Operating Instruction Manual

Generic DTM for EtherNet/IP Adapter Devices and Modular Generic DTM for modular EtherNet/IP Adapter Devices

Configuration of EtherNet/IP Adapter Devices

V1.203

Hilscher Gesellschaft für Systemautomation mbH

www.hilscher.com

DOC070203OI09EN | Revision 9 | English | 2018-03 | Released | Public

Table of Contents

 © Hilscher, 2007-2017

© Hilscher, 2007-2017

1 Introduction

1.1 About this Manual

This manual describes how to use the EtherNet/IP generic Adpater DTM to configure within a FDT Framework the device parameters of an EtherNet/IP Adapter device. Modular EtherNet/IP Devices can be configured using the modular generic EtherNet/IP Generic Adapter DTM.

To perform the configuration procedure the generic EtherNet/IP Generic Adapter DTM is inserted in a network project to the Master busline of an EtherNet/IP Scanner DTM.

1.1.1 Descriptions of the Dialog Panes

The table below gives an overview for the individual dialog panes descriptions:

Section	Subsection	Manual Page
Configuration	*Overview Configuration*	17
	General	19
	Modules (modular DTM) (for modular DTM only)	20
	Electronic Keying	21
	Connection	23
	Assembly	26

Table 1: Descriptions Dialog Panes

1.1.2 Online Help

The generic EtherNet/IP Generic Adapter DTM contains an integrated online help facility.

- To open the online help, click on the **Help** button or press the **F1** key.

1.1.3 List of Revisions

Index	Date	Version	Chapter	Revision
9	17-02-23	1.203 (and 1.0203)	All, 1.4.1	Revised and updated. Section *Requirements* updated (Windows 8 added.)
10	18-02-14	1.203 (and 1.0203)		Versioning information revised: Title page and this section

Table 2: List of Revisions

© Hilscher, 2007-2017

1.1.4 Conventions in this Manual

Notes, operation instructions and results of operation steps are marked as follows:

Notes

Important: <important note you must follow to avoid malfunction>

Note: <general note>

<note, where to find further information>

Operation Instructions

1. <instruction>
2. <instruction>

or

➢ <instruction>

Results

<result>

Langage Convention for EtherNet/IP

The EtherNet/IP specification defines the term "Scanner" instead of "Master" and "Adapters" instead of "Slave".

© Hilscher, 2007-2017

1.2 Legal Notes

Copyright

© Hilscher Gesellschaft für Systemautomation mbH

All rights reserved.

The images, photographs and texts in the accompanying materials (in the form of a user's manual, operator's manual, Statement of Work document and all other document types, support texts, documentation, etc.) are protected by German and international copyright and by international trade and protective provisions. Without the prior written consent, you do not have permission to duplicate them either in full or in part using technical or mechanical methods (print, photocopy or any other method), to edit them using electronic systems or to transfer them. You are not permitted to make changes to copyright notices, markings, trademarks or ownership declarations. Illustrations are provided without taking the patent situation into account. Any company names and product designations provided in this document may be brands or trademarks by the corresponding owner and may be protected under trademark, brand or patent law. Any form of further use shall require the express consent from the relevant owner of the rights.

Important notes

Utmost care was/is given in the preparation of the documentation at hand consisting of a user's manual, operating manual and any other document type and accompanying texts. However, errors cannot be ruled out. Therefore, we cannot assume any guarantee or legal responsibility for erroneous information or liability of any kind. You are hereby made aware that descriptions found in the user's manual, the accompanying texts and the documentation neither represent a guarantee nor any indication on proper use as stipulated in the agreement or a promised attribute. It cannot be ruled out that the user's manual, the accompanying texts and the documentation do not completely match the described attributes, standards or any other data for the delivered product. A warranty or guarantee with respect to the correctness or accuracy of the information is not assumed.

We reserve the right to modify our products and the specifications for such as well as the corresponding documentation in the form of a user's manual, operating manual and/or any other document types and accompanying texts at any time and without notice without being required to notify of said modification. Changes shall be taken into account in future manuals and do not represent an obligation of any kind, in particular there shall be no right to have delivered documents revised. The manual delivered with the product shall apply.

Under no circumstances shall Hilscher Gesellschaft für Systemautomation mbH be liable for direct, indirect, ancillary or subsequent damage, or for any loss of income, which may arise after use of the information contained herein.

Liability disclaimer

The hardware and/or software was created and tested by Hilscher Gesellschaft für Systemautomation mbH with utmost care and is made

available as is. No warranty can be assumed for the performance or flawlessness of the hardware and/or software under all application conditions and scenarios and the work results achieved by the user when using the hardware and/or software. Liability for any damage that may have occurred as a result of using the hardware and/or software or the corresponding documents shall be limited to an event involving willful intent or a grossly negligent violation of a fundamental contractual obligation. However, the right to assert damages due to a violation of a fundamental contractual obligation shall be limited to contract-typical foreseeable damage.

It is hereby expressly agreed upon in particular that any use or utilization of the hardware and/or software in connection with

- Flight control systems in aviation and aerospace;
- Nuclear fusion processes in nuclear power plants;
- Medical devices used for life support and
- Vehicle control systems used in passenger transport

shall be excluded. Use of the hardware and/or software in any of the following areas is strictly prohibited:

- For military purposes or in weaponry;
- For designing, engineering, maintaining or operating nuclear systems;
- In flight safety systems, aviation and flight telecommunications systems;
- In life-support systems;
- In systems in which any malfunction in the hardware and/or software may result in physical injuries or fatalities.

You are hereby made aware that the hardware and/or software was not created for use in hazardous environments, which require fail-safe control mechanisms. Use of the hardware and/or software in this kind of environment shall be at your own risk; any liability for damage or loss due to impermissible use shall be excluded.

Warranty

Hilscher Gesellschaft für Systemautomation mbH hereby guarantees that the software shall run without errors in accordance with the requirements listed in the specifications and that there were no defects on the date of acceptance. The warranty period shall be 12 months commencing as of the date of acceptance or purchase (with express declaration or implied, by customer's conclusive behavior, e.g. putting into operation permanently).

The warranty obligation for equipment (hardware) we produce is 36 months, calculated as of the date of delivery ex works. The aforementioned provisions shall not apply if longer warranty periods are mandatory by law pursuant to Section 438 (1.2) BGB, Section 479 (1) BGB and Section 634a (1) BGB [Bürgerliches Gesetzbuch; German Civil Code] If, despite of all due care taken, the delivered product should have a defect, which already existed at the time of the transfer of risk, it shall be at our discretion to either repair the product or to deliver a replacement product, subject to timely notification of defect.

The warranty obligation shall not apply if the notification of defect is not asserted promptly, if the purchaser or third party has tampered with the

 © Hilscher, 2007-2017

products, if the defect is the result of natural wear, was caused by unfavorable operating conditions or is due to violations against our operating regulations or against rules of good electrical engineering practice, or if our request to return the defective object is not promptly complied with.

Costs of support, maintenance, customization and product care

Please be advised that any subsequent improvement shall only be free of charge if a defect is found. Any form of technical support, maintenance and customization is not a warranty service, but instead shall be charged extra.

Additional guarantees

Although the hardware and software was developed and tested in-depth with greatest care, Hilscher Gesellschaft für Systemautomation mbH shall not assume any guarantee for the suitability thereof for any purpose that was not confirmed in writing. No guarantee can be granted whereby the hardware and software satisfies your requirements, or the use of the hardware and/or software is uninterruptable or the hardware and/or software is fault-free.

It cannot be guaranteed that patents and/or ownership privileges have not been infringed upon or violated or that the products are free from third-party influence. No additional guarantees or promises shall be made as to whether the product is market current, free from deficiency in title, or can be integrated or is usable for specific purposes, unless such guarantees or promises are required under existing law and cannot be restricted.

Confidentiality

The customer hereby expressly acknowledges that this document contains trade secrets, information protected by copyright and other patent and ownership privileges as well as any related rights of Hilscher Gesellschaft für Systemautomation mbH. The customer agrees to treat as confidential all of the information made available to customer by Hilscher Gesellschaft für Systemautomation mbH and rights, which were disclosed by Hilscher Gesellschaft für Systemautomation mbH and that were made accessible as well as the terms and conditions of this agreement itself.

The parties hereby agree to one another that the information that each party receives from the other party respectively is and shall remain the intellectual property of said other party, unless provided for otherwise in a contractual agreement.

The customer must not allow any third party to become knowledgeable of this expertise and shall only provide knowledge thereof to authorized users as appropriate and necessary. Companies associated with the customer shall not be deemed third parties. The customer must obligate authorized users to confidentiality. The customer should only use the confidential information in connection with the performances specified in this agreement.

The customer must not use this confidential information to his own advantage or for his own purposes or rather to the advantage or for the purpose of a third party, nor must it be used for commercial purposes and this confidential information must only be used to the extent provided for in this agreement or otherwise to the extent as expressly authorized by the disclosing party in written form. The customer has the right, subject to the

© Hilscher, 2007-2017

obligation to confidentiality, to disclose the terms and conditions of this agreement directly to his legal and financial consultants as would be required for the customer's normal business operation.

