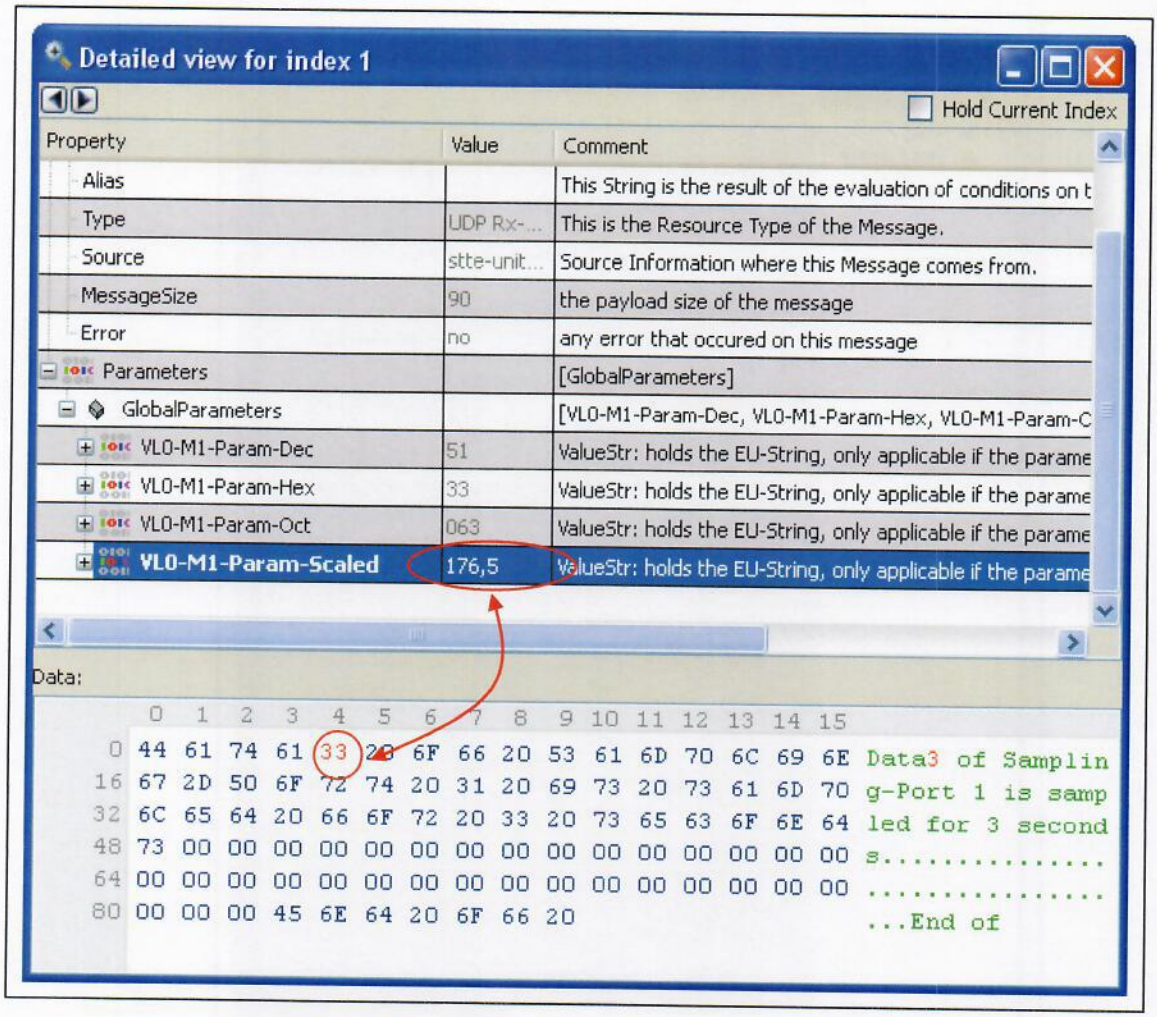


Figure 67: AFDX - Parameters in detailed view of Index View

2.4 ARINC825/CAN Parameters

2.4.1 Stream creation

CAN Streams are created via the corresponding 'CAN' node of the Parameter component tree. Right click on the node and select 'Create CAN Stream' from the context menu.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

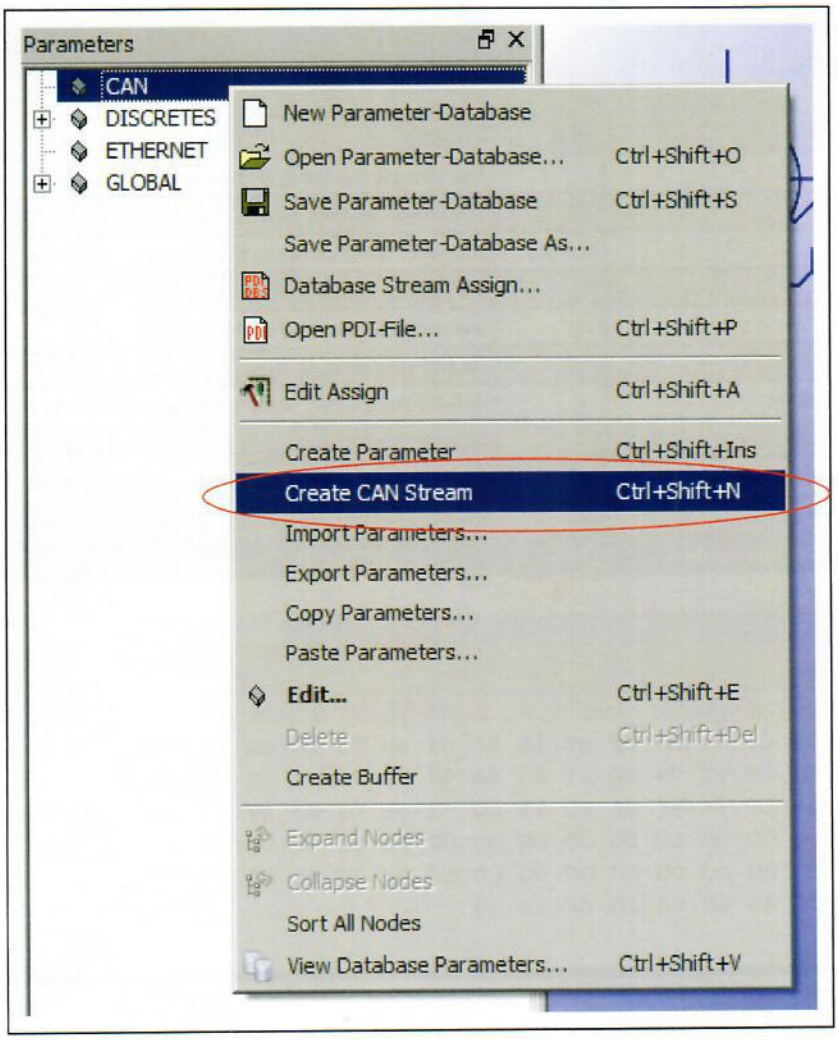


Figure 68: CAN - Stream creation

The 'Create CAN Stream' command creates a new CAN stream node for definition of CAN ID's and Parameters. The properties of a new stream node are the 'Alias' and the 'StreamId'. The 'StreamId' property defines the association with the corresponding resource in the way that it has to match with the 'DbsStreamId' property of the CAN Resource / CAN-Network.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

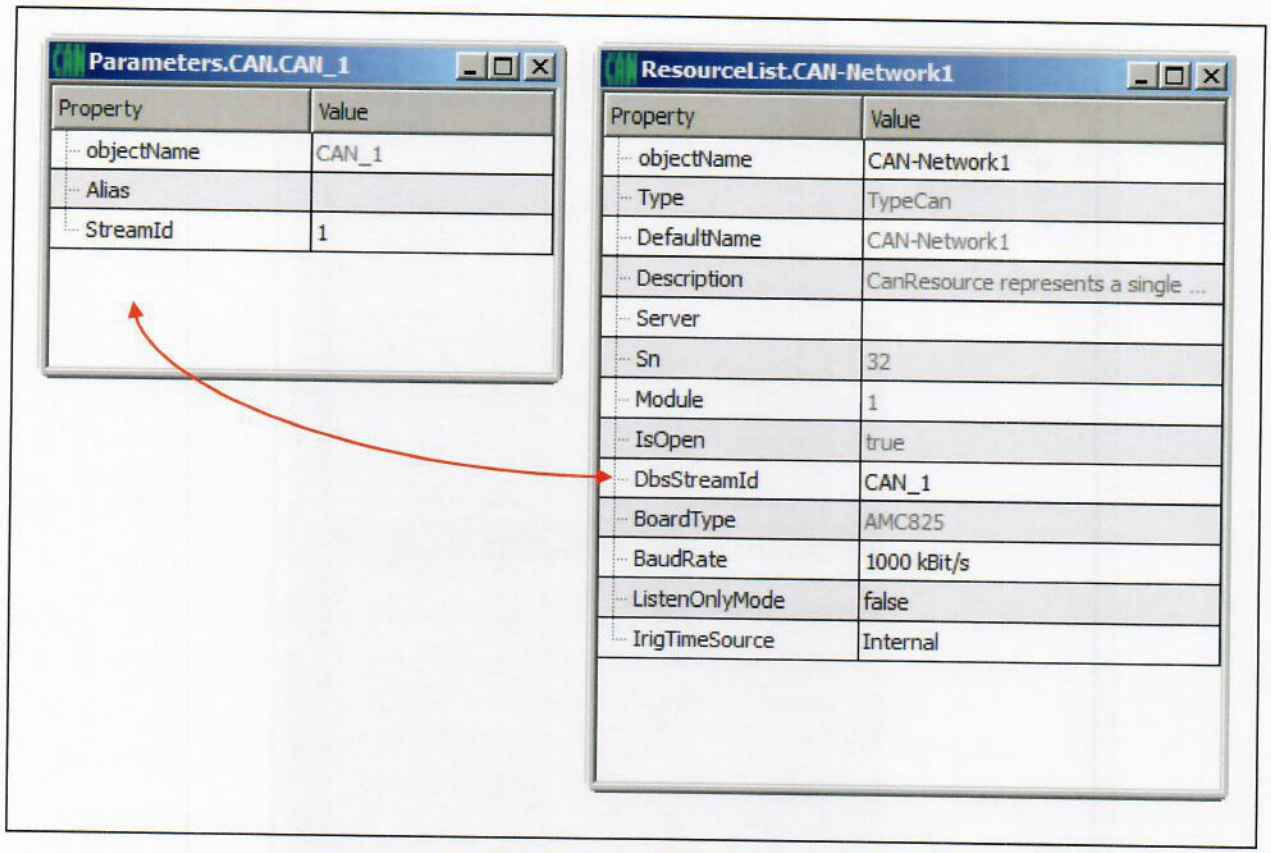


Figure 69: CAN - Database to Resource association

2.4.2 ID creation

Selection of a stream node (e.g. CAN_1) and right mouse click offers a context menu for further operations.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

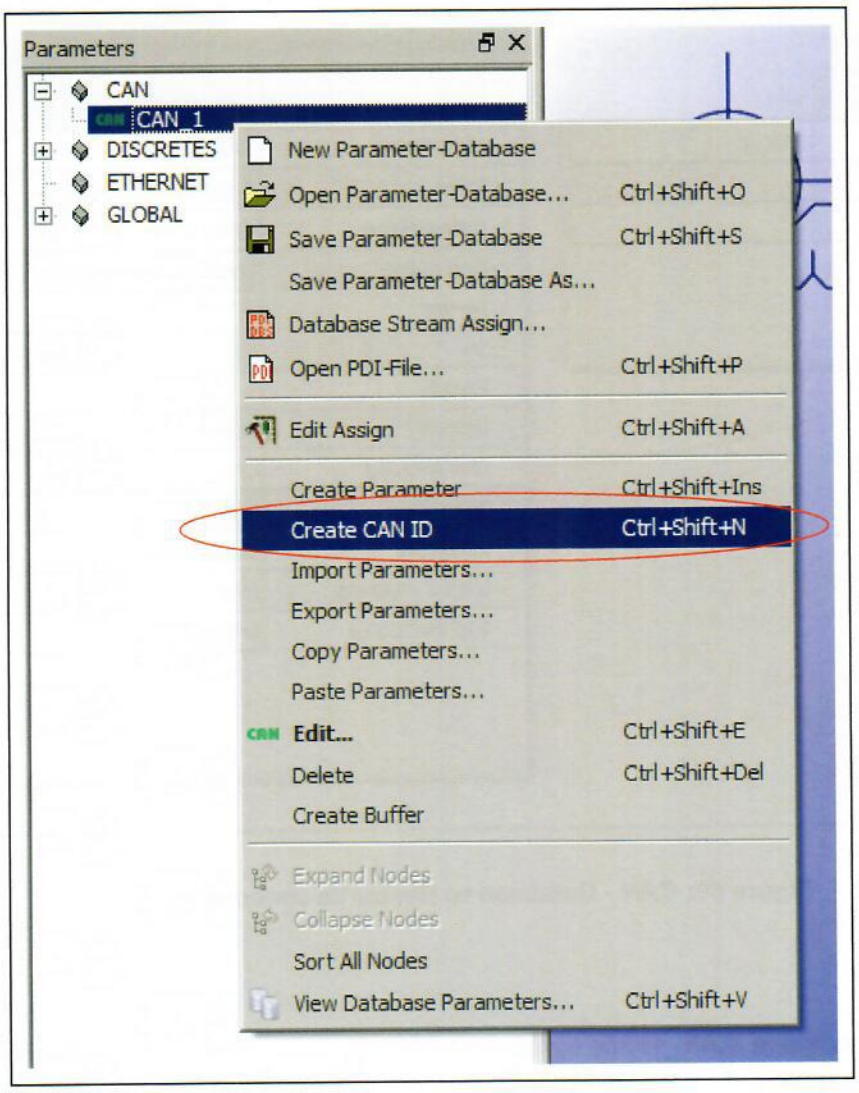


Figure 70: CAN - ID creation

'Create CAN ID' creates per default a new CAN ID sub node for definition of parameters which are associated with this section of a CAN-Network resource PBA.pro component.

A double click on the 'CAN_E_x' node opens the property dialog to configure the CAN ID alias ('Alias' property), used for this CAN ID. The ID can be changed via the 'ID' property. Furthermore, the 'Flags' property allows to set CAN specific flags for the use of ExtId (Extended Identifier) or RTR (Remote Transmission Bit).

Note that the Parameter definitions for a CAN ID are applicable for the RX (Capture) and TX (Transmit) side - if the corresponding ID is defined.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

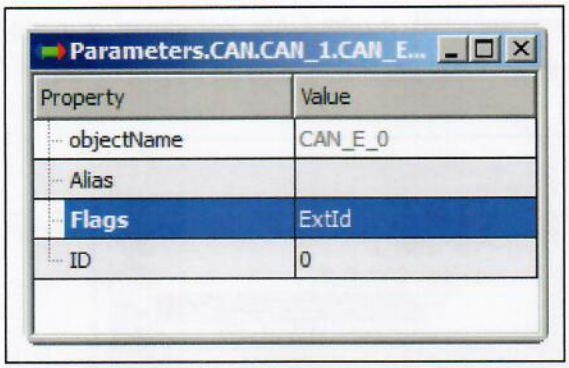


Figure 71: CAN - ID node configuration

2.4.3 Parameter creation

Selection of a CAN ID node and right mouse click offers a context menu for creation of CAN Parameters associated with the selected CAN ID.

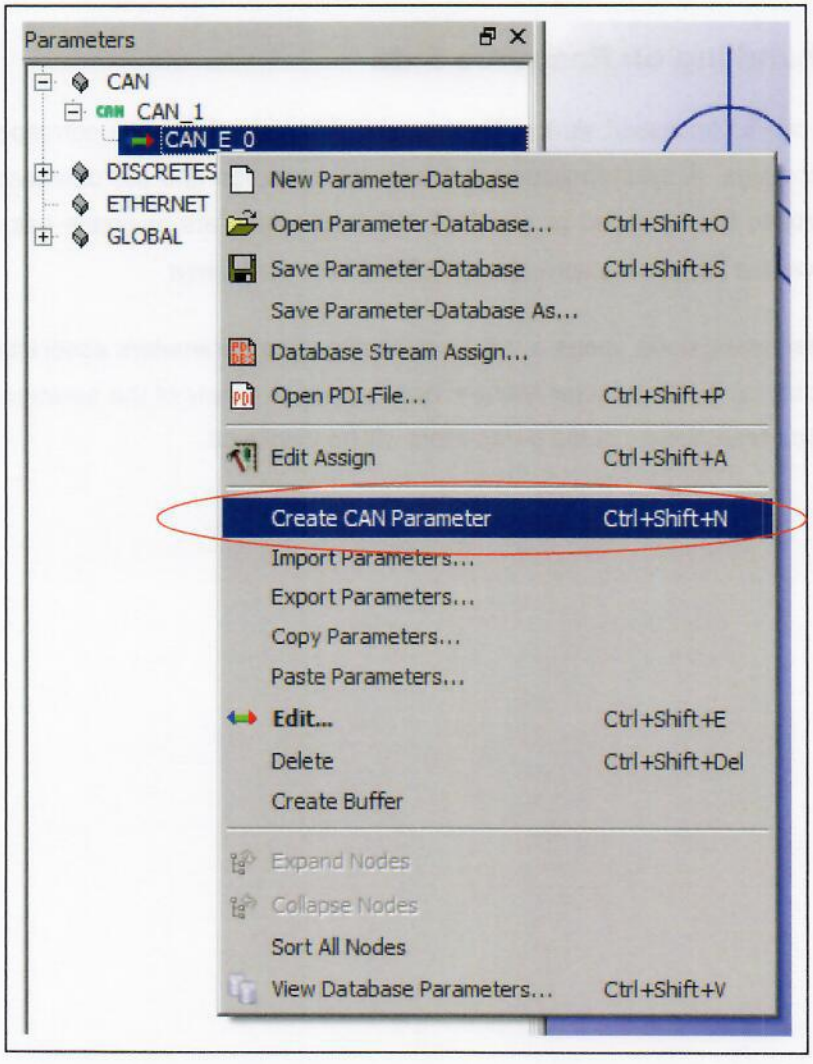


Figure 72: CAN - Parameter creation

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

If 'Create CAN Parameter' has been selected, a new CAN 'Parameter_x' node below the corresponding CAN node is created and appears in the CAN parameter tree as shown below.