Export provisions

The delivered product (including technical data) is subject to the legal export and/or import laws as well as any associated regulations of various countries, especially such laws applicable in Germany and in the United States. The products / hardware / software must not be exported into such countries for which export is prohibited under US American export control laws and its supplementary provisions. You hereby agree to strictly follow the regulations and to yourself be responsible for observing them. You are hereby made aware that you may be required to obtain governmental approval to export, reexport or import the product.

1.3 Registered Trademarks

Windows® XP, Windows® Vista, Windows® 7, Windows® 8, Windows® 8.1 and Windows® 10 are registered trademarks of Microsoft Corporation.

EtherNet/IP™ is a trademark of ODVA (Open DeviceNet Vendor Association, Inc).

All other mentioned trademarks are property of their respective legal owners.

© Hilscher, 2007-2017

1.4 About EtherNet/IP Generic Adapter DTM

You can use the EtherNetIP generic Adapter DTM to configure the EtherNetIP Slave devices described with EDS files within a FDT Framework. Modular EtherNet/IP Devices are configured using the modular EtherNet/IP generic Adapter DTM.

The information necessary for the configuration of the EtherNetIP Slave devices is stored within the EtherNetIP Scanner device when using the EtherNetIP generic Adapter DTM and thus the Scanner device is configured.

1.4.1 Requirements

System Requirements

- PC with 1 GHz processor or higher
- Windows® XP SP3,
  Windows® Vista (32-Bit) SP2,
  Windows® 7 (32-Bit and 64-Bit) SP1,
  Windows® 8 (32-Bit and 64-Bit),
  Windows® 8.1 (32-Bit and 64-Bit),
  Windows® 10 (32-Bit and 64-Bit)
- Administrator privilege required for installation
- Internet Explorer 5.5 or higher
- RAM: min. 512 MByte, recommended 1024 MByte
- Graphic resolution: min. 1024 x 768 pixel
- Keyboard and Mouse
- Restriction: Touch screen is not supported.

Note: If the project file is used on a further PC,

- this PC must also comply with the above system requirements,
- the device description files of the devices used in the project must be imported into the configuration software SYCON.net on the new PC,
- and the DTMs of the devices used in the project must also be installed on that further PC.

Requirements EtherNet/IP Generic Adapter DTM

Requirements for working with the EtherNetIP generic Adapter DTM are:

- Installed FDT/DTM V 1.2 compliant frame application,
- Installed EtherNet/IP Scanner DTM,
- EDS file of the devices to be configured. The parameters must be set manually according to the EDS file.
- The user needs to reload the Device Catalog

© Hilscher, 2007-2017

1.5 Dialog Structure of the EtherNet/IP Generic Adapter DTM

The graphical user interface of the DTM is composed of different areas and elements listed hereafter:

1. A header area containing the **General Device Information**,
2. The **Navigation Area** (area on the left side),
3. The **Dialog Pane** (main area on the right side),
4. **OK**, **Cancel**, **Apply**, **Help**,
5. The **Status Line** containing information e. g. the online-state of the DTM.

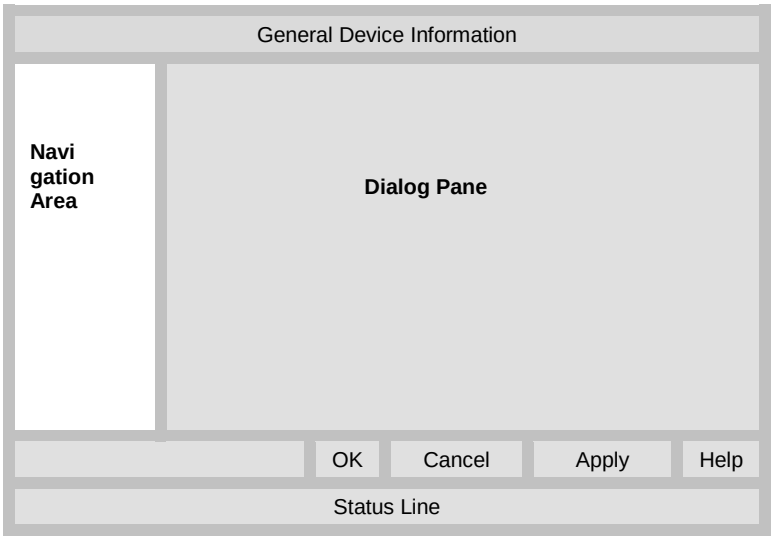


Figure 1: Dialog Structure of the Generic EtherNetIP Adapter DTM

© Hilscher, 2007-2017

1.5.1 General Device Information

Parameter	Meaning
IO Device	Name of the device
Vendor	Vendor name of the device
Device ID	Identification number of the device
Vendor ID	Identification number of the vendor

Table 3: General Device Information

1.5.2 Navigation Area

The **Navigation Area** contains folders and subfolders to open the dialog panes of the DTM.

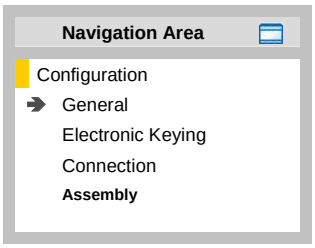


Figure 2: Navigation Area

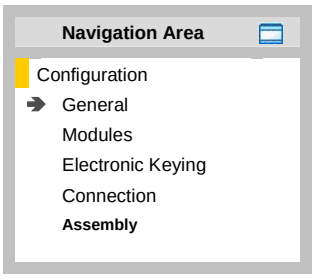


Figure 3: Navigation Area (modulare DTM)

➢ Select the required folder and subfolder.

➢ The corresponding Dialog pane is displayed.

Hide / display Navigation

	Hiding the navigation area (above right side).
Show navigation area	**Opening** the navigation area (below left side).

© Hilscher, 2007-2017

1.5.3 Dialog Panes

At the dialog pane the **Configuration** panes are opened via the corresponding folder in the navigation area.

Configuration	
General	On the pane **General** EtherNet/IP Adapter information is displayed. For further information, refer to section *General* on page 19.
Modules (modular DTM only)	At the **Modules** page the modules can be configured. For further information, refer to section *Modules (modular DTM)* on page 20.
Electronic Keying	At the **Electronic Keying** pane for online validation of adapters an electronic keying method can be selected and the keying can be configured. For further information, refer to section *Electronic Keying* on page 21.
Connection	At the **Connection** pane the connection can be parameterized. For further information, refer to section *Connection* on page *23* .
Assembly	At the **Assembly** pane the configured input / output connections of the EtherNet/IP adapter are displayed. For further information, refer to section *Assembly* on page *26*.

Table 4: Overview Dialog Panes

1.5.4 OK, Cancel, Apply and Help

OK, **Cancel**, **Apply** and **Help** you can use as described hereafter.

	Meaning
OK	To confirm your latest settings, click **OK**. All changed values will be applied on the frame application database. *The dialog then closes.*
Cancel	To cancel your latest changes, click **Cancel**. Answer to the safety query **Configuration data has been changed. Do you want to save the data?** by **Yes**, **No** or **Cancel**. **Yes**: The changes are saved or the changed values are applied on the frame application database. *The dialog then closes.* **No**: The changes are not saved or the changed values are not applied on the frame application database. *The dialog then closes.* **Cancel**: *Back to the DTM.*
Apply	To confirm your latest settings, click **Apply**. All changed values will be applied on the frame application database. *The dialog remains opened.*
Help	To open the DTM online help, click **Help**.

Table 5: OK, Cancel, Apply and Help

© Hilscher, 2007-2017

1.5.5 Status Bar

The **Status Bar** displays information about the current state of the DTM. The current activity, e.g. the DTM connection state, is signaled graphically via icons in the status bar.

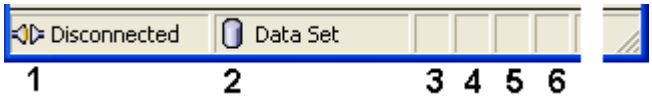
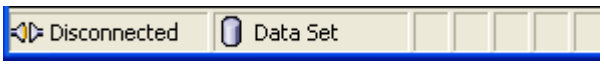


Figure 4: Status Bar – Status Fields 1 to 6

Status Field	Icon / Meaning	
1	**DTM Connection States**	
		Connected: Icon closed = Device is online
		Disconnected: Icon opened = Device is offline
2	**Data Source States**	
		Data set: The displayed data are read out from the instance data set (database).
		Device: The displayed data are read out from the device.
3	**States of the instance Date Set**	
		Valid Modified: Parameter is changed (not equal to data source).

Table 6: Status Bar Icons [1]

Offline State



Online State

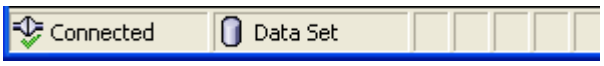


Figure 5: Status Bar Display Example

© Hilscher, 2007-2017

2 Getting started and Instructions Step by Step

2.1 Overview Configuration Steps

The following overview describes the steps to configure a EtherNetIP Adapter device with the EtherNetIP generic Adapter DTM as it is typical for many cases. At this time it is presupposed that the EtherNetIP Scanner DTM installation was already done.

The overview lists all the steps in a compressed form. For detailed descriptions of each step refer to the sections noted in the column *For detailed information see section*.

#	Step	Short Description	For detailed information see section	Page
1	Add EtherNet/IP generic Adapter DTM in the Device Catalog	Add the Adapter in the Device Catalog by importing the device description file to the Device Catalog. Depending of the FDT Container. For netDevice: - **Network > Import Device Descriptions**.	*(See Operating Instruction Manual netDevice and netProject)*	-
2	Load device catalog	Depending of the FDT Container: For netDevice: - select **Network > Device Catalog**, - select **Reload Catalog**.	*(See Operating Instruction Manual netDevice and netProject)*	-
3	Create new project / Open existing project	Depending of the frame application. For the configuration software: - select **File > New** or **File > Open**.	*(See Operating Instruction Manual of the Frame Application)*	-
4	Insert Scannner or Adapter device icon into configuration	Depending of the FDT Container: For netDevice: - in the Device Catalog click to the Scannner icon, - and insert the device icon via drag and drop **to the Root line** in the network view, - in the Device Catalog click to the Adapter device icon, - and insert the device icon via drag and drop to the **Master bus line** in the network view.	*(See Operating Instruction Manual netDevice and netProject)*	-
5	Configure Adapter device	Configure the Adapter device. - Double click to the device icon of the Adapter. - The generic Adapter DTM configuration dialog is displayed. In the generic Adapter DTM configuration dialog: - select **Configuration > General**, - set the IP settings for the adapter device, - select **Configuration > Modules** (for modular DTM only), - configure the modules of the modular EtherNet/IP Adapter, - select **Configuration > Electronic Keying**, - select the keying method and configure it if necessary, - select **Configuration > Connection**, - configure the connection points,	*Configuring Adapter Parameters* *General* *Modules (modular DTM) (for modular DTM only)* *Electronic Keying* *Connection*	18 19 20 21 23

© Hilscher, 2007-2017

#	Step	Short Description	For detailed information see section	Page
5	Configure Adapter device	- select **Configuration > Assembly**, - configure Instance ID and Data length, - configure a *modular device* according to the description *Configuring a modular EtherNet/IP Adapter*, - close the Adapter DTM configuration dialog via **OK**.	*Assembly* *Configuration Steps for modular EtherNet/IP Devices*	23 26 30
6	Configuration Steps Scannner device	Configure the Scanner device via EtherNet/IP Scanner DTM. **Important:** Enter the IP settings of the EtherNet/IP Adapter device.	*(See Operating Instruction Manual DTM for EtherNet/IP Scanner devices)*	-
7	Save project	Depending of the frame application. For the configuration software: - select **File > Save**.	*(See Operating Instruction Manual of the Frame Application)*	-

Table 7: Getting started - Configuration Steps

© Hilscher, 2007-2017

3 Configuration

3.1 Overview Configuration

Configuration Dialog Panes

The table below gives an overview for the **Configuration** dialog panes descriptions:

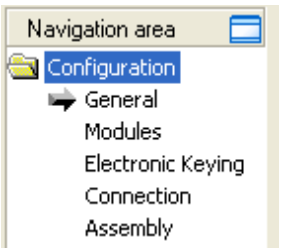
EtherNetIP Adapter DTM	Folder Name / Section	Page
 *Navigation Area – Configuration (EtherNet/IP Generic Adapter DTM)*	*General*	19
	Modules (modular DTM) (for modular DTM only)	20
	Electronic Keying	21
	Connection	23
	Assembly	26
 *Navigation Area - Configuration (EtherNet/IP Modular Generic Adapter DTM)*		

Table 8: Descriptions of the Dialog Panes Configuration

Notice the descriptions in the section *Overview Configuration Steps* on page 15 .

© Hilscher, 2007-2017

3.2 Configuring Adapter Parameters

The following steps are required to configure the parameters of the EtherNet/IP Adapter device using the EtherNet/IP generic Adapter DTM:

(I) In the EtherNet/IP Scanner DTM:

IP Settings for the Adapter Device

1. Make the IP settings for the Adapter device.

(II) In the generic EtherNet/IP Adapter DTM

Modules

2. Configure the modules of the modular EtherNet/IP Adapter (for modular DTM only):

➢ Select **Configuration > Modules** in the navigation area.

Electronic Keying

3. Select the electronic keying method and configure if necessary:

➢ Select **Configuration > Electronic Keying** in the navigation area.

Connection

4. Configure the connection points:

➢ Select **Configuration > Connetcion**in the navigation area.

Assembly

5. Configure Instance ID and Data length:

➢ Select **Configuration > Assembly** in the navigation area.

(III) Configure a Modular Device

6. Configure a *modular device* according to the description in section *Configuration Steps for modular EtherNet/IP Devices* on page 30.

Close Adapter DTM Configuration Dialog

7. Click **OK** in order to close the EtherNet/IP generic Adapter DTM configuration dialog and to store your configuration.

Further Information

For more information refer to section *Electronic Keying* on page 21, to section *Modules (modular DTM)* on page 20t, to section *Connection* on page *23* and to section *Assembly* on page *26* of this document.

© Hilscher, 2007-2017

3.3 General

The **General** dialog page shows the **Description** of the EtherNet/IP Adapter. The **IP Address** is set by the EtherNet/IP Scanner.

To show the current device settings:

- Select **Configuration > General** in the navigation area.

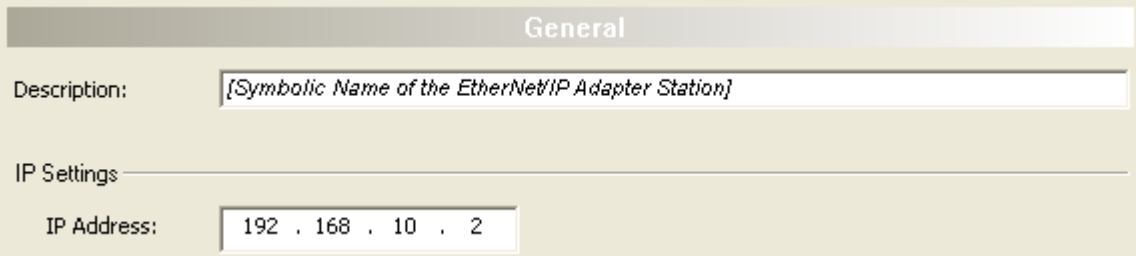


Figure 6: Configuration > General

Parameter	Meaning
Description	Symbolic Name of the EtherNet/IP Adapter device.
IP Settings of the EtherNet/IP Adapter device	
IP Address	The IP address of the EtherNet/IP Adapter device is set in the EtherNet/IP Scanner DTM. Here it is only displayed. The EtherNet/IP Scanner device transmits the IP address of the EtherNet/IP Adapter during startup via the EtherNet/IP network to the EtherNet/IP Adapter and thereby configures the EtherNet/IP Adapter.

Table 9: General Pane Parameters

© Hilscher, 2007-2017

3.4 Modules (modular DTM)

In the EtherNet/IP modular generic Adapter DTM at the **Modules** pane the modules of the modular EtherNet/IP Adapter can be configured.

➢ Select **Configuration > Modules** in the navigation area.

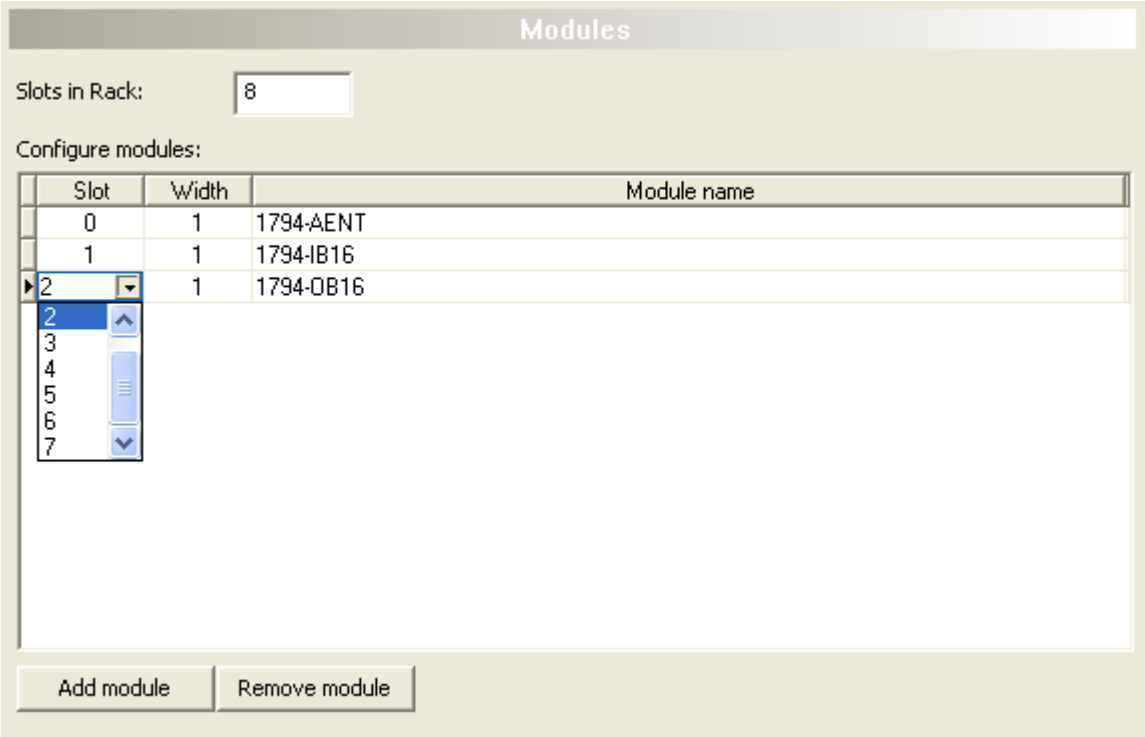


Figure 7: Configuration > Modules Pane (modular DTM)