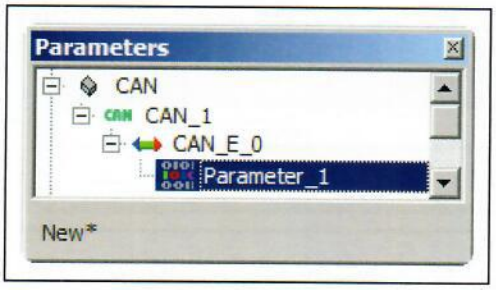


Figure 73: CAN - DBS Parameter node

A double click on the Parameter node offers modification of the basic parameter properties. See chapter 3, Edit Parameter Formats of this document for further details on editing Parameters.

2.4.4 Parameter handling on Resource Side

Associated parameters can be accessed via the 'Parameters' sub node below the corresponding RX and TX CAN ID's on resource side. A synchronisation between the resource and the database is performed automatically if the database has changed or modified. RX Parameters are available below the Capture Activity if a CAN ID is received for which a corresponding Parameter is defined.

A double click on a 'Parameters' node opens a view and displays the parameters associated with the selected node. A double click on a '<Parameter Name>' node opens the view of the selected parameter. In both cases the on-line data resp. values of the parameters will be displayed.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

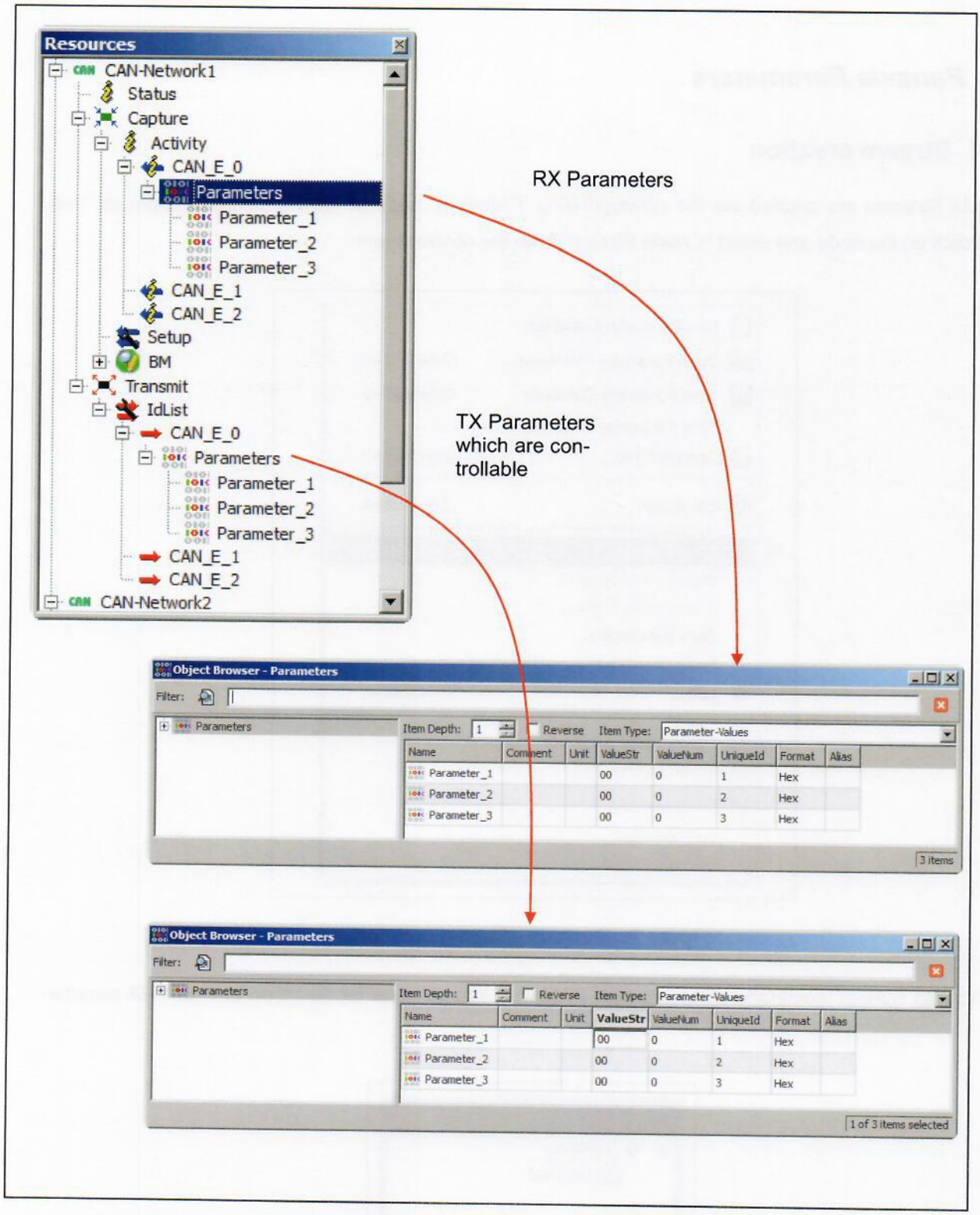


Figure 74: CAN - Online Parameter displays

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

2.5 Panavia Parameters

2.5.1 Stream creation

Panavia Streams are created via the corresponding 'PANAVIA' node of the Parameter component tree. Right click on the node and select "Create Stream" from the context menu.

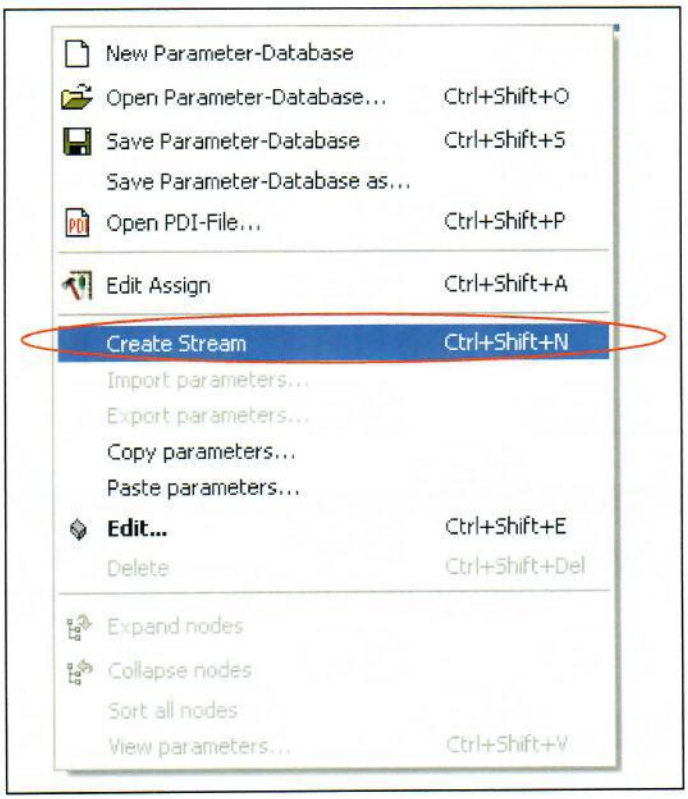


Figure 75: Panavia - Stream creation

The 'Create Stream' command creates a new PANAVIA stream node for definition of PANAVIA parameters.

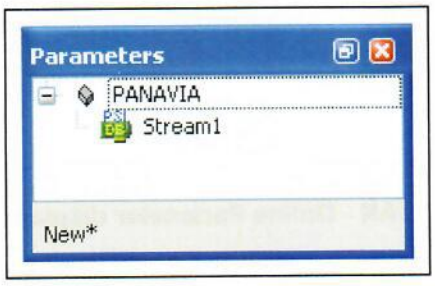


Figure 76: Panavia - DBS Stream Node

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

The properties of a new stream node are the 'Alias' and the 'StreamId'. The 'StreamId' property defines the association with the corresponding resource in the way that it has to match with the 'Module' property of the PANAVIA Resource (see below).

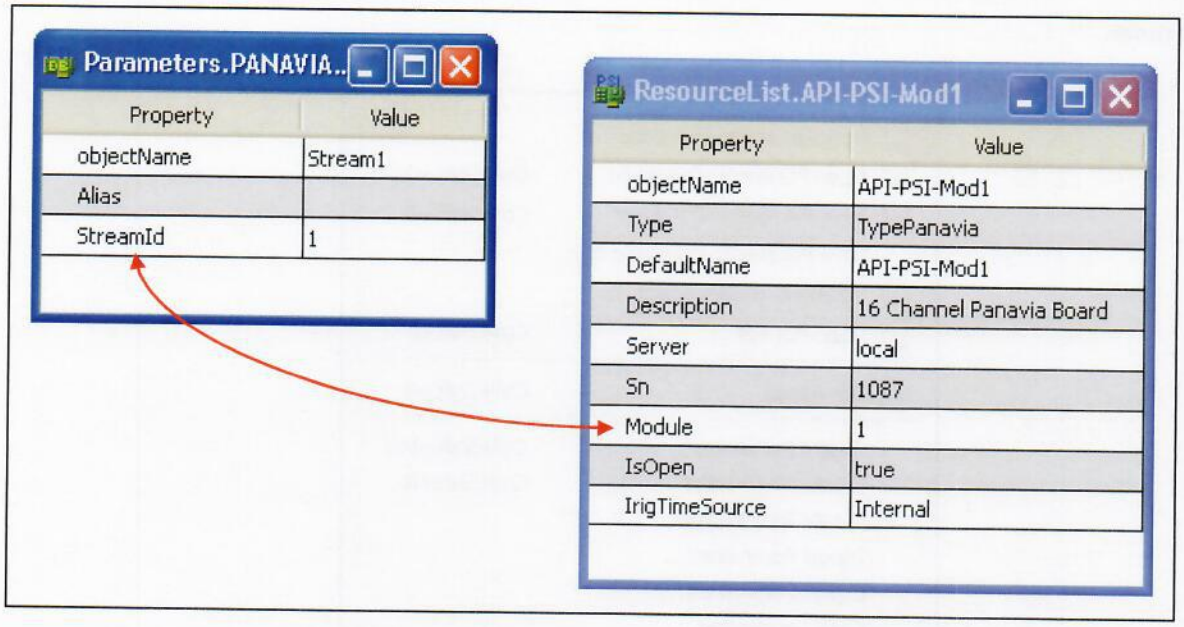


Figure 77: Panavia - Database to Resource association

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

2.5.2 Sender/Receiver node creation

Selection of a stream node (e.g. Stream1) and right mouse click offers a context menu for further standard operations for the database and the stream as well as two menu entries to create a send and a receive direction node.

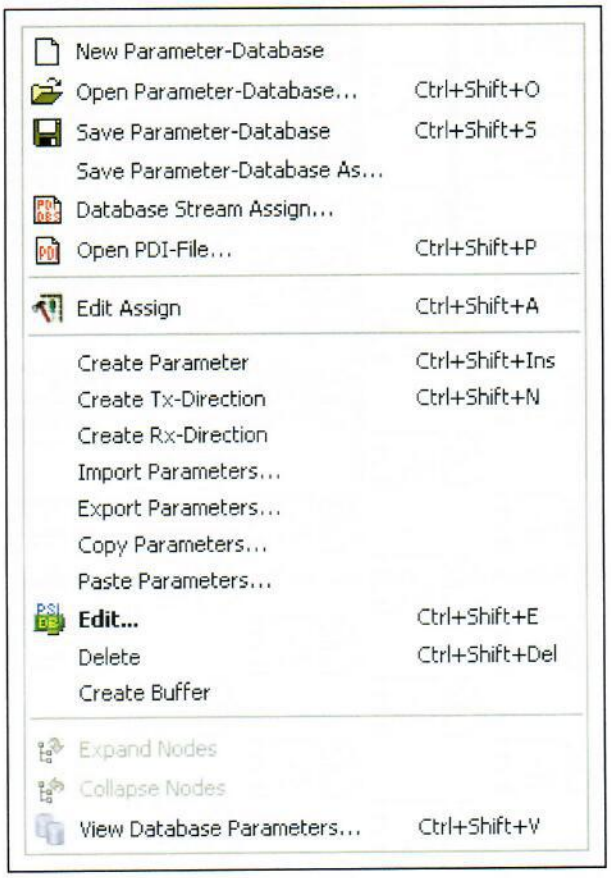


Figure 78: Panavia - Sender resp. receiver node creation

While the entry 'Create Tx-Direction ' creates the Send direction root node to define send parameters, the 'Create Rx-Direction' entry is used to create a Receive direction root node to define receive parameters.

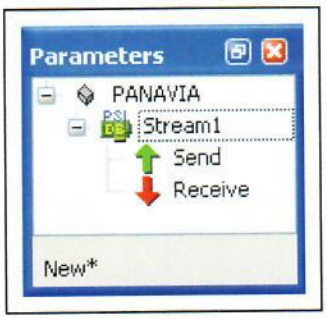


Figure 79: Panavia - DBS Send and Receive Nodes

A double click either on the Send or on the Receive node opens the property dialog for the direction nodes. If possible, the direction can be switched via the Direction property even after the creation.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

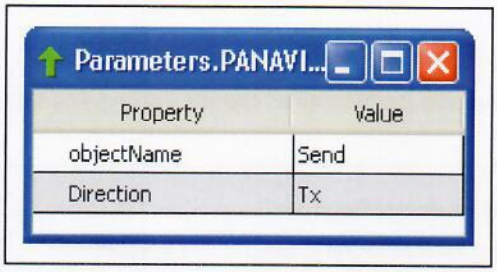


Figure 80: Panavia - Send node configuration

2.5.3 Channel creation

A right click on a direction node (Sender or Receiver) pops up the context menu of the receiver, where at maximum eight channel nodes can be created for each direction. A new channel is created via the menu entry “Create Channel”.

These channels are associated with the available channels below the corresponding direction, displayed within the PBA.pro PANAVIA Resource component tree. Therefore the "ChannelNr" property is used to indicate to which channel the parameter node belongs to.

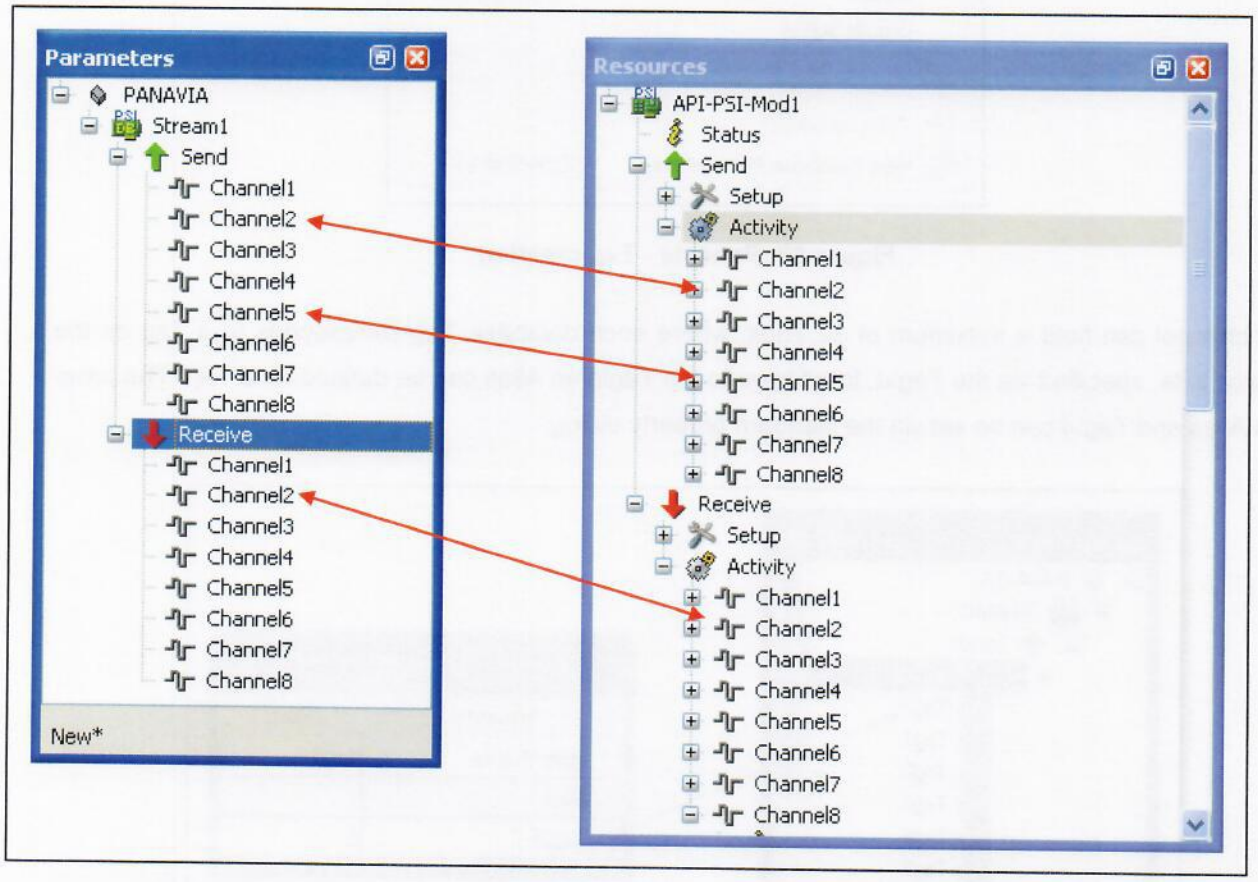


Figure 81: Panavia - DBS Channel to Channel instance association

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

2.5.4 Tag creation

The Context menu of the channel offers the possibility to create Tag nodes for a channel.