Parameter	Meaning
Slots in Rack	Total number of slots in Rack
Configure Modules	
Slot	Shows the current **Slot** number assigned to a module. When clicking the slot field, the drop-down-list of the **Slot** numbers is displayed.
Width	Width of the module
Module name	Textual modul name
'Add module'	Use **Add Module** to add a module to the device configuration.
'Remove module'	Use **Remove** to remove the selected module from the configuration.

Table 10: Modules Pane Parameters

© Hilscher, 2007-2017

3.5 Electronic Keying

The concept of **Electronic Keying** was introduced by Allen-Bradley, RA. EtherNet/IP scanner implements compatible concept.

A set of attributes of an EtherNet/IP Adapter can be regarded as its electronic identity which can be used to differentiate adapters based on these attributes. EtherNet/IP scanner employs this electronic identity to build an **Electronic Key** and uses it to verify that an adapter connected to the network is the expected one. **Electronic keying** allows flexible online validation of adapters, provides a method for reliable network configuration.

Attributes of the electronic identity that can be used in keying are as follows: Minor Revision, Major Revision, Product Code, Product Type and Vendor ID.

➢ Select **Configuration > Electronic Keying** in the navigation area.

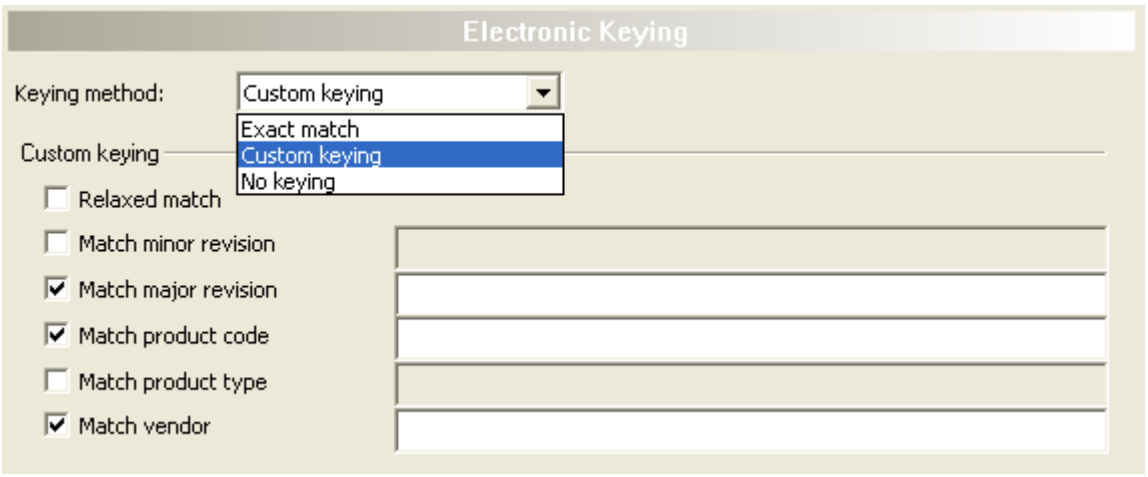


Figure 8: Configuration > Electronic Keying

EtherNet/IP Modular Generic Adapter DTM:

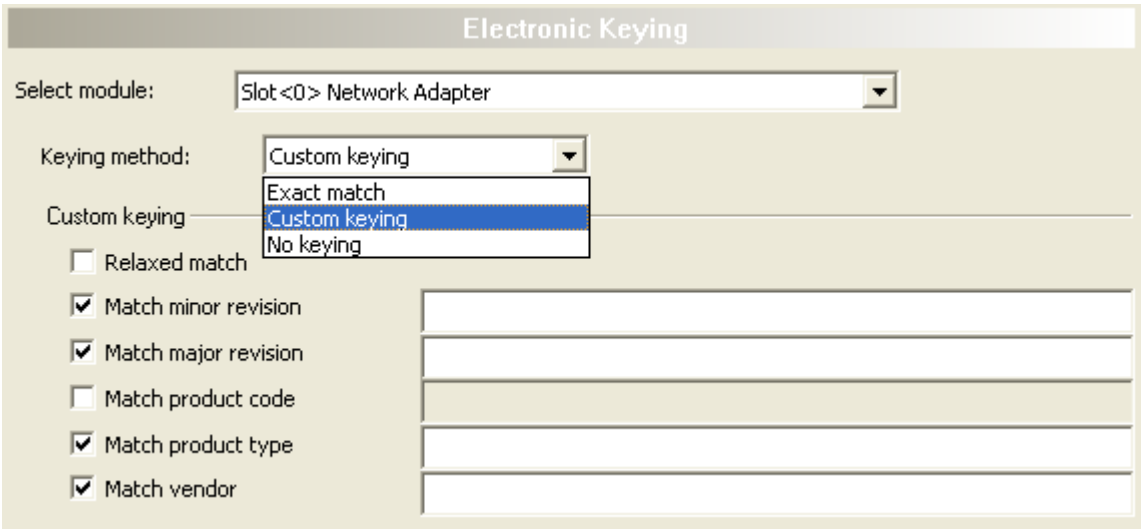


Figure 9: Configuration > Electronic Keying (modular DTM)

© Hilscher, 2007-2017

➢ Select a Module (modular DTM only).

Parameter	Meaning
Select module (modular DTM only)	For modular EtherNet/IP Adapter first in the modular generic Adapter DTM a module must be selected to parameterize the electronic keying parameters.

Table 11: Electronic Keying > Select module

➢ Select a **Keying method**.

Method	Meaning
Exact match	To validate an EtherNet/IP adapter connected to the network all attributes for the electronic identity must correspond to the attributes of an expected device.
Custom keying	To validate an EtherNet/IP adapter connected to the network all attributes must correspond to the configured keying.
No keying	No validation of the device identity.

Table 12: Electronic Keying > Keying Method

For Custom Keying:

➢ Select **Custom Keying** and configure the keying attributes.

Parameter	Meaning
Relaxed Match	If checked: Restricted validation of the electronic identity for devices. To indicate relaxed match to an adapter, the scanner sets bit 7 in major revision.
Match minor Revision	If checked: For electronic keying consistency to minor revision is relevant and gets verified.
Match major Revision	If checked: For electronic keying consistency to major revision is relevant and gets verified.
Match product code	If checked: For electronic keying consistency to product code is relevant and gets verified.
Match product type	If checked: For electronic keying consistency to product type is relevant and gets verified.
Match vendor	If checked: For electronic keying consistency to vendor ID is relevant and gets verified.

Table 13: Electronic Keying > Custom Keying

© Hilscher, 2007-2017

3.6 Connection

At the **Connection** page the connection can be parameterized. Usually the settings are made as given in the EDS file in section connection manager.

Please, refer to chapter *Configuring a modular EtherNet/IP Adapter* in this document explaining modular EDS concept and usage.