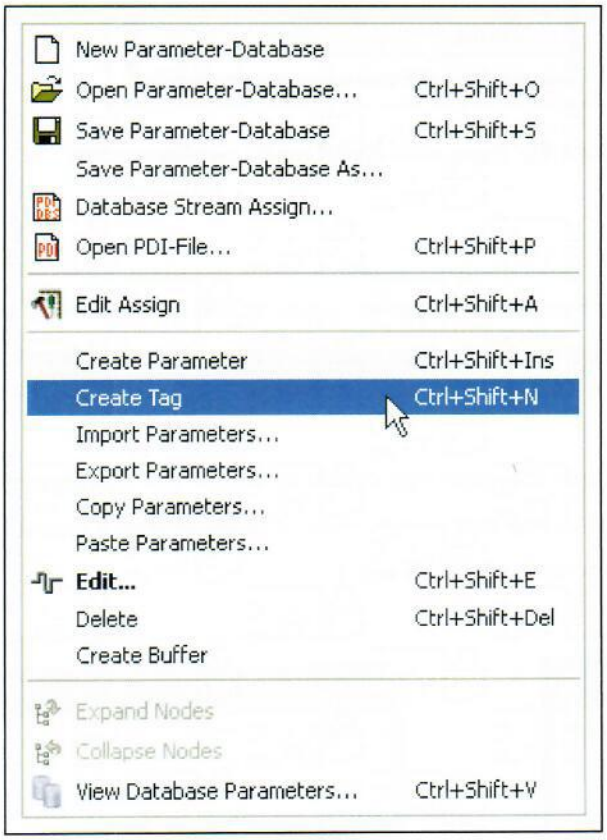


Figure 82: Panavia - Tag creation

Each channel can hold a maximum of 32 Tags, where each database Tag corresponds to a Tag on the resource side, specified via the TagId. In addition to the TagId an Alias can be defined for a Tag. The properties Alias and TagId can be set via the standard property dialog.

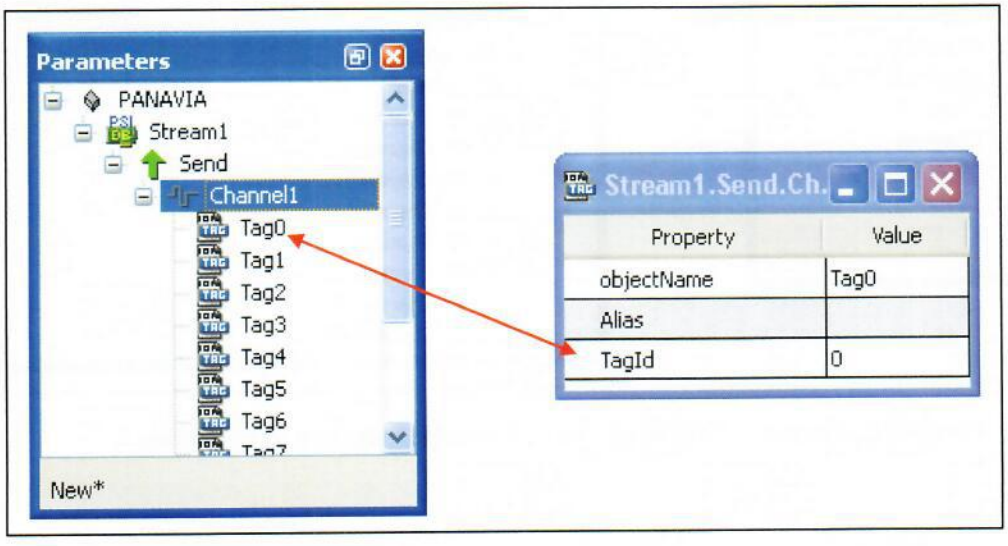


Figure 83: Panavia - Tag configuration

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

2.5.5 Parameter creation

Selection of a Tag node and right mouse click offers a context menu for creation of PANAVIA Parameters associated with the selected PANAVIA Tag.

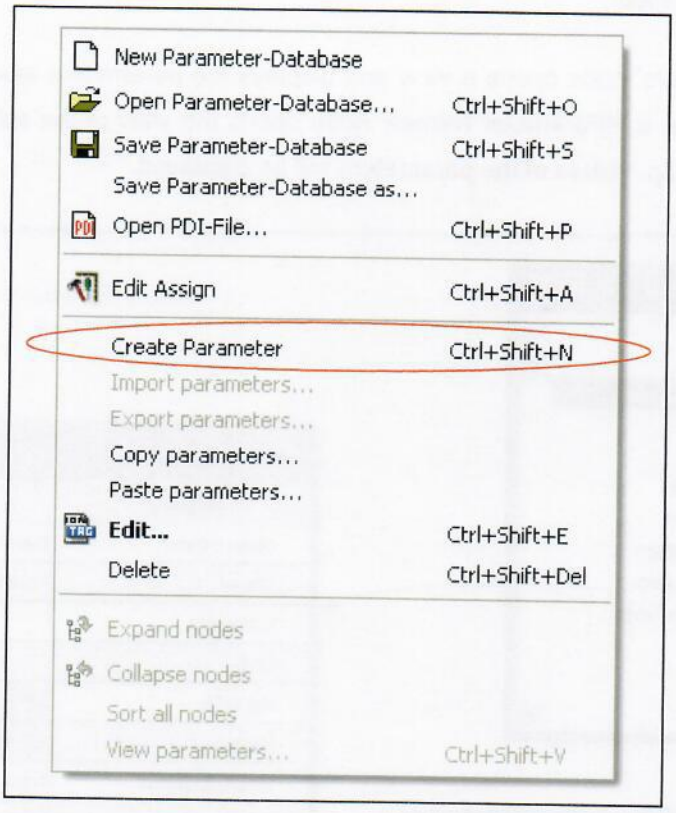


Figure 84: Panavia - Parameter creation

If 'Create Parameters' has been selected, a new PANAVIA 'Parameter_x' node below the corresponding Tag node is created and appears in the PANAVIA parameter tree as shown below.

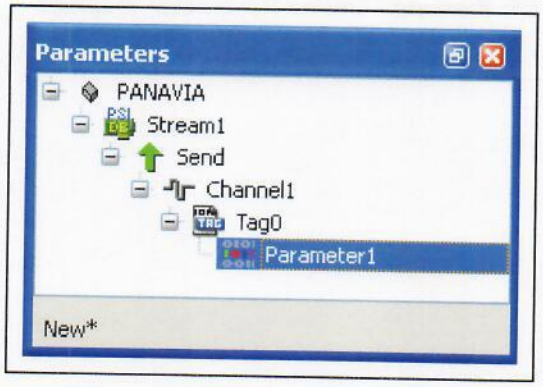


Figure 85: Panavia DBS Parameter node

A double click on the Parameter node offers modification of the basic parameter properties. See chapter 3, Edit Parameter Formats of this document for further details on editing Parameters.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

2.5.6 Parameter handling on Resource Side

Associated parameters can be accessed via the 'Parameters' sub node below the corresponding Tags on resource side. A synchronisation between the resource and the database is performed automatically if the database has changed or modified.

A double click on a 'Parameters' node opens a view and displays the parameters associated with the selected node. A double click on a '<Parameter Name>' node opens the view of the selected parameter. In both cases the on-line data resp. values of the parameters will be displayed.

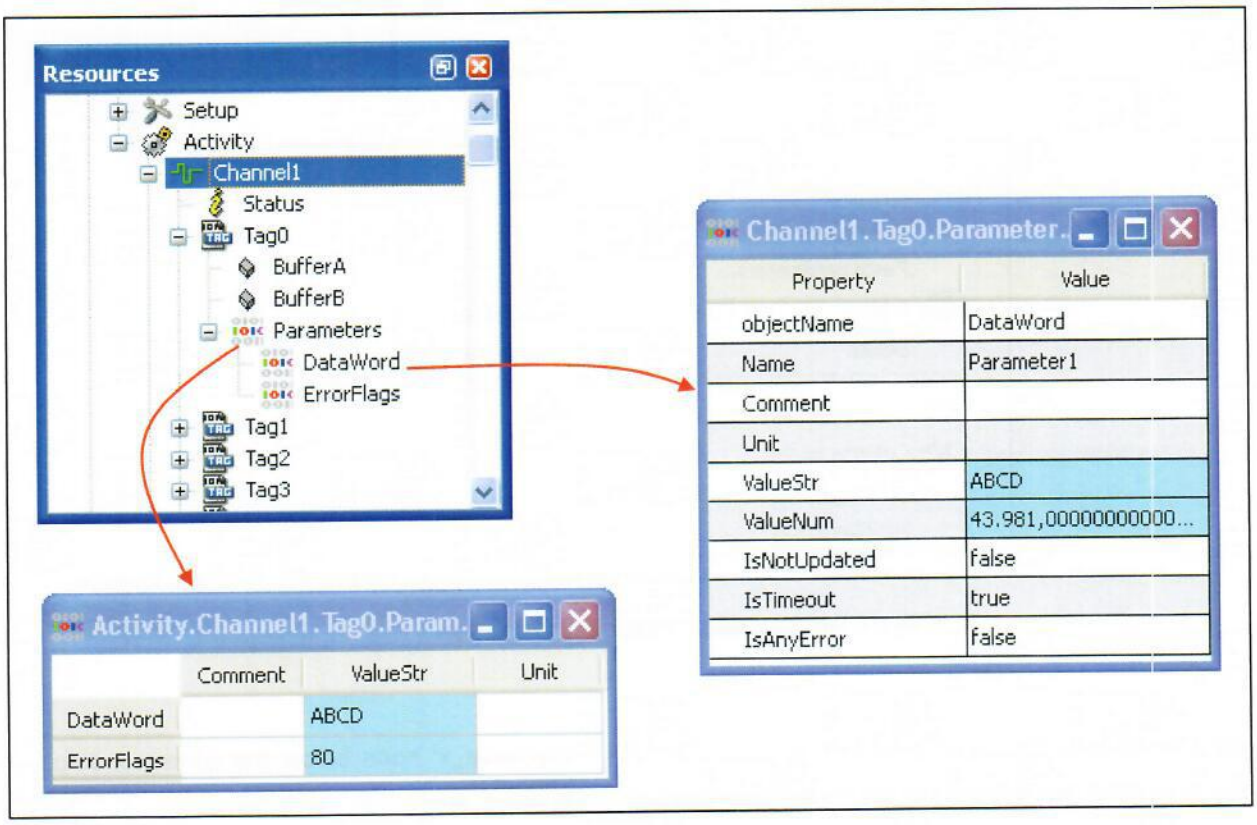


Figure 86: Panavia - Online Parameter displays

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

2.6 Discrete and Analogue Parameters

This chapter will explain how analogue and discrete Parameters are to be setup and used in the PBA.pro. As both Types of Parameters are setup and configured in the same way, (despite the Names of the corresponding context menu entries to choose) the following chapter will deal only with the discrete resource as reference.

2.6.1 Stream creation

The Discrete Stream related parameters are created via right click on the 'DISCRETES' node of the Database tree.

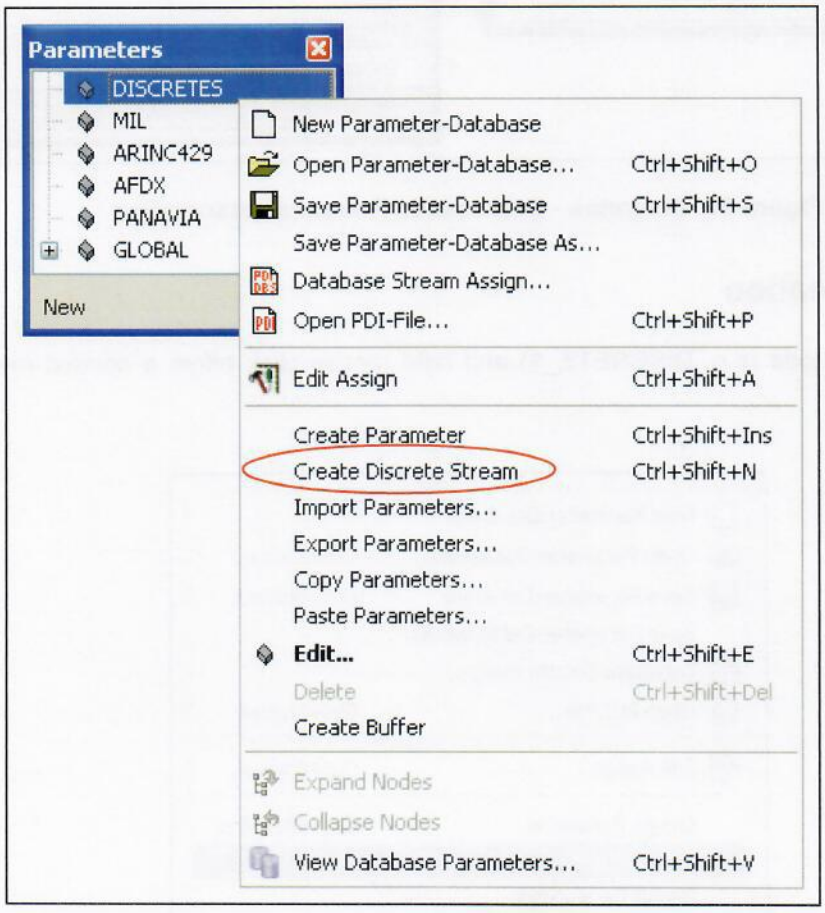


Figure 87: Discretes - Stream creation

The 'Create Discrete Stream' command creates a new discrete stream node for definition of discrete parameters.



Figure 88: Discrete - DBS Stream node

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

The main properties of a new stream mode are the 'Alias' and the 'StreamId'. The 'StreamId' property defines the association with the corresponding resource in the way that it has to match to the 'Module' property of the Resource (see below).

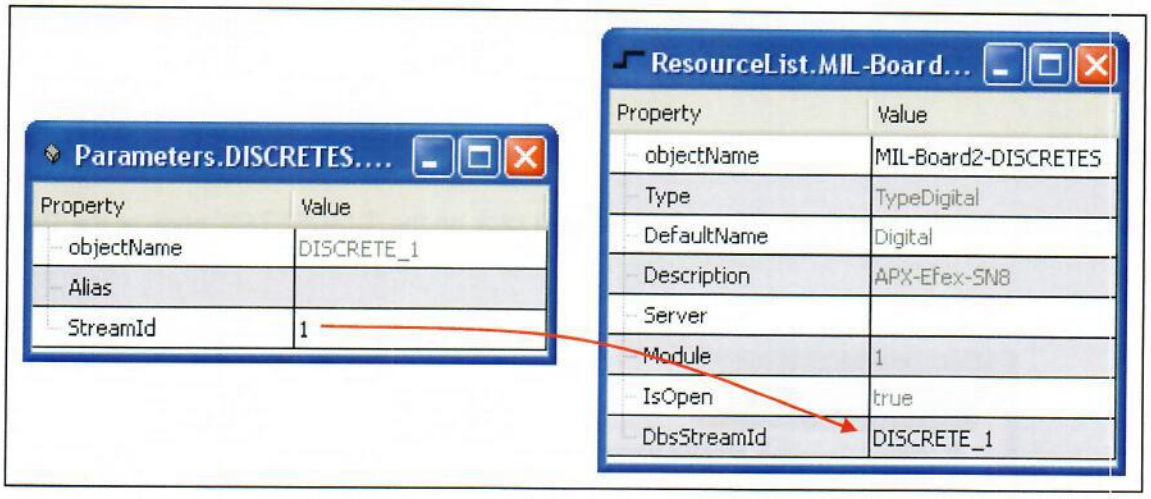


Figure 89: Discretes - Database to Resource association

2.6.2 Channel creation

Selection of a stream node (e.g. DISCRETE_1) and right mouse click offers a context menu for further operations.

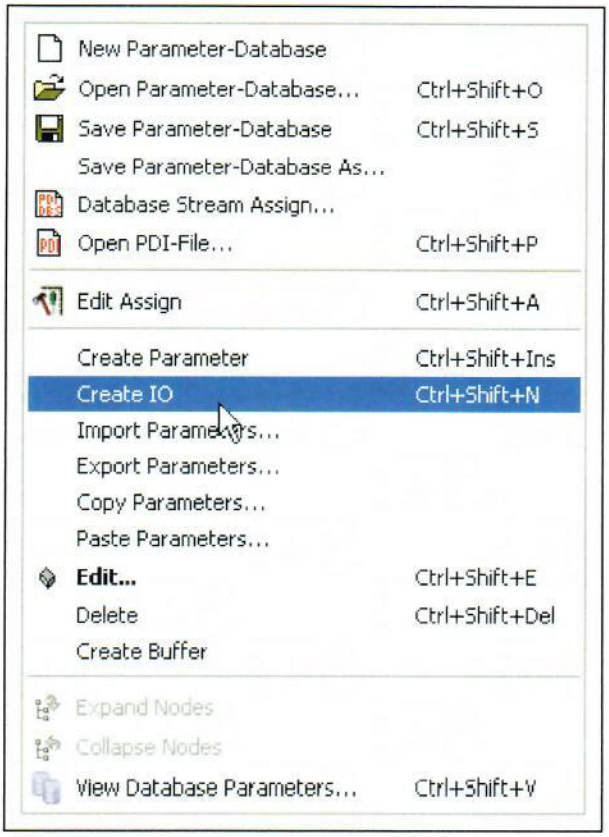


Figure 90: Discretes - Channel creation

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

'Create IO' creates per default a new 'Input' channel sub node for definition of parameters which are associated with a Discrete Input Channel section of a discrete resource PBA.pro component.



Figure 91: Discretes - Channel DBS node

A double click on the 'Input' node opens the property dialog which supports configuration of the channel alias ('Alias' property), used for this channel. Furthermore, the channel type can be changed via the 'Type' property to

- Input (fixed discrete inputs lines)
- Output (fixed discrete outputs lines)
- IO (programmable discrete lines)

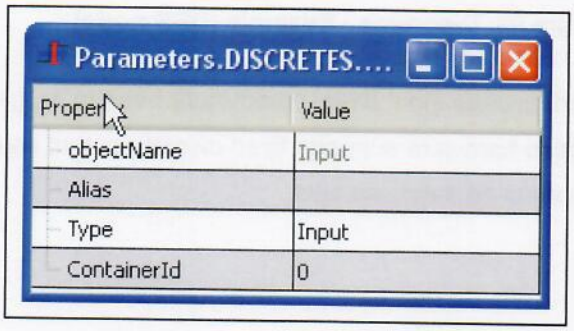


Figure 92: Discretes - Channel configuration

Further 'Create IO' commands will create one of the still not used channel types ('Input', 'Output', 'IO'). If more than one node for each channel type is to be configured, the property ContainerID is to be used to distinguish between the different containers of the same type. The channel types are associated with the available channels of the PBA.pro Discrete Resource component, which are dependent on the used discrete hardware. The component will also offer corresponding 'channel' nodes. E.g. discrete channels on AIM APM1553 PC-Card Interface Module:

Example: *PBA.pro Discrete Resource for APM1553 Discrete*

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

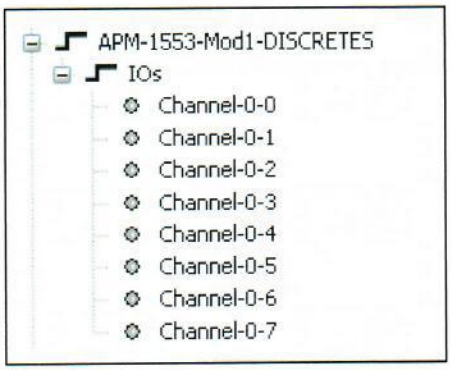


Figure 93: Discretes - Example (Resource node)

This hardware supports programmable discrete I/O lines which is reflected by the 'IOs' node. Hence, signals form the database, which should be associated with these lines, need to be defined below the 'IO' node (see below).



Figure 94: Discretes - Example (DBS node)

If the used hardware supports fixed discrete input lines, parameters need to be generated below an 'Input' channel on database side. If the used hardware supports fixed discrete output lines, parameters require to be generated below an 'Output' channel on database side.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

2.6.3 Parameter creation

Selection of an 'Input', 'Output' or 'IO' node and right mouse click offers a context menu to create Discrete Parameters associated with the selected Discrete Channel.

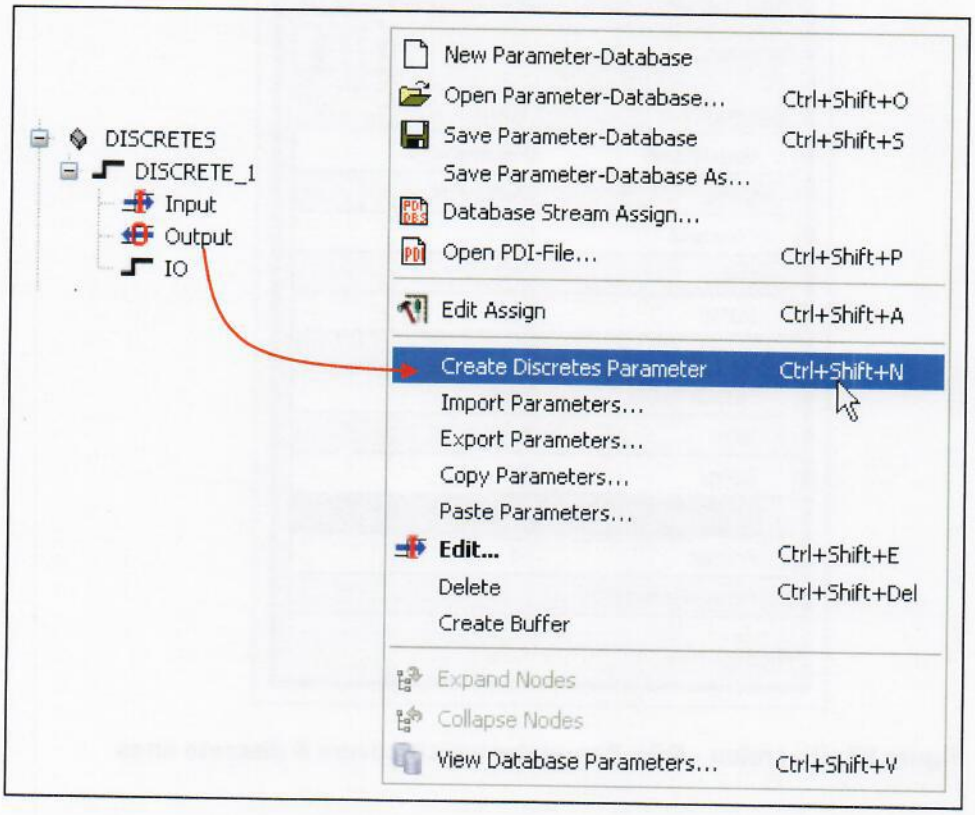


Figure 95: Discretes - Parameter creation

The selection of the 'Create Discretes Parameter' creates a new 'Parameter_x' node below the corresponding channel node representing a Discrete Parameter.

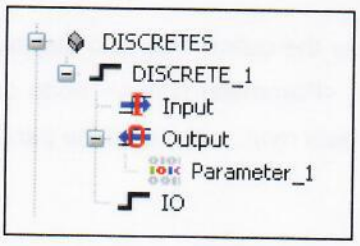


Figure 96: Discretes - DBS Parameter node

A double click on the Parameter node offers modification of the basic parameter properties. See chapter 3, Edit Parameter Formats of this document for further details on editing Parameters.

Note that a Discrete Parameter can also cover more than a single discrete line e.g. for using numeric formats (e.g. encoding a 1553 RT address using 5 discrete channels) or enumerated format. To do so, the 'BitLength' property of the parameter needs to be set accordingly (see picture below).

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

The 'BitLength' parameter cannot exceed the maximum number of available discrete lines of the corresponding channel.

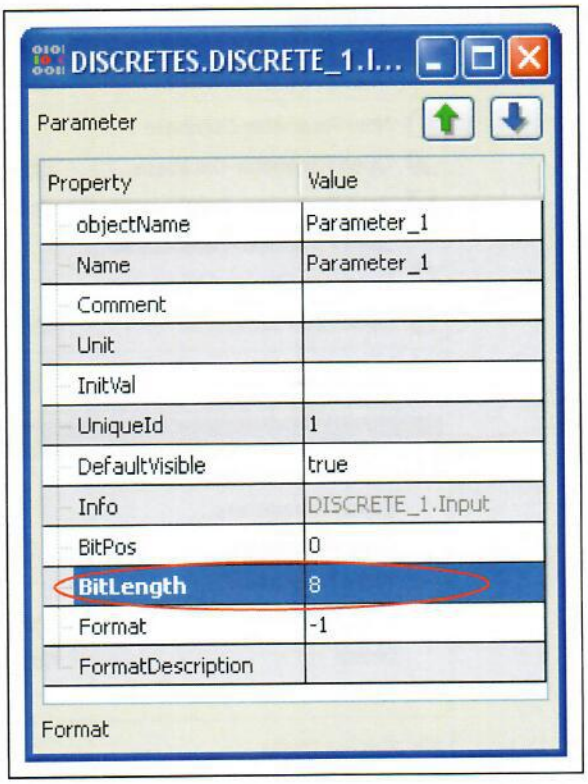


Figure 97: Discretes - DBS Parameter which covers 8 discrete lines

2.6.4 Parameter handling on Resource Side

Associated parameters can be accessed via the 'Parameters' sub node of a Discrete Channel sub node (e.g. IO's) . A synchronisation between the resource and the database is performed automatically as soon as the database has been changed or modified.

A double click on a 'Parameters' node opens the online view and displays a list of parameters associated with the selected node. A double click on a '<Parameter Name>' node opens the online display of the selected parameter. In both cases the on-line data resp. value(s) of the parameter(s) are displayed.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

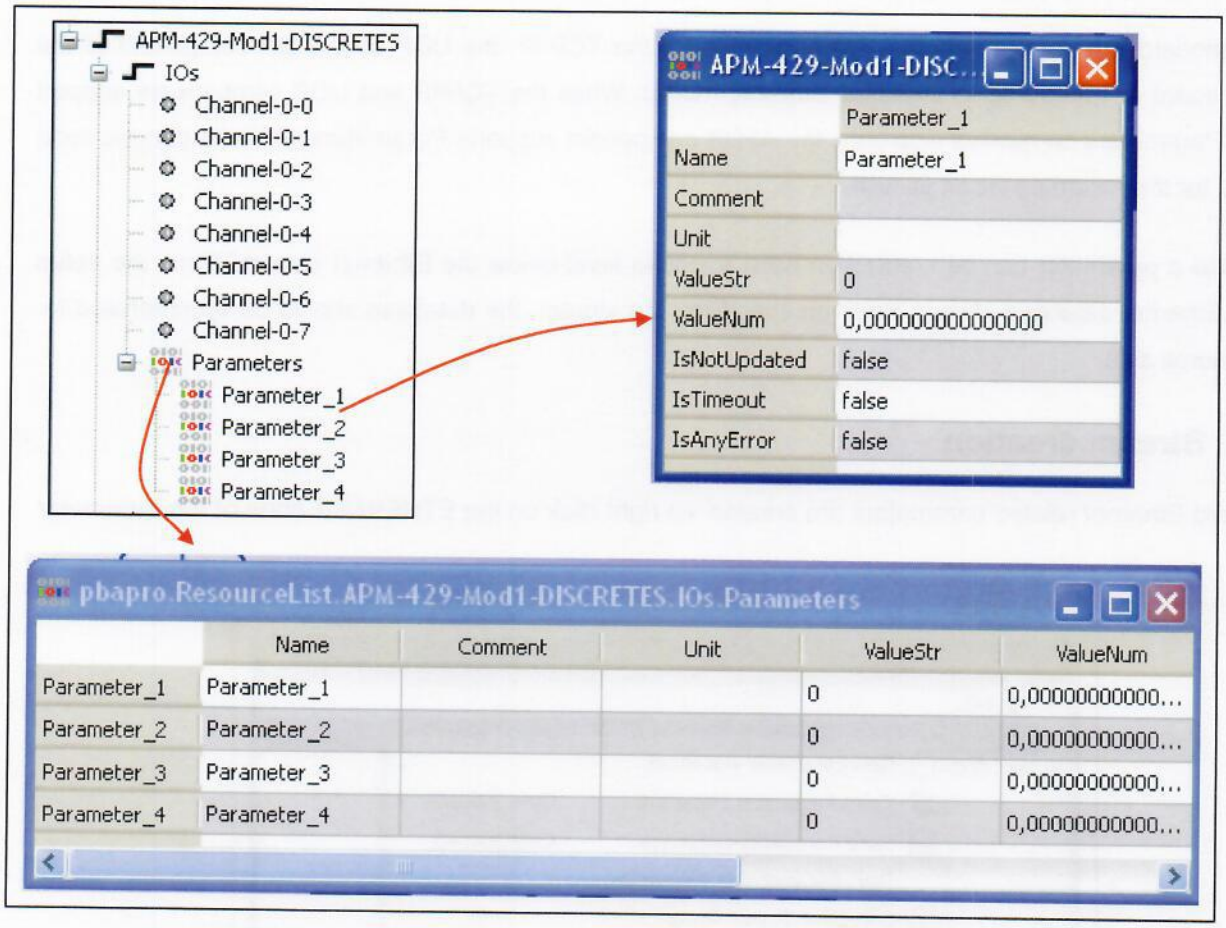


Figure 98: Discretes - Parameters on resource side

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

2.7 Standard Ethernet Parameters

The Standard Ethernet parameters are applicable for the TCP/IP, the UDP and AFDX component (if the AFDX board is configured to standard Ethernet mode). While the TCP/IP and UDP components support online Parameters on receive side only, the AFDX component supports Parameters within its generic send list and for the recording views as well.

Note that a parameter can be created on each structure level below the Ethernet stream. Thus, the setup for an Ethernet database always starts on specifying the stream, the database should be synchronised to, on resource side.

2.7.1 Stream creation

Standard Ethernet related parameters are created via right click on the ETHERNET node of the Parameter tree.

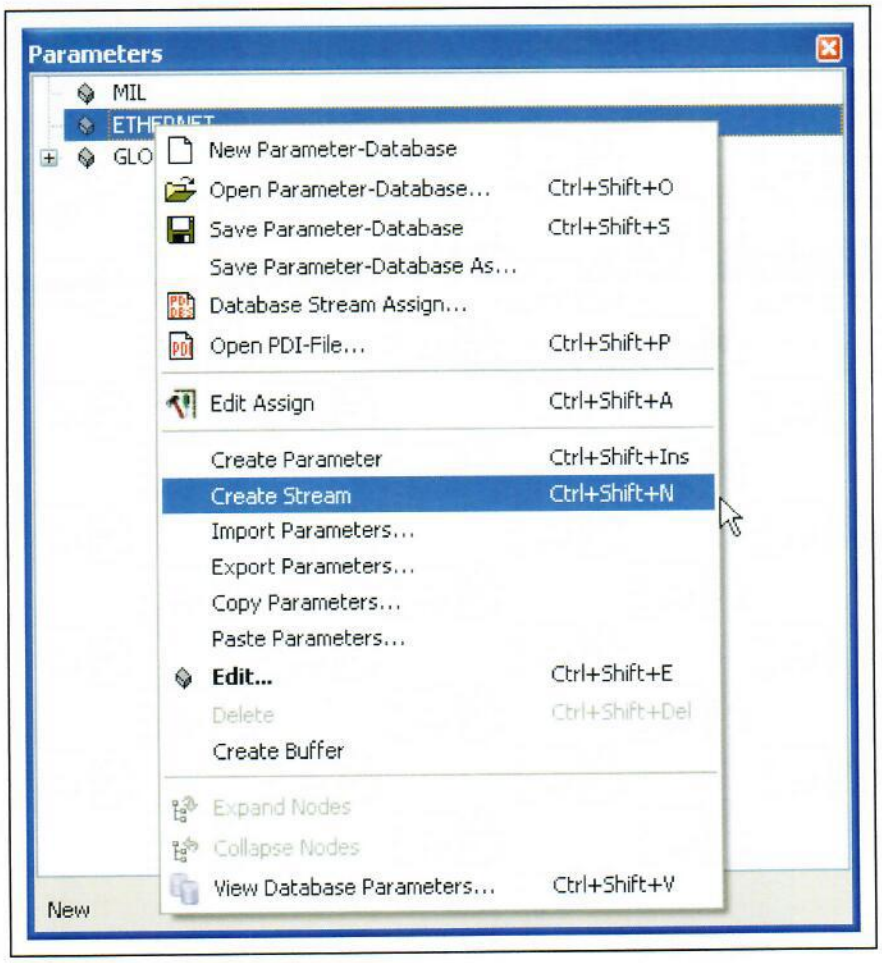


Figure 99: Standard Ethernet - Stream creation

A right click on the ETHERNET node in the parameter Tab pops up the Ethernet node's context menu, where the 'Create Stream' command creates a new stream node for definition of Ethernet parameters.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com



Figure 100: Standard Ethernet - DBS Stream node

The properties of a new stream node are the 'Alias' and the 'StreamId'.

The 'StreamId' property defines the association with the corresponding Ethernet port of a TCP/IP, UDP or AFDX-Ethernet port. Thus, this property resp. the value of the property has to match to the 'StreamId' or 'DbsStreamId' property of the resource port.

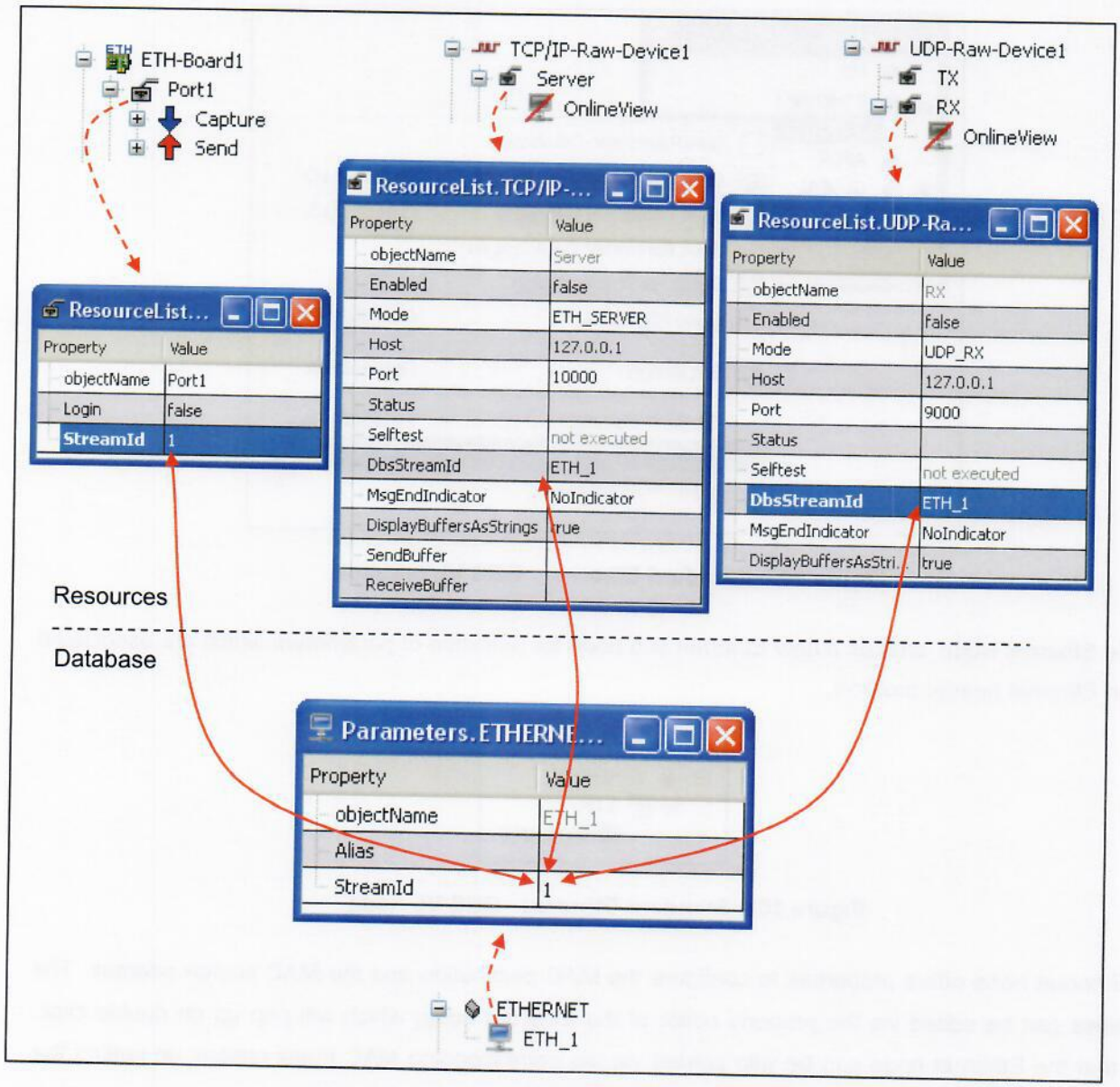


Figure 101: Standard Ethernet – Database to Resource association

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

2.7.2 AFDX Ethernet specific nodes

As the AFDX component (running in standard Ethernet mode) provides functionality to modify the whole Ethernet frame (Header and Payload), the standard Ethernet component provides functionality to setup the database according to some standard Ethernet headers like MAC, IP and UDP. Hence, Byte 0 will point to the first Byte of the Frame's Header.

2.7.2.1 Ethernet Header Creation

Selection of a stream node (e.g. ETH_1) and right mouse click offers a context popup menu for further operations.