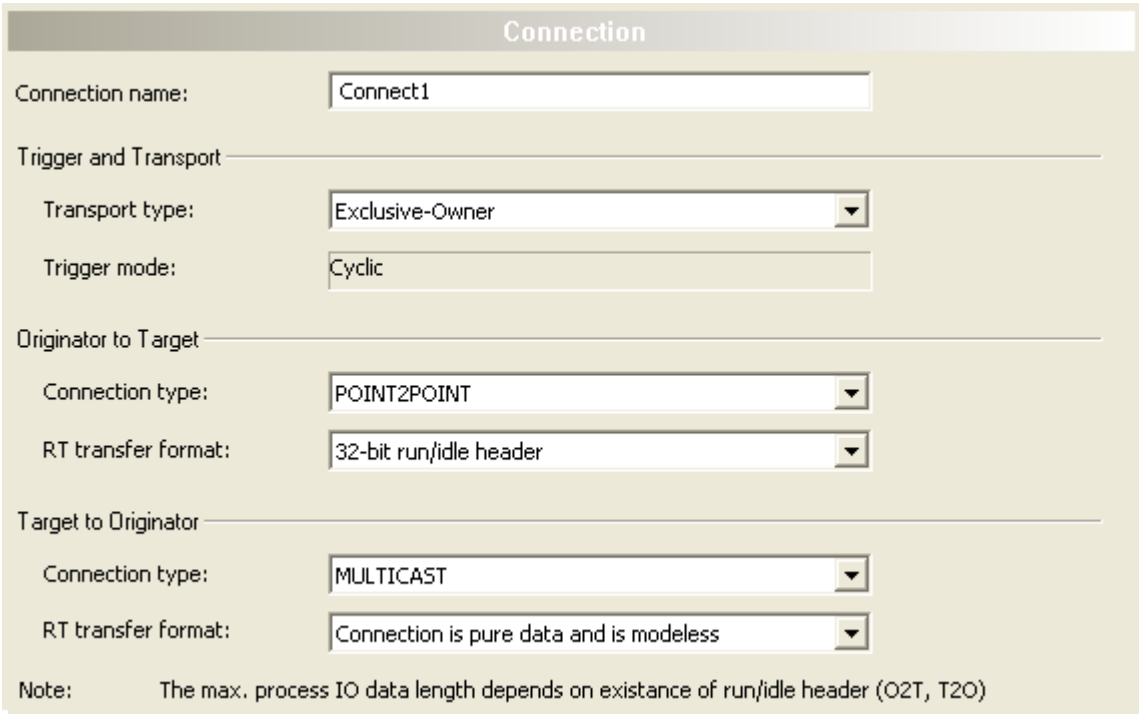


Figure 10: Configuration > Connection

© Hilscher, 2007-2017

EtherNet/IP Modular Generic Adapter DTM:

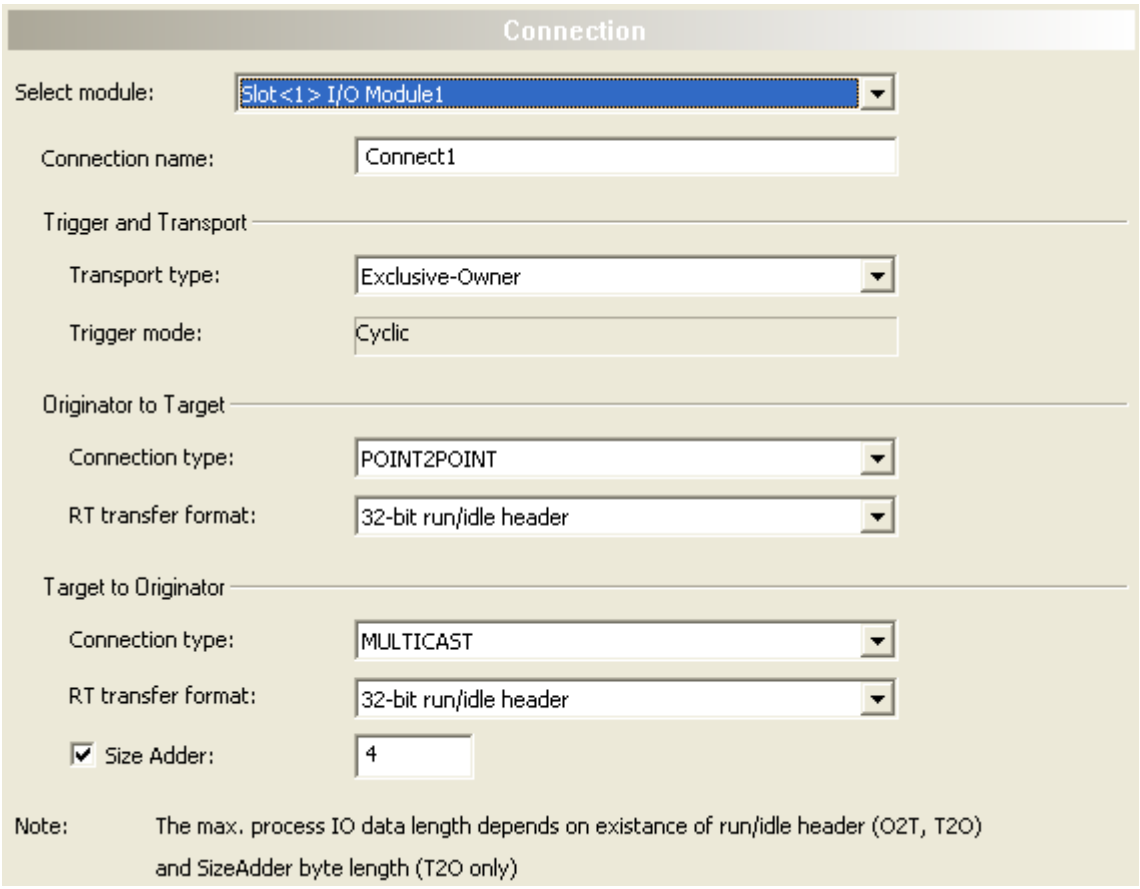


Figure 11: Configuration > Connection, display including SizeAdder (modular DTM). If SizeAdder is not defined, the max. process IO data length depends on existence of run/idle header (O2T, T2O) only

Note: For slot <0> (communication module) there is no Size Adder, for other slots (set modules) there are.

Parameter	Meaning	Range of Value / Value
Select module (modular DTM only)	For modular generic Adapter a module must be selected to parameterize module's connection.	
Connection name	User defined or EDS defined connection name.	Standard: „Connect1“
Trigger and Transport		
Transport type	Transport type Only one of the transport types shall be set.	Listen-Only, Input-Only, Exclusive-Owner
Trigger mode	Only “Cyclic” trigger mode is supported	Cyclic
Originator to Target (O2T)		
Connection type	Connection type for transfer direction originator to target	POINT2POINT, MULTICAST Default: POINT2POINT
RT transfer format	Real time transfer format for transfer direction originator to target	Connection is pure data and is modeless, heartbeat, 32-bit Run/Idle header
Target to Originator (T2O)		
Connection type	Connection type for transfer direction target to originator	POINT2POINT, MULTICAST Default: MULTICAST

© Hilscher, 2007-2017

Parameter	Meaning	Range of Value / Value
RT transfer format	Real time transfer format for transfer direction target to originator	Connection is pure data and is modeless, 32-bit run/idle header
Size Adder	See description on the following page.	

Table 14: Parameter Configuration > Connection

Run/Idle Mode for Realtime Transfer Format

The Run/Idle header is a 32-bit field added to packets flowing in the O2T or T2O direction. In O2T direction it contains several bits of status info. Of primary interest is the least significant bit, which reflects the mode of the connection originator. When this bit is set it means the originator is in Run mode, actively controlling I/O. When cleared this indicates that the originator is in Idle mode, not actively controlling the I/O. Run/Idle is not counted as part of the configured data size in the EDS Connection Manager section. It is counted in the FwdOpen Message O2T and T2O sizes however.

SizeAdder

There could be *size adder* defined in the adapter EDS file. "**ProxyParamSizeAdder**" keyword shall be used to provide minimum, maximum and default values to be added to the "**ProxyParam**" minimum, maximum and default values. "**ProxyParamSizeAdderN**" shall be combined with the corresponding "**ProxyParamN**" entry. The "**ProxyParamSizeAdder**" keyword provides a means for an adapter on a module connection ("**ProxyConnect**") to add adapter data to the module data and return the combined data on the connection. For more details, refer to [2], chapter 7-3.7.2.2.

Note: If **Size Adder** is described in the adapter EDS file, it should be always set to default value.

© Hilscher, 2007-2017

3.7 Assembly

The **Assembly** dialog page shows the list of all input or output connections configured to the EtherNet/IP Adapter.

Please, refer to chapter *Configuring a modular EtherNet/IP Adapter* in this document explaining modular EDS concept and usage.

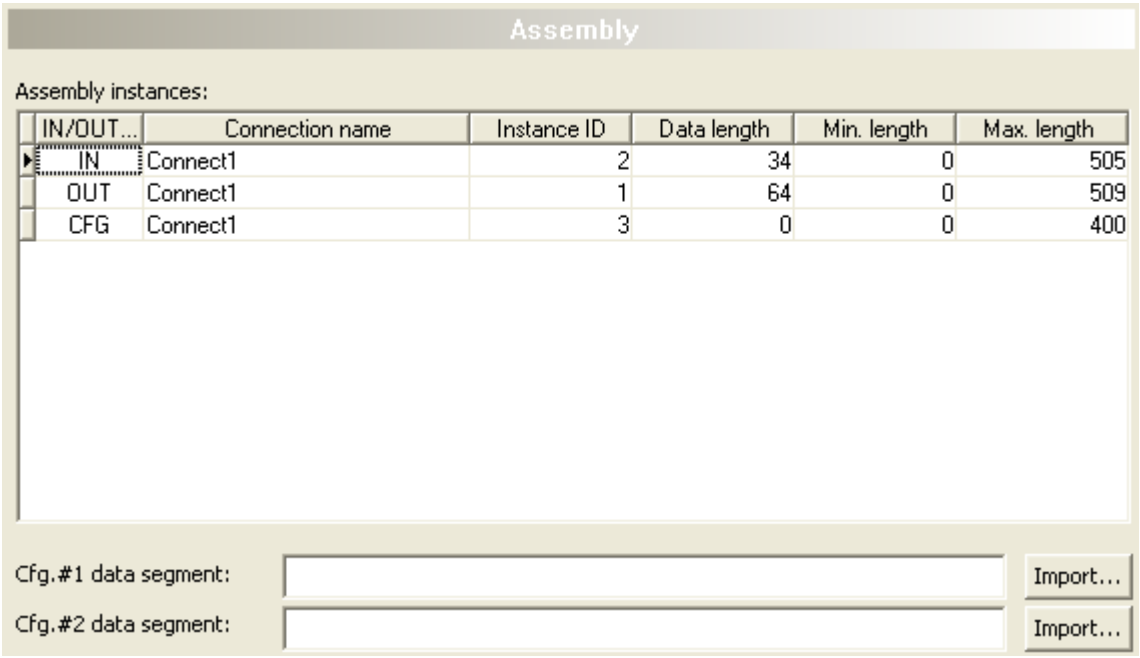


Figure 12: Configuration > Assembly

EtherNet/IP Modular Generic Adapter DTM:

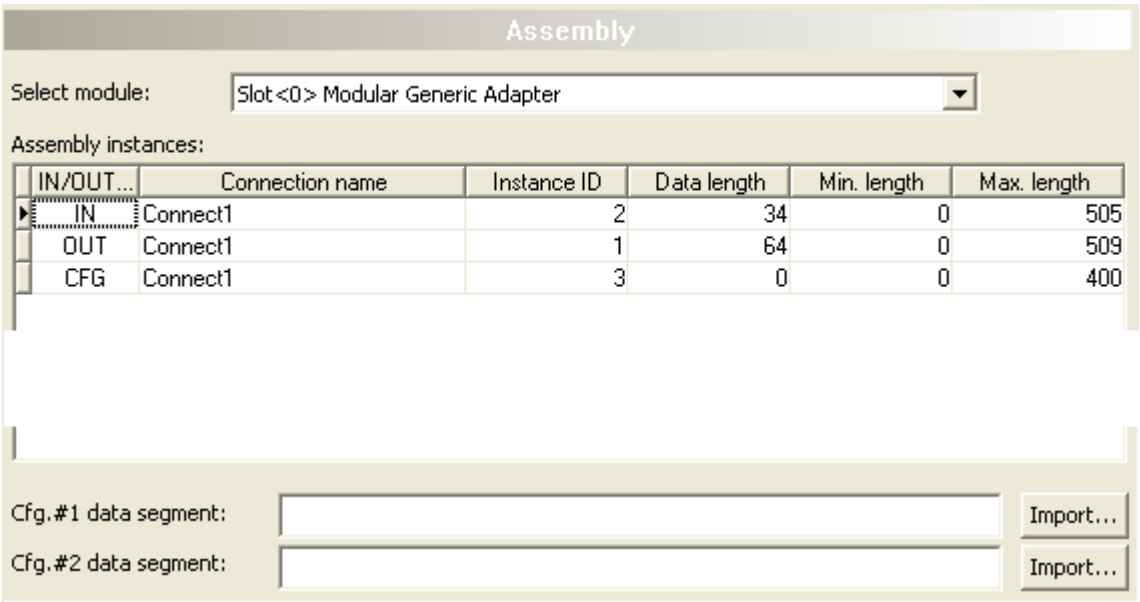


Figure 13: Configuration > Assembly (modular DTM)

© Hilscher, 2007-2017

Parameter	Meaning	Range of Value / Value
Select module (modular DTM only)	For modular generic Adapter a module must be selected to parameterize module's connection.	
IN/OUT/CFG	Input/Output/Config. connection point	
Connection name	Name of the connection configured to the EtherNet/IP Adapter	
Instance ID	Instance ID of the connection (editable)	1-255
Data length	Data length in Bytes (editable)	
Min. length (IN, OUT, CFG)	Minimum data length in Bytes	0
Max. length (IN, OUT)	Maximum data length in Bytes For the IO messaging there are 511 Bytes max., 9-Bit CIP Container Length, available. Apart from process IO data, the CIP message contains: -1- CIP Counter, 2 Bytes, always -2- Real-Time Header (32-bit Run/Idle Header) 4 Bytes, if configured (for each direction) -3- SizeAdder Bytes (according to module EDS) in modular Adapters, for T2O direction, if available. Hence, the **max. Assembly Instance data length** is calculated as follows: **Max_Data_Len** = 509 - sizeof(run_time_header) - sizeof (module_SizeAdder) 509 Bytes = 511 Bytes (Max. avail.) - 2 Bytes CIP Counter (always) sizeof(run_time_header) = 4 Bytes if configured, 0 Bytes else. sizeof(module_SizeAdder) = n Bytes (EDS) if available for T2O direction. For O2T it is always 0 Bytes.	
Max. length (CFG)	Maximum data length in Bytes	400
Cfg.#1 data segment	Configuration data segment 1 Config #1 is information that the adapter needs in order to be able to communicate with I/O modules on the backplane. To some extent it is configurable.	
Cfg.#2 data segment	Configuration data segment 2 Config #2 is user-configurable items such as safe state data, channel configuration, operating mode, etc. The I/O Module's EDS file parameterizes these items.	
Import …	The data segment configuration file can be imported.	

Table 15: Parameters Dialog Page Assembly

© Hilscher, 2007-2017

4 Configuring a modular EtherNet/IP Adapter

4.1 Requirements

In order to be able to configure a modular EtherNet/IP Adapter with the Modular Generic Adapter DTM, some EDS knowledge is required as the DTM does not come with an EDS parser for this purpose.

Along with the Chassis, the Adapter Module and the I/O Modules EDS files and printed data sheets for them are delivered.

Required Electronic Data Sheets are:

1. Chassis EDS
2. Adapter Module's EDS
3. I/O Modules' EDS

Required Knowledge:

- To the concept for Proxying
- To the implementation methods of modular devices with configurable parameters
- To the relationship between adapter module and I/O modules

If you're using modular devices with configurable parameters, you must be familiar with their implementation methods and understand configurable parameters relationship between the adapter module and the I/O modules.

© Hilscher, 2007-2017

4.1.1 Overview - Configuring Adapter Module and I/O Modules

In EtherNet/IP Modular Generic Adapter DTM the used Adapter Module and I/O Modules must be configured.

- **Configure Adapter Module**

1. In the Adapter Module EDS (Electrical Data Sheet) select a connection from the [Connection Manager] section.
2. Setting the Connection Parameters and the I/O and Configuration Assembly Information in the DTM.

➢ Use the connection parameters and the I/O and configuration assembly information from the Adapter Module's EDS, according to the settings in the selected connection.

3. Omitte step 2, if it is not necessary to configure the connection for the Adapter Module.

Or:

4. Deactivate an already configured connection to the Adapter Module in the Scanner DTM on **Scanlist** page.

- **Configure an I/O Module**

1. In the I/O Module's EDS select a connection from the [Connection Manager] section.
   Always selected connection: ProxiedConnectionN.
2. Setting the Connection Parameters and the I/O and Configuration Assembly Information in the DTM

➢ Use the connection parameters and I/O and configuration assembly information from the I/O Module's and the Adapter Module's EDS, according to the concept for Proxying for modular EDS files.

- **Parameters for Adapter and/or I/O Modules**

For parameters for the Adapter and/or the I/O Modules use EDS default settings or values in the given range. These settings are required for the O2T/T2O data length and/ or config. #1,#2 parts of the connections.

(O2T: Originator to Target; T2O: Target to Originator)

Note: EtherNet/IP Generic Adapter DTM doesn't support configuration using parameter objects or parameter object stubs embedded within the device!

Remark: For successfull configuring the device use the EtherNet/IP Modular Generic Adapter DTM and the devic's EDS file.

© Hilscher, 2007-2017

4.2 Configuration Steps for modular EtherNet/IP Devices

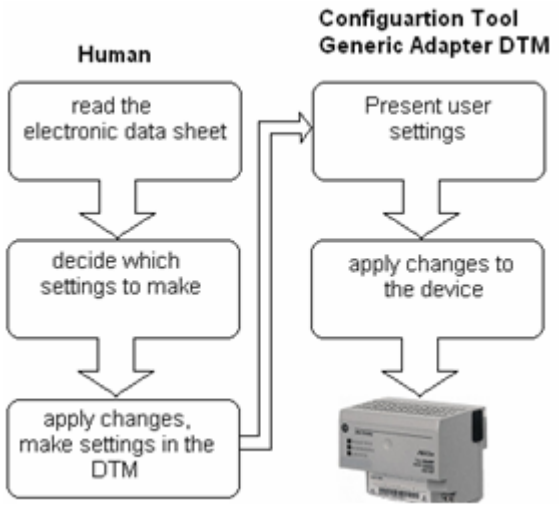


Figure 14: Parameterizing a modular EtherNet/IP Device

© Hilscher, 2007-2017

4.3 Modular EDS files

Modular EDS files are used for modular devices description. An EtherNet/IP modular Device consists of a network adapter known as adapter module with network connection and pluggable I/O modules. The latter can only be communicated using the network adapter module.