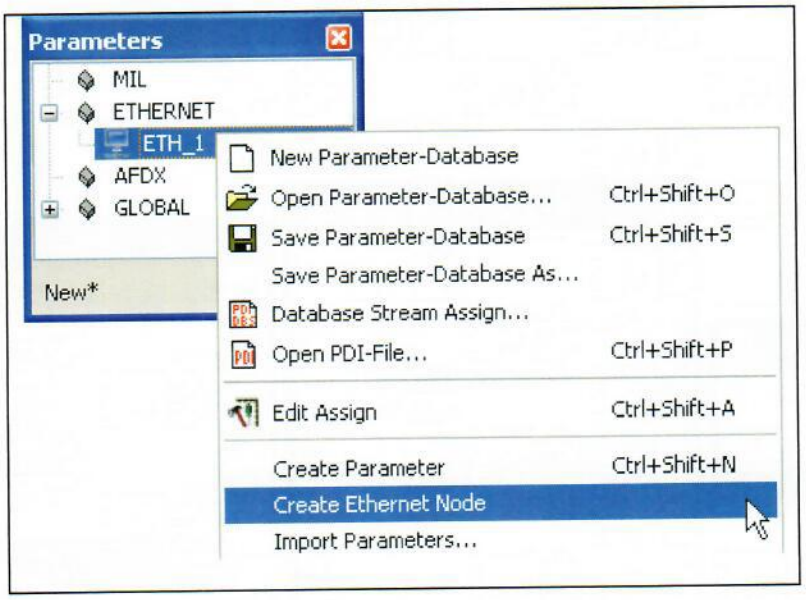


Figure 102: Standard Ethernet - DBS VL creation

'Create Ethernet Node' creates a new Ethernet sub node for definition of parameters which are associated with an Ethernet header protocol.



Figure 103: Standard Ethernet - DBS VL node

The Ethernet node offers properties to configure the MAC destination and the MAC source address. The properties can be edited via the property editor of the Ethernet node, which will pop up on double click. Note that the Ethernet node can be wild carded via the corresponding MAC mask options on setting the UseMacMasks property to true.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

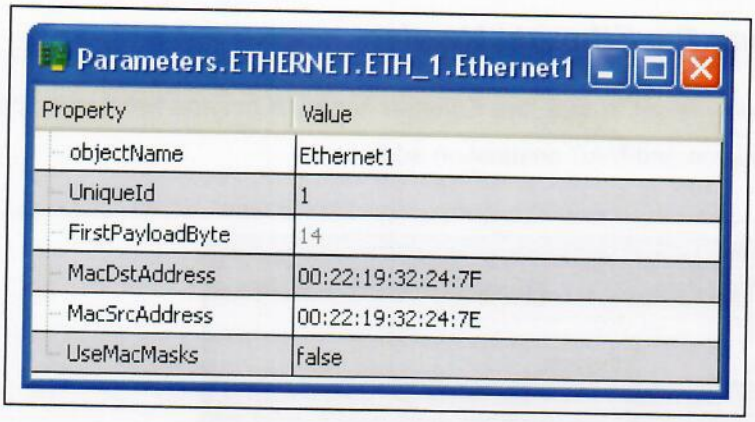


Figure 104: Standard Ethernet - DBS VL configuration

- **objectName:**

  The name of the Ethernet node

- **UniqueId:**

  A unique Ethernet node identifier

- **FirstPayloadByte:**

  This is a read only property. This property displays the first byte, which does not belong to the header anymore

- **MacDstAddress:**

  The MAC destination address

- **MacSrcAddress:**

  The MAC source address

- **UseMacMasks:**

  This property indicates if MAC masking features are used or not. See the wildcard section below for further details

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

2.7.2.1.1 Wildcarding the Ethernet Header

If the UseMacMasks property is set to true, the Ethernet node will provide two further properties to set-up wildcarding for the MAC source and MAC destination addresses.

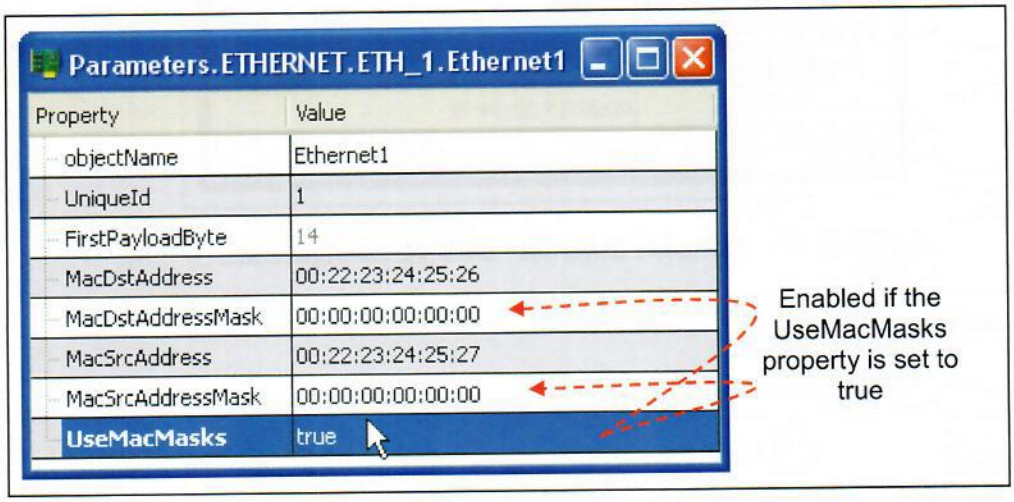


Figure 105: Standard Ethernet - DBS MAC Wildcarding

Wildcarding of a MAC is comparable to sub netting in Ethernet networks. Like a MAC address, the MAC mask contains six bytes (48 bits) and is often written using the same "double-dotted-hexadecimal" notation. For example, a MAC mask in binary representation 00000000 00100010 00100011 00100100 00100101 00100110 is typically shown in the equivalent, more readable form 00:22:23:24:24:26. All bits of the Mask, set to 0 are ignored resp. accepted. All bits set to 1 have to match the MAC definition exactly.

Example:

All MAC source addresses from 00:22:23:24:24:00 *to* 00:22:23:24:24:FF *should be accepted. Thus, the following configuration is to be set up:*

- *UseMacMasks property is to be set to true*
- *the MacSrcAddress property is set to* 00:22:23:24:24:xx, *where xx can be any value*
- *the MacSrcAddressMask property has to be set to FF:FF:FF:FF:FF:00*

Note: As all values of the last Mac Byte will be accepted the MacSrcAddress property can be set to any of the accepted MAC addresses.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

2.7.2.2 IP Header creation

Selection of an Ethernet node and right mouse click offers a context menu for creation of an IP Header node, related to the Ethernet header.

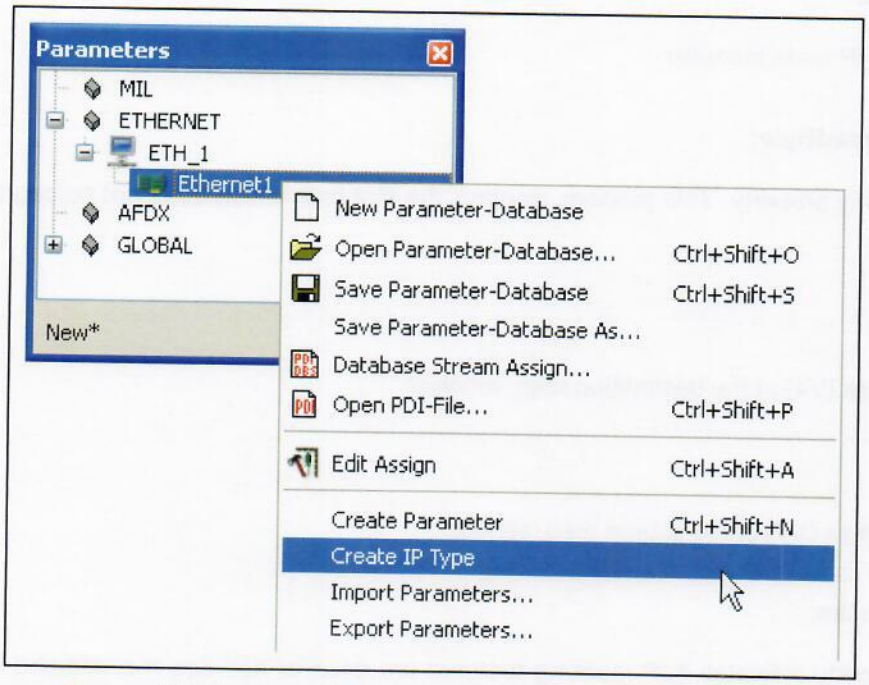


Figure 106: Standard Ethernet – DBS IP Header creation

The 'Create IP Type' command will create a new IP header node for the selected Ethernet header node.

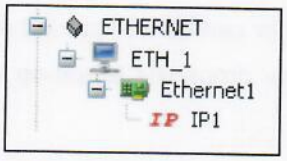


Figure 107: Standard Ethernet – DBS IP Header node

A double click on the 'IPx' node opens the property dialog which supports configuration of the IP properties.

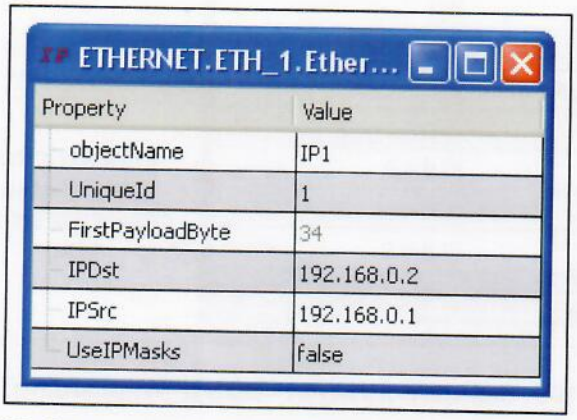


Figure 108: Standard Ethernet - DBS IP Header configuration

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

- **objectName:**

  The name of the IP node

- **UniqueId:**

  A unique IP node identifier

- **FirstPayloadByte:**

  A read only property. This property displays the first byte which does not belong to the header anymore

- **IPDst:**

  IP-Address (V4) of the destination resp. receiver

- **IPSrc:**

  IP-Address (V4) of the source resp. sender

- **UseIPMasks:**

  This property indicates if IP masking features are used or not. See the wildcard section below for further details

2.7.2.2.1 *Wildcarding the IP Header*

Comparable with the MAC header, the IP header can be wild carded as well. If the UseIPMasks property is set to true, the IP node will provide two further properties to set-up wildcarding for the IP source and IP destination addresses.

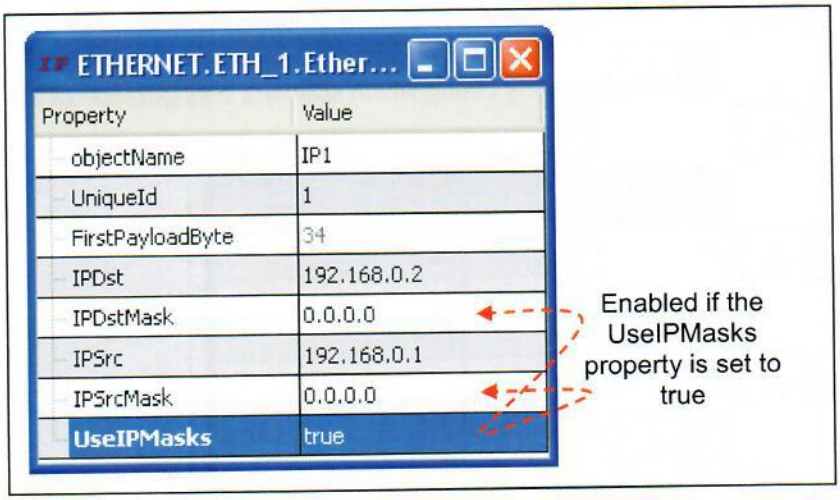


Figure 109: Standard Ethernet - DBS IP Wildcarding

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Wildcarding of an IP is comparable to sub netting in Ethernet networks. Like an IP address the IP mask contains four bytes (32 bits) and is often written using the same "dotted-decimal" notation. For example, a very common IP mask in binary representation 11111111 11111111 11111111 00000000 is typically shown in the equivalent, more readable form 255.255.255.0. All bits of the Mask, set to 0 are ignored resp. accepted. All bits set to 1 have to match the IP definition exactly.

Example:

All IP's from 192.168.0.0 to 192.168.0.255) should be accepted. Thus the following configuration is to be set up:

- *UseIPMasks property is to be set to true*
- *the IPSrc property to 192.168.0.0*
- *The IPSrcMask property to 255.255.255.0*

Note: As all values of the last IP section will be accepted the IPSrc property can be set to any accepted IP address like 192.168.0.50 or 192.168.0.123 as well

2.7.2.3 UDP Header creation

Selection of an IP node and right mouse click offers a context menu for creation of an UDP Header node related to the IP header.

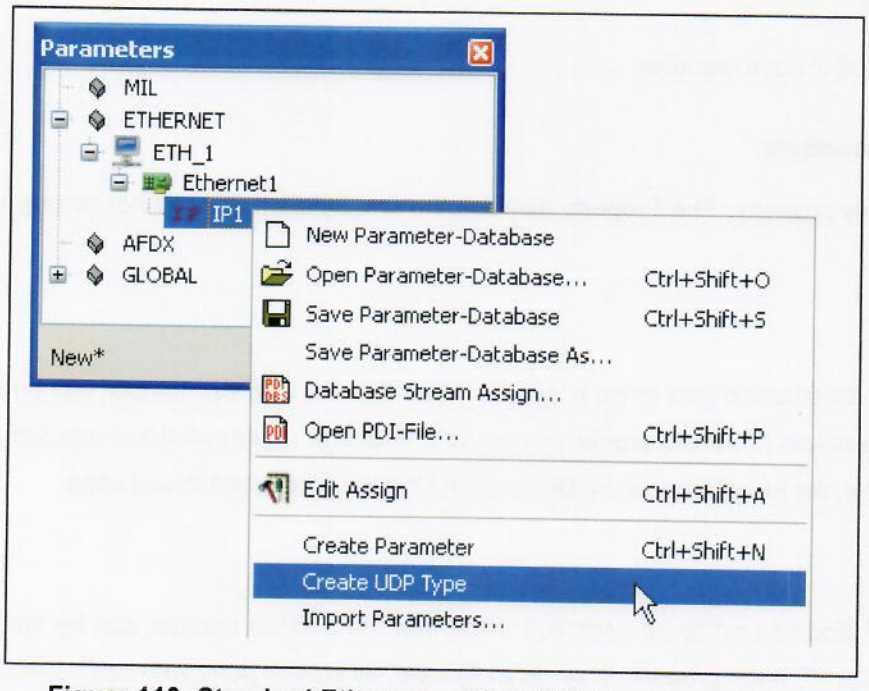


Figure 110: Standard Ethernet – DBS UDP Header creation

The 'Create UDP Type' command will create a new UDP header node for the selected IP header node.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

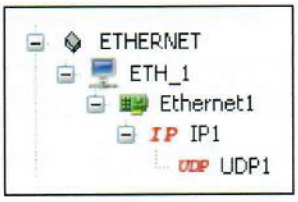


Figure 111: Standard Ethernet – DBS UDP Header node

A double click on the 'UDPx' node opens the property dialog which supports configuration of the UDP properties.

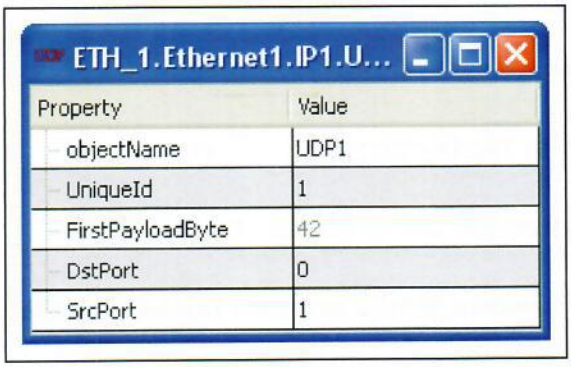


Figure 112: Standard Ethernet - DBS UDP Header configuration

- **objectName:**

  The name of the UDP node

- **UniqueId:**

  A unique UDP node identifier

- **FirstPayloadByte:**

  A read only property. The Property displays the first byte which does not belong to the header anymore

- **DstPort:**

  The UDP Destination port to be accepted. Note that the DstPort number can be set to a wild-card placeholder "*". If the property is set to a wildcard, all destination Ports will match to this node. Thus, the knowledge of the Destination Port is not required in any case

- **SrcPort:**

  The UDP Source port to be accepted. Note that the SrcPort number can be set to a wildcard placeholder "*". If the property is set to a wildcard, all source ports will match to this node. Thus, the knowledge of the Source Port is not required in any case

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

2.7.3 Custom Protocol creation

In addition to the AFDX specific extension to create an Ethernet, IP or UDP header, custom protocols can be setup on each node below an Ethernet stream. Basically Protocol conditions are the same as Conditional Parameters, where an additional property "ProtocolCondition" is provided to indicate that the conditional Parameter is a Protocol Parameter. For further information about Conditional Parameters see chapter 4, Conditional Parameters.

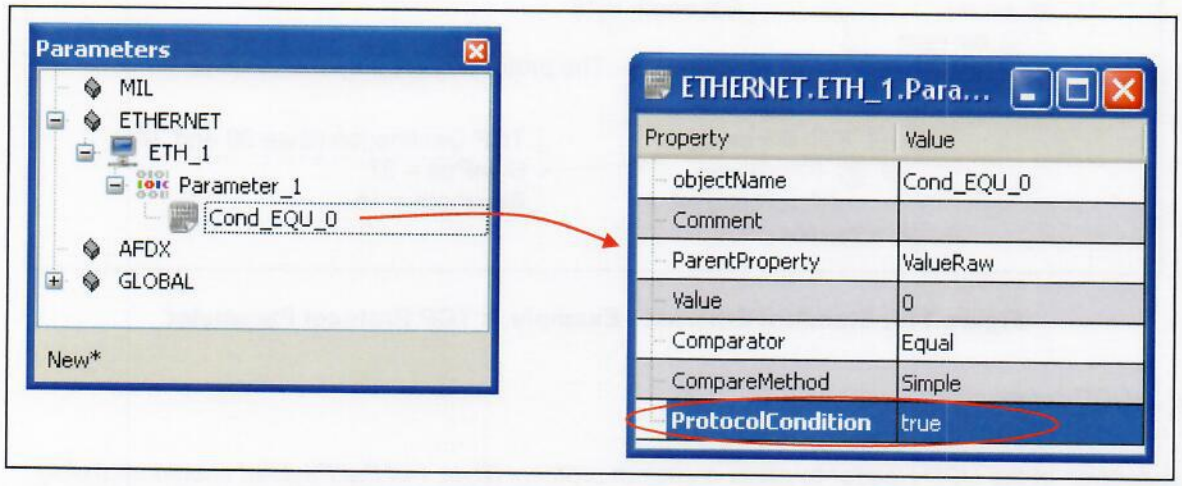


Figure 113: Standard Ethernet - DBS Custom Protocol

Note that the first byte, to which Parameters as well as the Custom Protocols can be placed on, are component dependent. The AFDX component running in standard Ethernet mode supports Parameters and custom protocols to be placed even on the Ethernet Frame Header. Hence, a completely custom protocol can be set-up. In contrast to this, the TCP/IP and UDP components only provide parameters and custom protocols to be placed on the payload as the header information is not available on socket level.

The following examples demonstrate the set-up resp. usage of a custom protocol tree.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Example 1 (AFDX, running in standard Ethernet mode):

Extension of the IP Header to parse TCP Header Information (TCP Source Port and TCP Destination Port)

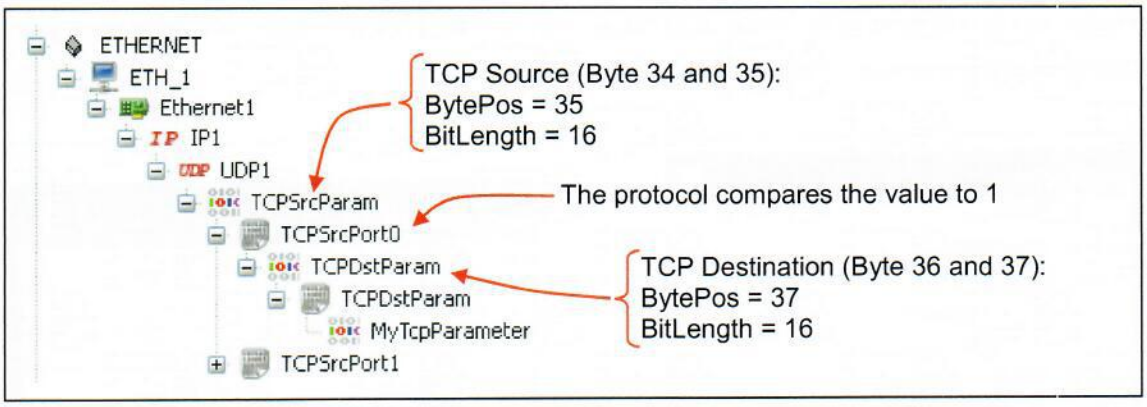


Figure 114: Standard Ethernet - Example 1: TCP Protocol Parameter

Example 2 (UDP component):

Extension of the UDP Header to parse a custom protocol (Byte 1 of the Payload should be 0xAA)

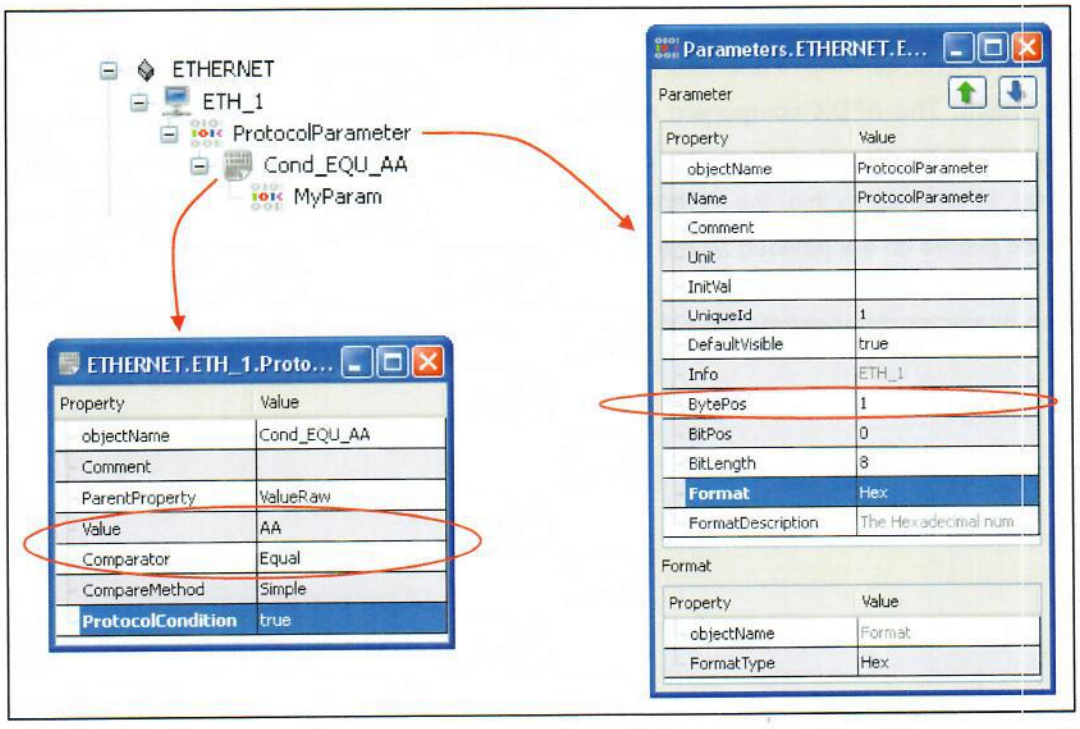


Figure 115: Standard Ethernet - Example 2: UDP Header Extension

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Example 3 (AFDX, running in standard Ethernet mode):

Setup of a custom protocol, where Byte 0-3 and Byte 4-7 should be less than 0x1234 resp. less than 0x5678

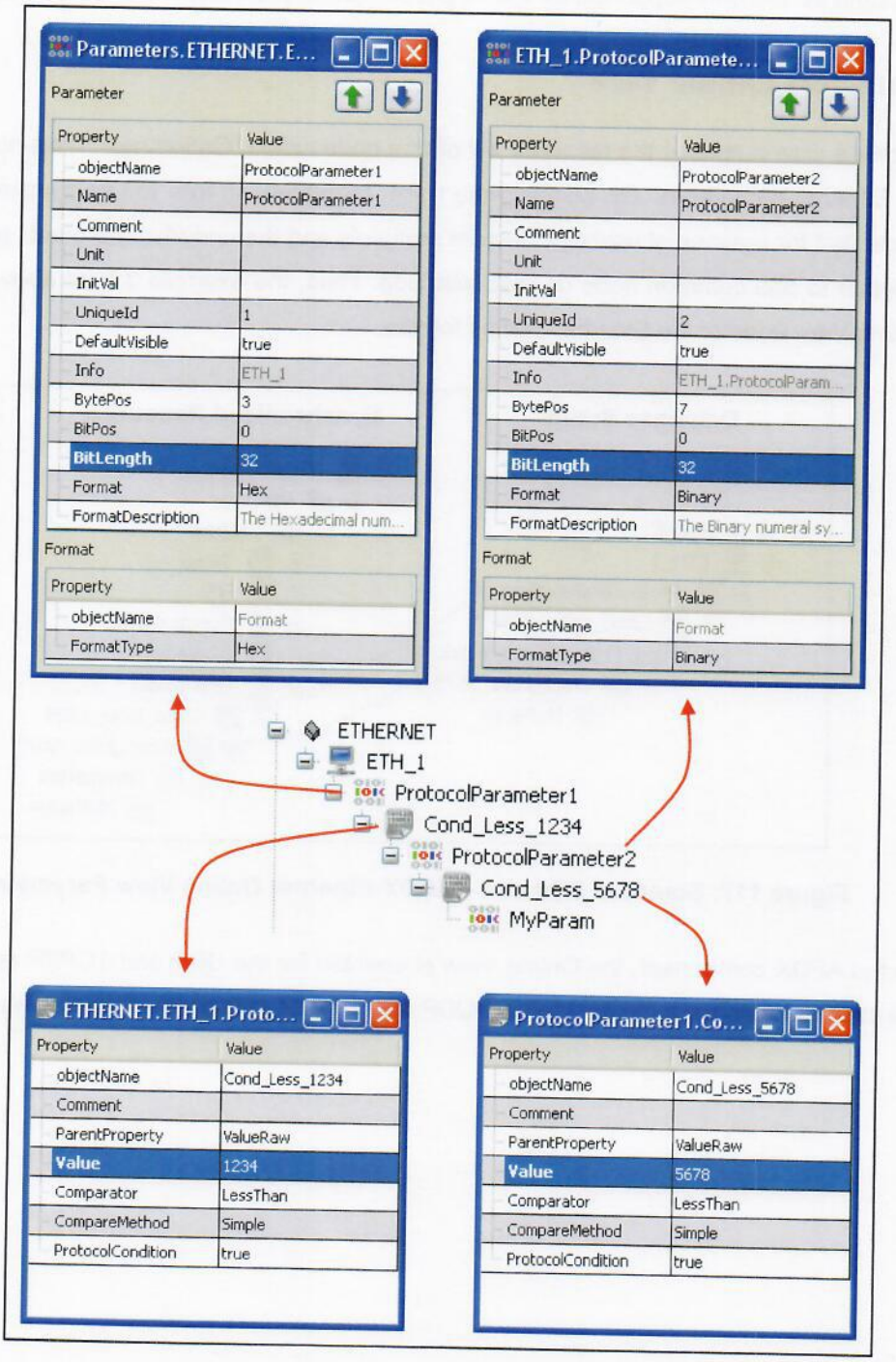


Figure 116: Standard Ethernet - Example 3: Custom Protocol

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

2.7.4 Parameter handling on Resource Side

Depending on the resource, standard Ethernet parameters are supported on different locations. While the Parameter online view is available on every resource, the parameters in the chronological capture mode and the generic send list are only supported for the AFDX component running in standard Ethernet mode.

2.7.4.1 Parameters Online View

For parameter online view purposes the resource will offer a node called "OnlineView". This node has to be enabled to use Ethernet parameters. On enabling the node, the database tree will be synchronised to the Online View. Note that for reasons of usability, custom protocols and the underlying protocol parameter will be grouped together to one common node on resource side. Thus, the example 3 from above is synchronised to the Online View node of the Ethernet port as follows.

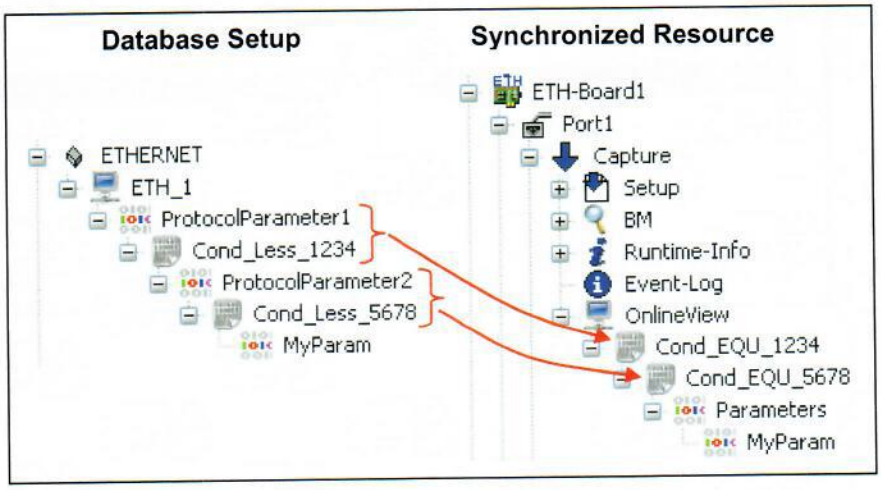


Figure 117: Standard Ethernet – AFDX-Ethernet Online View Parameters

Comparable to the AFDX component, the Online View is enabled for the UDP and TCP/IP receive buffers. Hence, the tree structure below the Online View for UDP and TCP/IP will be exactly the same:

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

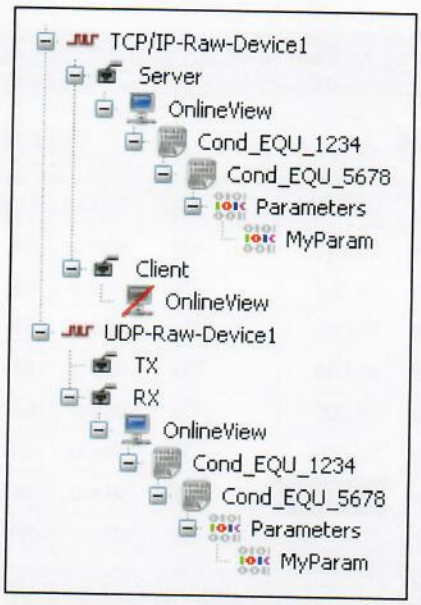


Figure 118: Standard Ethernet – UDP and TCP/IP Online View Parameters

2.7.4.2 Parameters in the chronological capture mode

If the ports are opened and the parameters are properly setup within the database, the chronological capture mode will display the parameters associated to a particular captured Ethernet Frame. Selecting a message in the chronological bus monitor opens the detailed message view. In addition to the Meta information a 'Parameters' node is displayed in the 'Message Information' window. Expanding that node will display the list off all parameters, associated with the current message.

Selection of a single parameter displays its location in the raw data of the message by highlighting the associated data word(s) or parts of it in red:

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

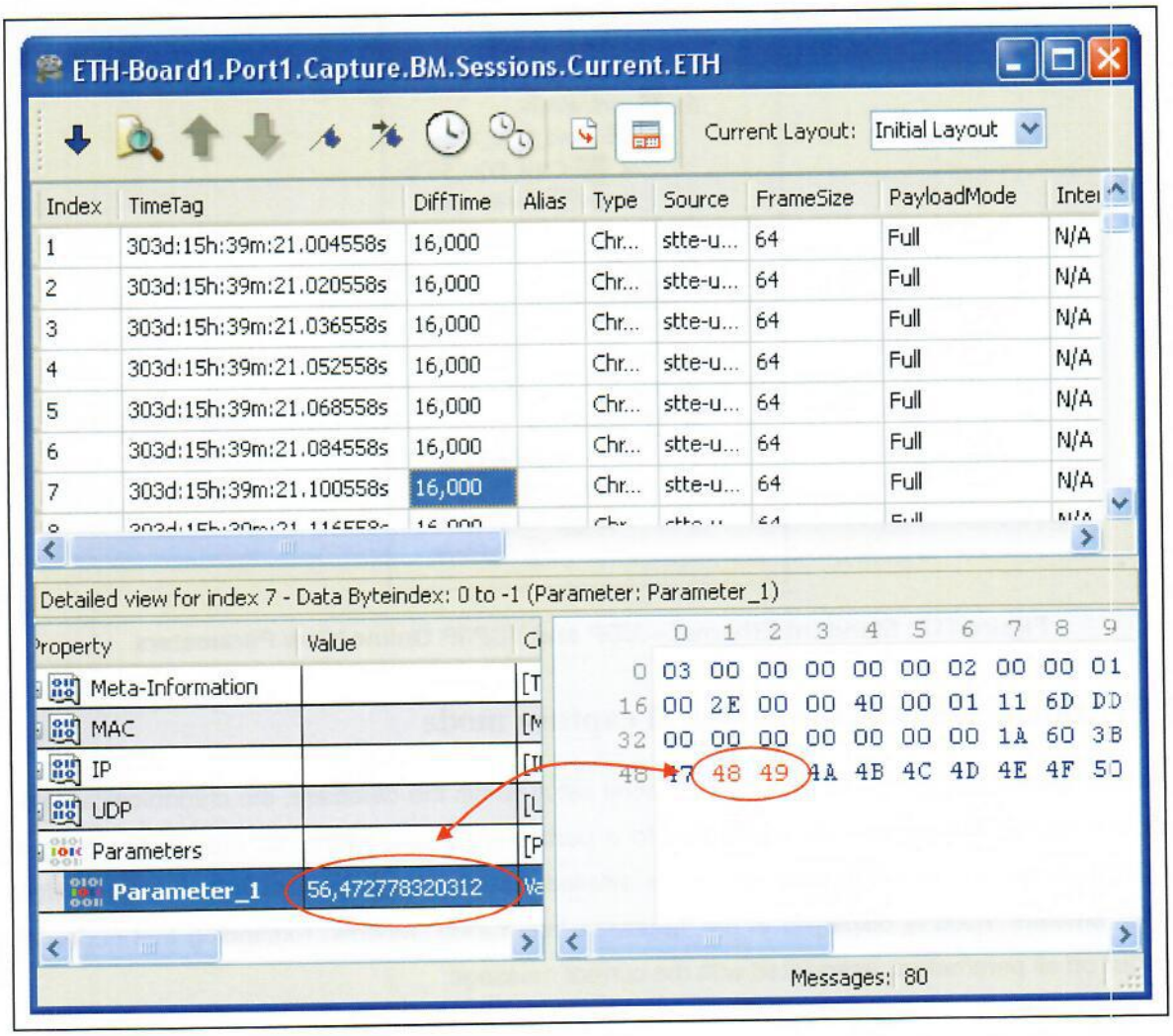


Figure 119: Standard Ethernet - Parameters in the Index View

2.7.4.3 Parameters in the generic send list

Comparable to the Parameters of the chronological capture mode, the generic send list provides parameters as well. Opening a detailed view of the chronological capture mode will be exactly the same view as used for the generic send list, despite that the parameters of the generic send list are editable. Thus, this is the only parameter location, where standard Ethernet parameters can be used for send purposes as well.