Each adapter and pluggable I/O, comes with its own module EDS file.

Chassis have their own EDS files. From the Chassis EDS file, the [Params] section is of interest and the value of **DefineSlotsInRack**, the number of slots.

The concept for Proxying has been introduced for pooling of distributed configuration information needed to get the whole device configured and running.

4.3.1 The Concept for Proxying

EtherNet/IP allows direct module connections. The Scanner device needs information how to get/set the data from/to a specific I/O module in the rack. As an Ethernet connection only is established to the network adapter module, the module connections have to be done via the adapter module. On the other hand, the adapter module does not have information to the characteristics of a pluggable I/O module. This information the Adapter Modlue gets from the Modular EDS files and with help of the concept for Proxying.

To describe the modular device's network characteristics, regarding available connections, the configuration parameters and assemblies, the adapter module's EDS file and the I/O modules' EDS file must be evaluated together.

The concept for Proxying allows distributed description of the configuration information, each module has its own EDS file.

The prefixes **Proxy** and **Proxied** are used to create a link between [Params]/[Assembly]/[Connection Manager] sections in different EDS files. The **Proxy** prefix is used in the adapter module's EDS file, the **Proxied** prefix in the I/O modules EDS file. So, for example, the "**ProxyConnect1**" entry in the adapter EDS file has it's corresponding "**ProxiedConnect1**" entry in the I/O modules EDS file. Zipping method is used to get complete Assembly, Param or Connect description. Combining "**ProxyConnect1**" with "**ProxiedConnect1**" results in a full description of the direct ‚**Connect1**' module connection available to the network.

© Hilscher, 2007-2017

4.3.2 Rehearse a Configuration by means of Genuine Modular EDS files

- **Example Adapter Data**

For demonstration purpose here serve the EDS files for 1794-AENT Adapter, 1794-OB16 and 1794-IB16 pluggable modules. For complete EDS listings, please, refer to the online resource
http://www.ab.com/networks/eds.html

Module	Product name	EDS file
1794-AENT	1794-AENT FLEX I/O Ethernet Adapter	0001000C005A0100.eds
1794-IB16	1794-16 Point 24V DC Input, Sink	0001000700220100.eds
1794-OB16	1794-16 Point 24V DC Output, Source	0001000700230100.eds

Table 16: Example Adapter Data

- **Assembling physical Device**

1. Assemble a device using the modules mentioned above in a compatible chassis, according to the Rack**N** entries in the [Modular] section in Modules' EDS files.

- **Guidance for Configuration**

2. Follow the descriptions given in the subsequent sections to perform the necessary configuration steps.

- **ConfigurationTask**

The configuation task is to configure connections to different modules in a rack.

- **Sample Modules**

As sample the two I/O modules and a network module mentioned above are used.

- **Used Connection**

The [Connection Manager] section of the adapter and I/O module EDS file specifies which connection from the available ones are to be used in the configuration.

© Hilscher, 2007-2017

4.3.3 Configuring Modules

- **Configuration Steps**

1. Setting the physical device configuration in the DTM:

The modules configuration set in the backplane must be configured in the DTM.

➢ In the DTM on the **Modules** page set the slot numbers, the module names and the module widths according to the configuration of the physical device, the Flex I/O backplane.

2. Setting slot, module width and module name:

The adapter EDS file indicates that the 1794-AENT module is plugged in slot 0. The physical device assembly in the sample here must include this.

➢ Set in the raw with slot 0 the values: **1** for the *Width* and "**1794-AENT FLEX I/O Ethernet Adapter**" or simply "**1794-AENT**" for the *Module name*.

Note: The module name isn't used in the communication. It is only used for convenience. Thus shorter denomination for a module can be used.

If the 1794-IB16/A module physically is plugged into slot 1:

➢ Set in the next raw *Slot* to **1**, *Width* to **1** and the *Module name* to "**1794 - 16 Point 24V DC Input, Sink**".

If the 1794-OB16/A module physically is plugged into slot 2:

➢ Set in the next raw *Slot* to **2**, *Width* to **1** and the *Module name* to "**1794 - 16 Point 24V DC Output, Source**".

© Hilscher, 2007-2017

4.3.4 Configuring 1794-IB16/A Flex module

4.3.4.1 Selecting Connection

For the 1794-IB16 module the EDS file entry pair "**ProxyConnect1**+**ProxiedConnect1**" has been chosen.

"**ProxyConnect1**" is described in the 1794-AENT Adapter EDS file:

```
ProxyConnect1 =
    0x04010002,    $ trigger & transport
                $  0-15  = supported transport classes (class 1)
                $  16    = cyclic (1 = supported)
                $  17    = change of state (0 = not supported)
                $  18    = on demand (0 = not supported)
                $  19-23 = reserved (must be zero)
                $  24-27 = exclusive owner
                $  28-30 = reserved (must be zero)
                $  31    = client 0 (don't care for classes 0 and 1)
    0x44240405,    $ point/multicast & priority & realtime format
                $  0     = O=>T fixed (1 = supported)
                $  1     = O=>T variable (0 = not supported)
                $  2     = T=>O fixed (1 = supported)
                $  3     = T=>O variable (0 = not supported)
                $  4-7   = reserved (must be zero)
                $  8-10  = O=>T header (4 byte run/idle)
                $  11    = reserved (must be zero)
                $  12-14 = T=>O header
                $  15    = reserved (must be zero)
                $  16-19 = O=>T point-to-point
                $  20-23 = T=>O multicast
                $  24-27 = O=>T scheduled
                $  28-31 = T=>O scheduled
    ,ProxyParam7,ProxyAssem3,  $ O=>T default,description
    ,ProxyParam1,Assem5,       $ T=>O default,description
    ,Assem3,                   $ config part 1 (dynamic assemblies)
    ,ProxyAssem5,              $ config part 2 (module configuration)
    "Exclusive Owner",         $ connection name
    "",                        $ Help string
    "01 SLOT_MINUS_ONE 20 04 24 03 2C 01 2C 02"; $ exclusive owner
path
```


© Hilscher, 2007-2017

"**ProxiedConnect1**" is described in the 1794-IB16 EDS file:

```
ProxiedConnect1 = 0x00000000,
                  0x00000000,
                  , , ,  $ O=>T
                  , , ,  $ T=>O
                  , ,    $ Config #1
                  , ,    $ Config #2
                  "",
                  "",
                  "";
```

4.3.4.2 Trigger & Transport Configuration

Note: Scanner firmware supports only **cyclic** Trigger Mode.

In the "**ProxyConnect1**" key definition the entry

```
0x04010002,    $ trigger & transport
```

defines in the bits 24-27 the transport type as `exclusive owner`, bit 26 is set.

- Set Transport Type to "exclusive owner".

For Trigger & Transport details, refer to [2], Chapter 7-3.6.9.1.

4.3.4.3 Connection Type Configuration

In the "**ProxyConnect1**" key definition the entry:

```
0x44240405,    $ point/multicast & priority & realtime format
```

defines in the bits 16-19, O2T connection type as POINT2POINT, bit 18 is set.

- In the DTM chose in the configuration **POINT2POINT** for the O2T connection.

The bits 20-23 define T2O connection type as MULTICAST, bit 21 is set.

- In the DTM set T2O connection type to **MULTICAST**.

For Connection Type details, refer to [2], Chapter 7-3.6.9.2.

© Hilscher, 2007-2017

4.3.4.4 Real-time Transfer Format Configuration

In the "**ProxyConnect1**" key definition the entry

```
0x44240405,     $ point/multicast & priority & realtime format
```

defines in the bits 8-10, the O2T header, value 100h = 4 bytes. The modules "**ProxiedConnect1**" key definition doesn't add any information to it.

- Configure **Run/Idle header** for the O2T connection.

In the bits 12-14, the T2O header, value 000h = 0 bytes is set. No additional info is available in "**ProxiedConnect1**".

- For the T2O set RTT format to "**connection is pure data and is modeless**".

For RTT Format details, refer to [2], Chapter 7-3.6.9.2.

4.3.4.5 Specifying Cfg.,O2T & T2O Instance IDs

Adapter EDS defines the connection path in **ProxyConnect1** as:

```
    "01 SLOT_MINUS_ONE 20 04 24 03 2C 01 2C 02"; $ exclusive
owner path
```

`20 04` is an Assembly Object Class, followed by the Instance for configuration `24 03`, Instance ID **3**, optional O2T Connection Point `2C 01`, Instance ID **1**, and optional T2O Connection Point `2C 02`, Instance ID **2**.

- Set these IDs for the selected module on the dialog page **Assembly** in the DTM.

For Module connections the connection path is extended with `SLOT_MINUS_ONE`, where the `SLOT` is the slot number of the module.

© Hilscher, 2007-2017

4.3.4.6 Specifying O2T & T2O Data Rate

RPI in this “**ProxyConnect1**” connection for both directions is not specified, the I/O module EDS does not add any further restrictions in “**ProxiedConnect1**” either. This means that the full range of RPI is available.

RPIs are set in the Scanner DTM, **Scanlist** page, for each connection separately.