To use parameters, create and double click on the packet to open the detailed view editor of the packet.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

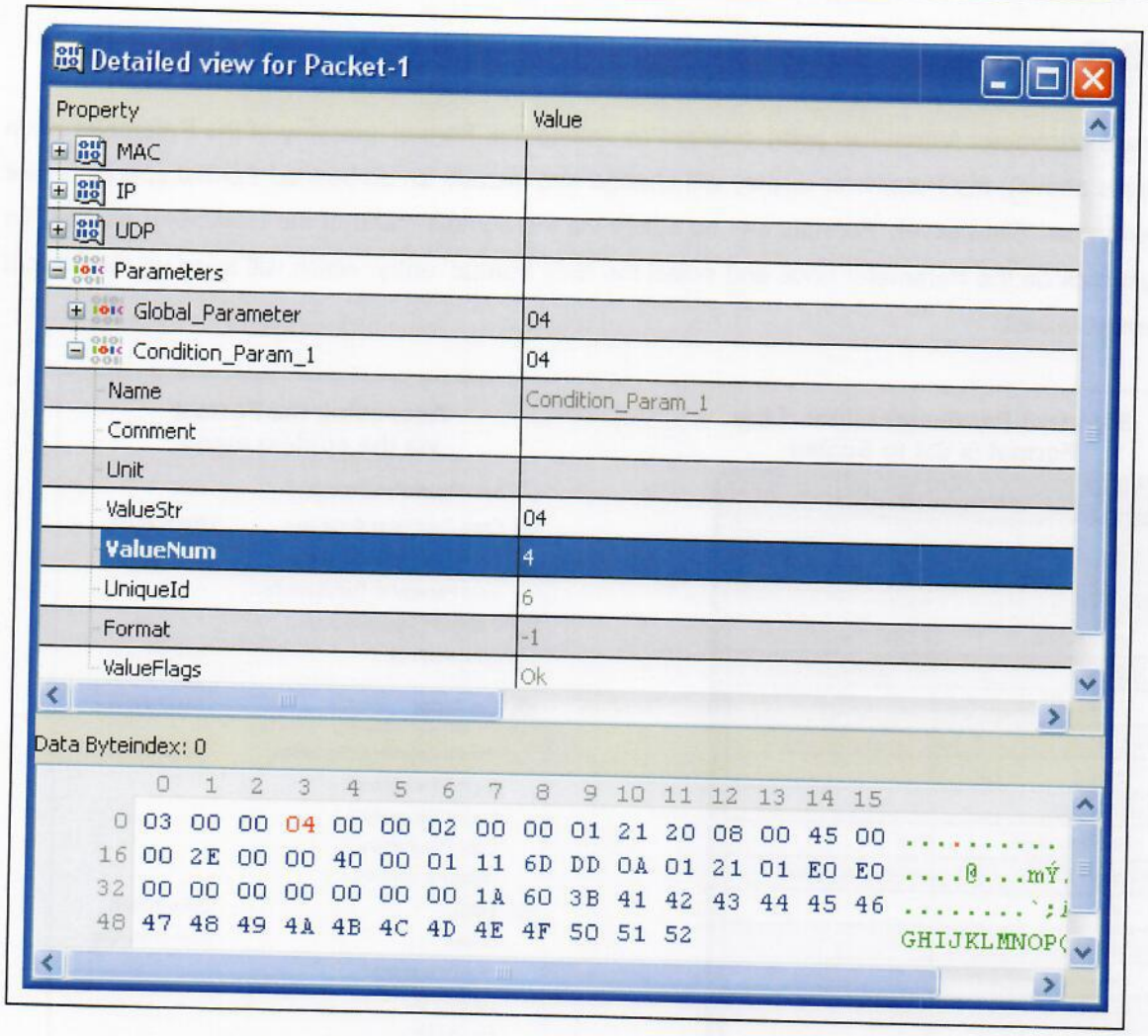


Figure 120: Standard Ethernet - Parameters for a generic send list packet

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

3 Edit Parameter Formats

As soon as a parameter format has been selected by setting the 'Format' property of the Parameter node (see sections above), the Parameter display will change and provide an embedded Format specific editor for the Parameter. Alternatively Formats can be edited via the context menu of the Parameter as well. To do so, right click on the Parameter node and select the 'Edit Format' entry, which will bring up the Format specific editor as well.

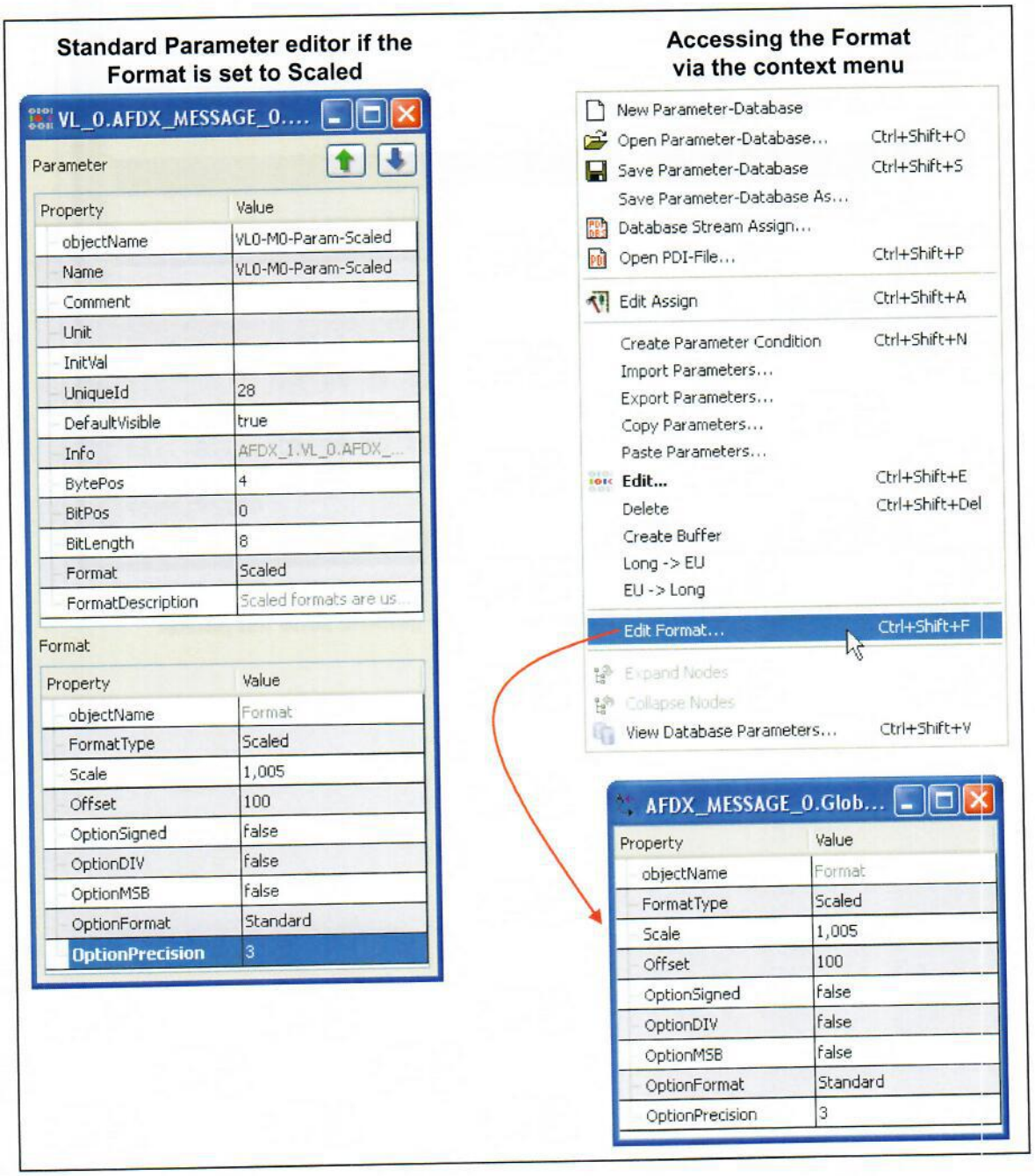


Figure 121: Editing Parameter Formats – Scaled Example

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Note that not all Parameter Formats offer further configuration via the Format editor. The following sections will describe the most important parameter formats resp. their configuration via the Format editor. Currently the following formats are supported:

- Simple Formats
  - Binary
  - Octal
  - Decimal
  - Hexadecimal
  - ASCII
  - ASCII 7
- Complex Formats
  - BCD
  - Float
  - Scaled
  - Enumerated
  - Logic
  - Function

3.1 Scaled, BCD and Float

The example above displays the editor of a 'Scaled' or 'BCD' parameter type. While Scaled and BCD parameters can be of any bit length, Float Parameters are either 32 or 64 bit. Thus, the displayed Engineering unit value of the parameter is derived according to the function:

*Value = Scale * Raw + Offset, where Raw is the Binary raw value of the Parameter*

All options of a scaled parameter can be edited in the table:

- The **OptionSigned** allows definition of 2's complement interpretation (true) or unsigned handling (false)
- If the **OptionDIV** is set, the related scale value is used as divisor instead of a Multiplier: Value = Raw/ Scale + Offset. This may be used to prevent rounding errors if Scale would be odd (e.g. rounding errors due to Scale would need too many digits for 1/7)
- If the **OptionMSB** is set to true, the Scale property specifies the MSB (Most Significant Bit) instead of the LSB (Least Significant Bit). This may also be used to prevent odd or long Scale Values

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

- The **OptionFormat** is used to specify the output string format, which can be either Standard, Decimal, Scientific or FixedPrecision
- The **OptionPrecision** specifies the amount of precision digits to be displayed

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

3.2 Enumerated

If the Format is set to Enumerated, the Editor provides a table style interface to setup the enumerated states.

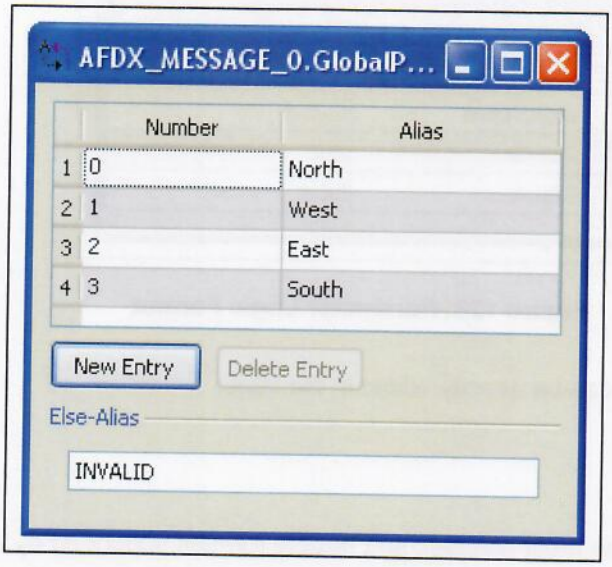


Figure 122: Parameter Enum Format

New enumerated definitions are added via the 'New Entry' button. The 'Number' and 'Alias' can be edited directly within the table. The 'Else-Alias' case will be used for all not defined enum values. The Else-Alias string may also contain the following placeholders to insert the raw value of the parameter:

- %h: Hex Raw Value
- %H: Hex Raw Value with uppercase letters
- %d: Decimal Raw Value
- %o: Octal Raw Value
- %b: Binary Raw Value

Furthermore it is possible to define Enum Ranges.

You can specify for an Alias a Value Range e.g. 1-10. In this case for the Raw Values 1 to 10 the corresponding Alias is displayed. This has the following Restrictions:

- If you control such Parameters and you enter the Alias belonging to a Range the minimum value is used written.
- Raw Values matching more than one overlapping range result in an undefined Alias display
- You may use the % placeholders from above also within a Enum Range, but they cannot be used for controlling

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

3.3 Logic

See following example for a 'Logic' parameter type. A logic parameter has to be exactly one bit wide.

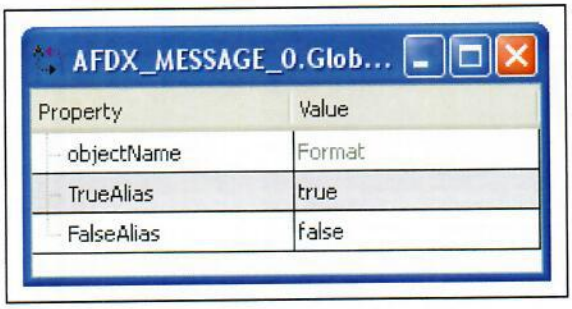


Figure 123: Parameter Logic Format

The 'TrueAlias' and 'FalseAlias' can be directly edited in the table.

3.4 Function

Function formats are currently the most complex and flexible formats supported by the PBA.pro. The Function format can be used either to calculate the EU value according to a function formula or to convert it via a script routine.

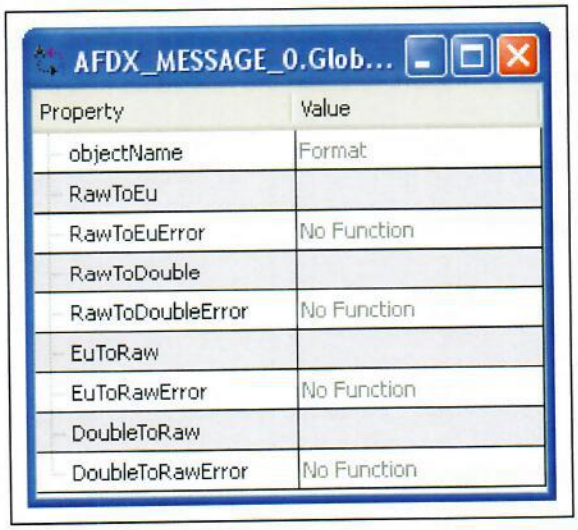


Figure 124: Parameter Function Format

To use a function format, the following conversion routines resp. properties have to be populated resp. configured if required:

- **RawToEu:** The formula of this property is used to convert the raw value to the Engineering unit string displayed via ValueStr property of the corresponding parameter instance on the resource side.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

- **RawToDouble:** The formula of this property is used to convert the raw value to a double value displayed via ValueNum property of the corresponding parameter instance on the resource side
- **EuToRaw:** The formula of this property is used to convert the Engineering unit string to the raw value. This EU String can be edited via the ValueStr property of the corresponding parameter instance on the resource side
- **DoubleToRaw:** The formula of this property is used to convert the double value to the raw value. This double value can edited via ValueNum property of the corresponding parameter instance on the resource side

As Formula either a standard supported Formula or a script method, of which the return value will be used to populate the corresponding parameter instance property (ValueNum resp ValueStr), can be called.

An example of a formula as well as of a Script is given via the following example. Note that the current raw value is specified via the RAW keyword.

Formula:

SIN(RAW/255)

Script:

Python defines a function in the following way, the file containing the script must be sourced

def TellNumber(Value):

return "The Number is: " + str(Value)

The function above may be called with:

SCRIPT(py,TellNumber,RAW)

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

4 Conditional Parameters

For all standard Parameters conditions can be created below a parameter as child node. Conditions are used to build parameter hierarchies based on conditional evaluation of a parameter value. Depending on the current condition all Parameters below the condition can become valid or invalid. Thus, a kind of multiplexing or conditional parameters can be realised via conditions, where a condition can hold multiple children of type standard Parameter.

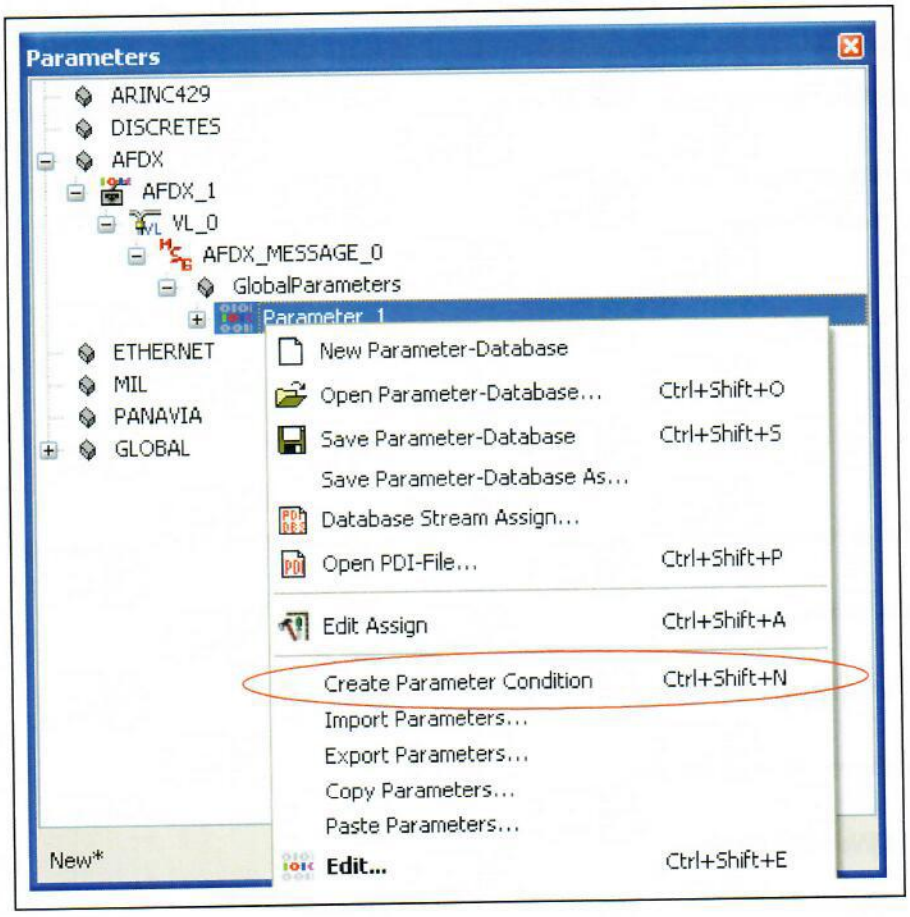


Figure 125: Creating a Conditional Parameter

Structural Example:

```
Param1
   |_Cond1 (Param1 < 10, Validate children)
   |    |_P2
   |    |_P3
   |_Cond2 (Param1 > 10, Validate Children)
   |    |_P4
   |    |_P5
   |_Cond3 (Param1 == 10, --> Set Parent color to RED)
```

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Based on the evaluation within the condition, a Boolean result value indicates if a condition (and all its children) should become valid. The evaluation is based on a simple value compared to

- Equal
- Not equal
- Less or Equal
- Less than
- Higher or Equal
- Higher than

of either the RAW value (Bit value), the numeric or the string EU value of the parameter. For more complex compare operations the compare type can be set to function, where V1 of the Formula is to be used as placeholder for the input parameter value.

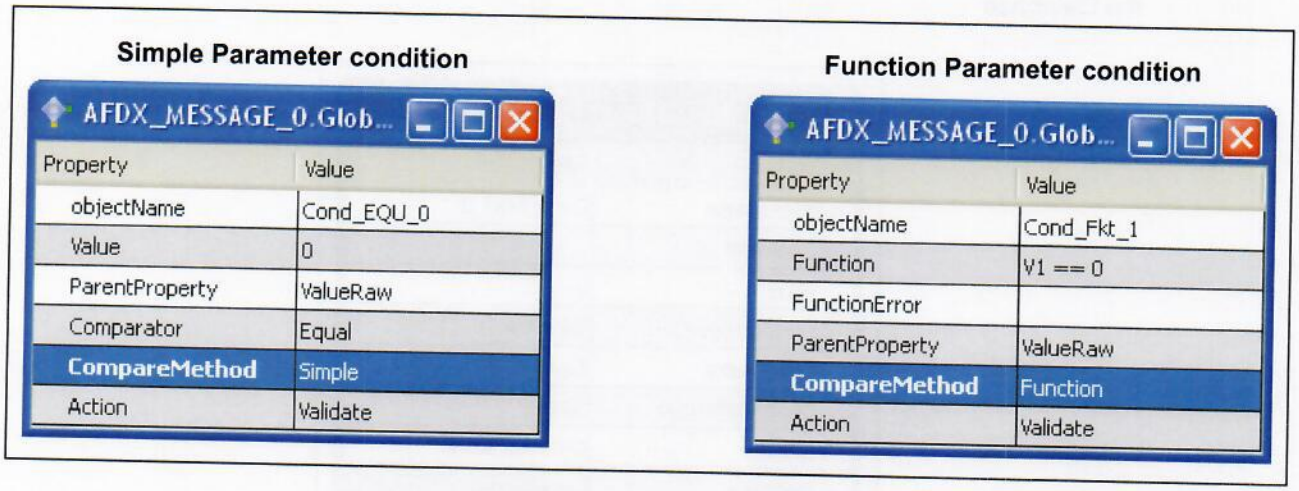


Figure 126: Parameter condition configuration

Depending on the Condition Type, the following properties are provided:

- **objectName:** The Name of the Condition
- **Value** (Simple Conditions only): The Value to compare the property of the parent parameter to
- **Function** (Function Conditions only): The formula to compare the property of the parent parameter to. Note that V1 is used as placeholder for the parent parameter property. Furthermore the function result will be converted to a Boolean value, where all values not equal to 0 are interpreted as true (e.g. "(V1 > 100) && (V1 < 200)")
- **FunctionError** (Function Conditions only): Displays an error information if the current function formula is not correct

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

- **ParentProperty:** The value source resp. parent property to compare the condition to
- **Comparator** (Simple Conditions only): The Compare operation which can be set to Equal, Not equal, Less or Equal, Less than, Higher or Equal or Higher than
- **CompareMethod:** Indicates whether a standard, simple compare operation is to be done or a function should be used for the evaluation
- **Action:** The action to perform if the condition matches:
  - **Disabled**

    The Condition is not used and therefore always invalid
  - **ColorizeParent**

    If the condition returns true, the ValueStr and ValueNum properties of the Parent parameter will be displayed coloured according to colours defined in the condition. If the Action is set to ColorizeParent, two additional properties are provided to specify the text and background colours. This can be used as General Alarm condition which is already included in the Database

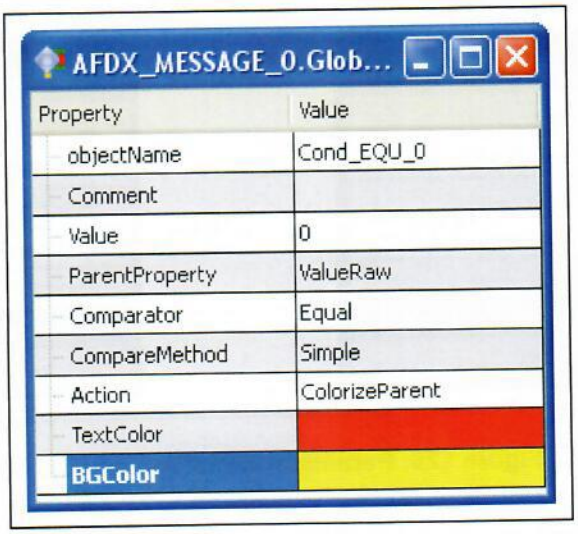


Figure 127: Parameter Condition configuration - Colorize Parent

  - **Validate**

    If the condition returns *true*, the condition and all its sub nodes are set to valid else to invalid
  - **Invalidate**

    If the condition returns *false*, the condition and all its sub nodes are set to valid else to invalid

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

- **Relative**

This mode is used to define Relative Structure Parameters

A Parameter, which holds one or more "Relative" conditions is used as switch parameter for extracting several structures repeated in the buffer. Mixing Relative and Non Relative conditions on the same Parameter is not supported.

Example:

P1

CondType_1 (P1 == 1)
PC1-1
PC1-2
PC1-3
CondType_2 (P1 == 2)
PC2-1
PC2-2
CondType_3 (P1 == 3)
PC3-1
PC3-2
PC3-3
CondType_4 (P1 == 4)
PC4-1
PC4-2

P1 is a Parameter starting at position x in the Buffer. This parameter defines the type of the structure. CondType_x are conditions, which have the Action set to Relative. PCx-y are Conditional Parameters located on the related condition. These parameters can be used to extract the following scenarios

P1	P1 == 1
PC1-1	
PC1-2	
PC1-4	
P1	P1 == 3
PC3-1	
PC3-2	
PC3-3	
P1	P1 == 4
PC4-1	
PC4-2	

Table 2: Parameter Condition - Scenarios

- The different Conditions may have different Parameters and different sizes as displayed in the Property *StructureSize* of the Condition
- If the property *SizeMode* is set to Automatic, the Properties *StructureSize* and *StructureOffset* are calculated according the parameters belonging to the structure. If you use *Manual* Structure size mode, you have to specify the boundaries of the structure with this Properties
- The property *StructureOffset* can be used to define, where the start of the first parameter of the structure should be. Hence, this property is a kind of padding area to shift the parameters of the structure the given amount of Bits.
- The first P1 starts at the defined P1 Position in the Database
- The following P1 starts at the offset specified through the Property *StructureSize* of the previous matched Condition. Eventually you have to add Padding Parameters to the structure to archive the correct size, resp. use *Manual* Structure Size mode and specify the related size and offset
- The Evaluation is stopped at the end of the buffer or if a value of P1 is not defined through a Condition. If the value of P1 is not defined through a Condition, the following structures cannot be evaluated and the evaluation is aborted
- It is in the responsibility of the user, to verify that each condition is only valid once. If more conditions may match to the value of P1, the first valid is used. (e.g. if you have 2 Conditions specifying Value == 5)
- If at least one Relative Condition is defined on a Parameter, the other conditions should also have the Relative type, or they are ignored if any relative condition matches
- If the same condition matches more than one time, the parameters will display the values of the last matching structure.
- The parameter P1 may be located anywhere in the structure. If the Condition Parameter is not the first in the structure, the property *StructureOffset* must be used accordingly

General Notes for Conditions

- If Conditional Parameters are used on the Standard Online Sources (AFDX Sampling/ Queuing Port, ARINC 429 Label, MIL-1553 RT Sub addresses etc), you may not receive every occurrence of the parameter as the values are sampled with the Display Update Rate. Due to this problem you may not receive every occurrence of the Data. If it is required that every occurrence has to be handled resp. checked, the parameters have to be added to the "Bus Monitor Online Message View" (See the Bus Monitor Chapter in the PBA.pro Getting Started Manual). This will ensure that every occurrence of the data is evaluated

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

- For MIL-Bus and ARINC 429 parameter you may also initiate a faster update with aid of the *Interrupt(uint)* Signal of the MIL-Bus Transfer, MIL-Bus Sub Address or Receiver Labels. Create a Signal Slot Connection between the *Interrupt* Signal and the *UpdateFromSource* Slot of the *Parameters* Object, either by Script or the Global Connection Manager (see PBA.pro Getting started)

- Parameters located below a Condition can only be controlled, if the Condition currently matches. Otherwise the Write Request is ignored

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

5 Global Parameters

The following chapter will give a short introduction to Global Parameters. Global Parameters assist during the creation and definition of parameters in a general and global way. They are reusable for more than one resource and can be applied repetitive. Structures are a set of parameters, they are defined below the GLOBAL/ Structures node. Once a structure is defined, it can be applied to the specific resource node (MIL/ ARINC/ AFDX...). The characteristic feature of a structured parameter is, that there will always be a reference to its GLOBAL definition. All changes in the GLOBAL definition will be reflected in the specific parameter object.



Figure 128: Global Structures DBS Node

5.1 Global Structures

The 'Structures'" object is always a sub node of GLOBAL and can not be deleted. It is only a container for structured parameter objects without further properties. Selection of a 'Structures' node and right mouse click offers a context to create a new Structure. Structures can be applied to all nodes in the parameter tree where an ordinary parameter can be placed (all parameter container objects). For example a MIL Sub address, an AFDX Message or an ARINC Label.

5.1.1 Global Structures

Structures are used to organize parameters in a hierarchical way as a kind of structured data block. A Structure contains a set of general described Parameters and Paddings (which are used as simple placeholders) and can have further Sub-Structures for grouping.

Properties of a Structure:

A Structure has an (object-)Name and a specific, optional comment. The BlockSize defines the usual bit pattern size of the used parameters. It is used as a helper to know where your parameters are located. In a MIL environment it is common to use a Block Size of 16 Bits (Word) and calculate with Word Positions, whereas a Block Size of 8 Bit (Byte) is the common value for AFDX. The BitOffset can be used to shift all parameters defined below a Structure relative to the bit position defined by structs/ parameters/ paddings and their Struct-BitOffsets above.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

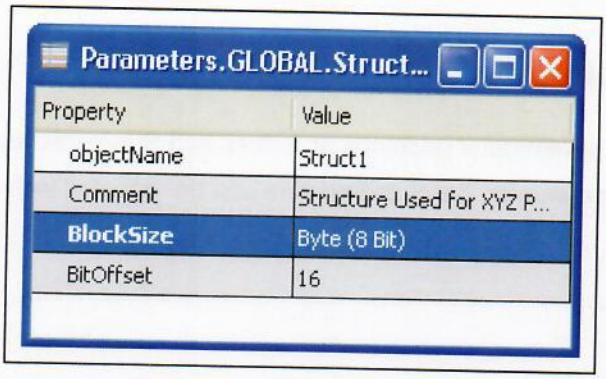


Figure 129: Global Structure configuration

5.1.2 Creating Global Sub-Structures, Global Parameters and Paddings

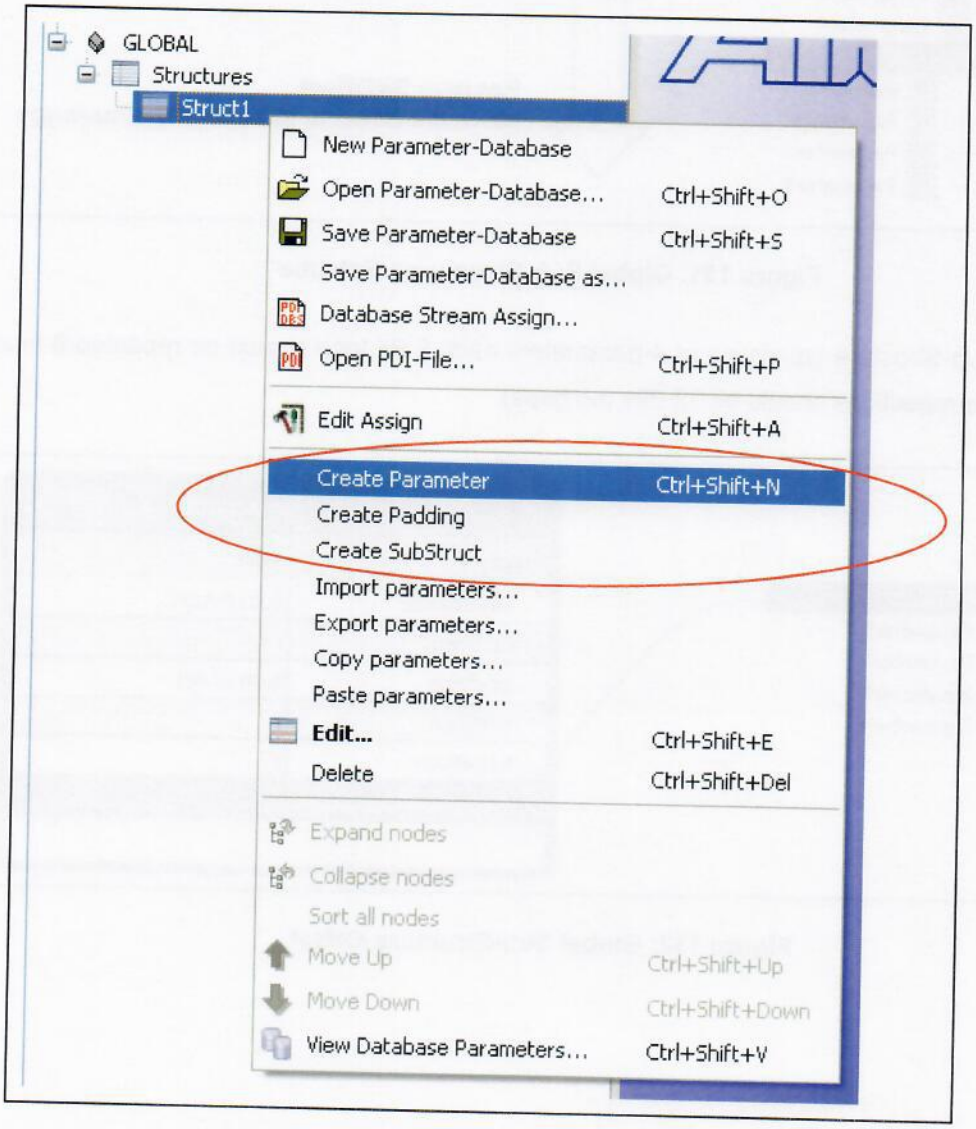


Figure 130: Global Structures context menu

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

5.1.2.1 Global Sub-Structures

A Sub-Structure groups a set of parameters/ paddings or further Sub-Structures as structure. If a structure is applied to a specific resource parameter, the Sub-Structure can be repeated with an offset between each repetition.

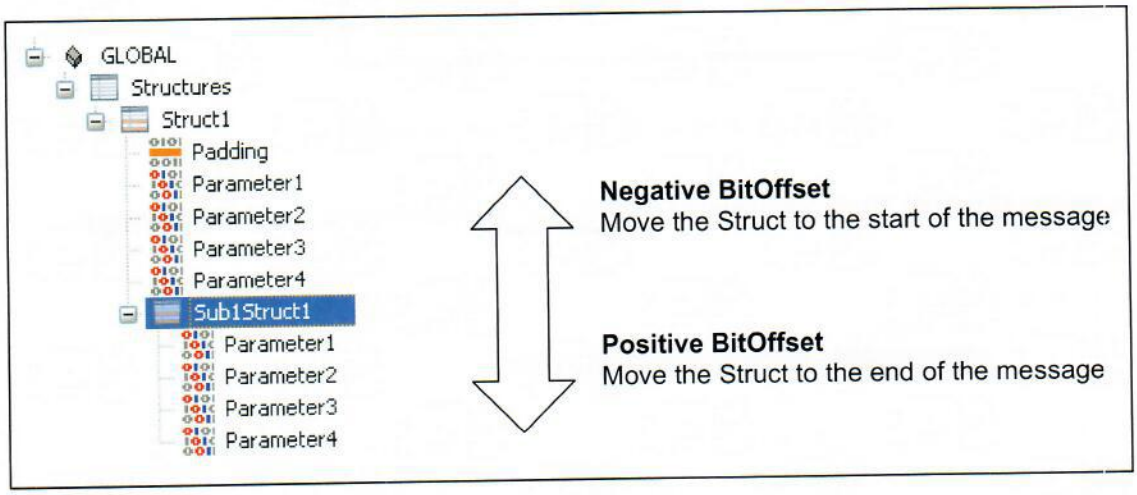


Figure 131: Global Sub-Structures-Scheme

For example a Sub-Structure consisting of 4 parameters each 8 Bit long should be repeated 8 times. The offset between the repetitions should be 32 Bits (no gaps).

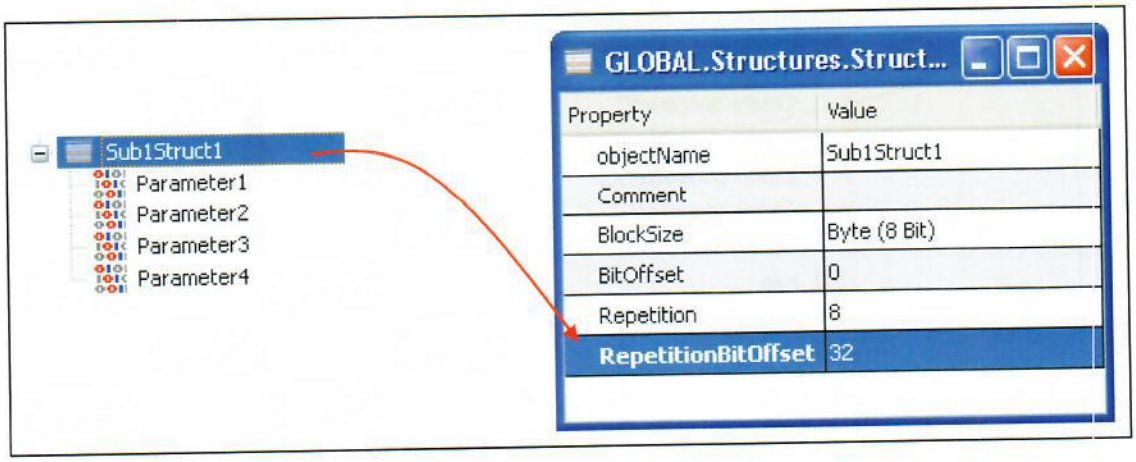


Figure 132: Global Sub-Structure-Offset

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Artisan Technology Group is an independent supplier of quality pre-owned equipment

Gold-standard solutions

Extend the life of your critical industrial, commercial, and military systems with our superior service and support.

We buy equipment

Planning to upgrade your current equipment? Have surplus equipment taking up shelf space? We'll give it a new home.

Learn more!

Visit us at **artisantg.com** for more info on price quotes, drivers, technical specifications, manuals, and documentation.

Artisan Scientific Corporation dba Artisan Technology Group is not an affiliate, representative, or authorized distributor for any manufacturer listed herein.

We're here to make your life easier. How can we help you today?

(217) 352-9330 | sales@artisantg.com | artisantg.com

ARTISAN® TECHNOLOGY GROUP