4.3.4.7 Specifying O2T Data Size

In ProxyConnect1 there’s an entry:

```
,ProxyParam7,ProxyAssem3, $ O=>T default,description
```

Adapter EDS	I/O Module EDS
ProxyParam7 = 0, ,, 0x0000, 0xC7,**2**, "output size","","", Module,Module,Module, 0,0,0,0, ,,,, 0;	ProxiedParam7 = 0, ,, 0x0000, ,, "output size","","", **0**,**0**,**0**, ,,,, ,,,, ;

Table 17: Adapter and I/O Module EDS

The Adapter defines the channel size as 2 bytes, see “**ProxyParam7**”, but the I/O module’s EDS defines min, max, default values all zero, see ”**ProxiedParam7**”. This is a pure input module. The O2T data size (Output Assembly Instance size) is 0.

© Hilscher, 2007-2017

4.3.4.8 Specifying T2O Data Size

In ProxyConnect1 there's an entry:

```
,ProxyParam1,Assem5, $ T=>O default,description
```

Adapter EDS	I/O Module EDS
ProxyParam1 = 0, ,, 0x0000, 0xC7,**2**, "input size", "","", Module,Module,Module, 0,0,0,0,,,,, 0;	ProxiedParam1 = 0, ,, 0x0000, ,, "input size", "","", **2**,**4**,**4**, ,,,, ,,,, ;

Table 18: Adapter and I/O Module EDS

The range of the T2O data size is 2-4 bytes as given in "**ProxiedParam1**", module's EDS. Each channel is 2 bytes long given in "**ProxyParam1**", adapter's EDS, so it is possible to configure less then all 2 channels.

- Set the T2O data size (Input Assembly Instance size) to 2 or 4 bytes.

In the adapter EDS a *size adder* could be defined. "**ProxyParamSizeAdder**" keyword shall be used to provide minimum, maximum and default values to be added to the "**ProxyParam**" minimum, maximum and default values. "**ProxyParamSizeAdderN**" shall be combined with the corresponding "**ProxyParamN**" entry. The "**ProxyParamSizeAdder**" keyword provides a means for an adapter on a module connection ("**ProxyConnect**") to add adapter data to the module data and return the combined data on the connection. For more details, refer to [1], chapter 7-3.7.2.2.

In the adapter EDS *size adder* is defined - 4 bytes minimum, 4 bytes maximum and 4 bytes default *size adder*.

- Use always the default one, marked red in this EDS snippet.
- In the DTM on the **Connection** page set T2O *size adder* exactly to 4 Bytes.

```
ProxyParam1 = . . . ;
ProxyParamSizeAdder1 = 4,4,4;
```

The Adapter configuration must indicate to the Scanner that the module input data will consist not only of configured 2 input bytes, but there will be additional 4 bytes the adapter adds to the input packet.

- In the **Connection** Page set T2O *Size Adder* to 4.

The Input Data Section (Packet) is 8 bytes long as 2 bytes input data, 4 bytes size adder and 2 bytes CIP sequence counter are available.

© Hilscher, 2007-2017

4.3.4.9 Specifying Cfg.#1,#2

In "**ProxyConnect1**" **Assem3** defines the format of Cfg.#1.

…

```
,Assem3,    $ config part 1 (dynamic assemblies)
```

…

Here a closer look at the **Assem3** definition:

```
Assem3 = "private config",,,,,,
      16,Param29,    $ configuration revision
      16,0x0002,     $ size of output data
      16,Param30,    $ output assembly, width
      16,0x0c7d,     $ output assembly, class/attribute
      16,0x0004,     $ size of input data
      16,Param31,    $ input assembly, width
      16,0x0a7d,     $ input assembly, class/attribute
      16,Param32,    $ status assembly, width
      16,0x0b7d,     $ status assembly, class/attribute
      16,Param33,    $ size of config data
      ,ExternalID,   $ module key
      16,ProxyAssem7,$ idle action, fault action, automatch,
                     fault from idle, hold last input,
                     not used
      16,ProxyParam3,   $ input class size
      16,ProxyParam4,   $ status class size
      16,ProxyParam5,   $ output class size
      16,ProxyParam6;   $ config class size


Cfg Data Segment is made up of


- cfg. revision –
- Output Data Section –
- Input (+status) Data Section –
- Cfg. Data Section –
```

For the decoding of Assem3 entries, consult [2], Chapter 7: Electronic Data ***Sheets", chapter "*** 7-3.6.7.2.7 Member Size/Member Reference Fields".

© Hilscher, 2007-2017

Cfg.#1 is composed of:

```
01 00 Configuration Revision (Param29)

02 00 Size of Output Data Section (in 16-bit words) ::next 2 words
00 00 Output Assembly Width (bits), pure input module
7D 0C Output Assembly Class/Attrib

04 00 Size of Input Data Section (in 16-bit words)  ::next 4 words
10 00 Input Assembly Width (T->O (Input) data size in bits = 16)
7D 0A Input Assembly Class/Attrib
30 00 Status Assembly Width (Param32+SizeAdder = 16+32 = 48* bits)
7D 0B Status Assembly Class/Attrib

07 00 Size of Cfg data (Param33 = Cfg Data Size)    ::next 7 words
81 02 Module Key:      (From EDS=ExternalID=ProxiedParam2)
00 00 Module Flags:    (ProxyAssem7 = Module Flag Data)
01 00 Input Class Size (ProxyParam3 = Module Input Size  = 1 words)
01 00 Status Class Size(ProxyParam4 = Module Status Size = 1 words)
00 00 Output Class Size(ProxyParam5 = Module Output Size = 0 words)
01 00 Config Class Size(ProxyParam6 = Module Config Size = 1 words)
00 00 Cfg.#2, module cfg
```

* - 48 bits of status assembly width are made up of 2 bytes given in Param32 and 4 additional bytes of size adder.

Cfg.#1 data segment value should be set to “**0100 0200 0000 7D0C 0400 1000 7D0A 3000 7D0B 0700 8102 0000 0100 0100 0000**”.

© Hilscher, 2007-2017

The Cfg.#2 is defined as:

```
,ProxyAssem5,     $ config part 2 (module configuration)
```

Adapter EDS	I/O Module EDS
ProxyAssem5 = "module config and safe state", ,, Module,, , ,ProxyAssem1, ,ProxyAssem6;	ProxiedAssem5 = "module config and safe state",, ,0x0000,,;
ProxyAssem1 = "module config", "20 7D 24 SLOT 30 0D", ,Module,,, ModuleMemberList;	ProxiedAssem1 = "module config",, ,0x0000,,, 3, Param1, 3, Param2, 6,, 1, Param3, 1, Param4, 2,;
ProxyAssem6 = "safe state", "20 7D 24 SLOT 30 0E", ,,,, ModuleMemberList;	ProxiedAssem6 = "safe state",, ,,,;

Table 19: Adapter and I/O Module EDS

ProxyAssem1 is “module config”, which is 16 bits long. Consult 1794-IB16/A users manual to know the details.

```
Param1 =
  0,          $ first field shall equal 0
  ,,          $ path size,path
  0x0000,     $ descriptor - read only
  2,          $ data type : 16-bit Unsigned Integer
  2,          $ data size in bytes
  "Points 0-11 Input Filter Time", $ name
  "",            $ units
  "",            $ help string
  0,7,0,         $ min,max,default data values
  0,0,0,0,    $ mult,dev,base,offset scaling not used
  0,0,0,0,    $ mult,dev,base,offset link not used
  0;          $ decimal places not used
```


© Hilscher, 2007-2017

```
      Enum1 =
        0,"On->Off/Off->On= 0.25 ms",
        1,"On->Off/Off->On= 0.5 ms",
        2,"On->Off/Off->On= 1 ms",
        3,"On->Off/Off->On= 2 ms",
  4,"On->Off/Off->On= 4 ms",
  5,"On->Off/Off->On= 8 ms",
  6,"On->Off/Off->On= 16 ms",
  7,"On->Off/Off->On= 32 ms";
```

- Set for these 3 bits EDS default 0, marked red.
- Then `"Points 0-11 Input Filter Time"` is set to `0.25ms`.

```
Param2 = 0,          $ first field shall equal 0
        ,,          $ path size,path
        0x0000,     $ descriptor - read only
        2,          $ data type : 16-bit Unsigned Integer
        2,          $ data size in bytes
        "Points 12-15 Input Filter Time", $ name
        "",            $ units
        "",            $ help string
        0,7,0,         $ min,max,default data values
        0,0,0,0,    $ mult,dev,base,offset scaling not used
        0,0,0,0,    $ mult,dev,base,offset link not used
        0;          $ decimal places not used
      Enum2 =
        0,"On->Off/Off->On= 0.25 ms",
        1,"On->Off/Off->On= 0.5 ms",
        2,"On->Off/Off->On= 1 ms",
        3,"On->Off/Off->On= 2 ms",
  4,"On->Off/Off->On= 4 ms",
  5,"On->Off/Off->On= 8 ms",
  6,"On->Off/Off->On= 16 ms",
  7,"On->Off/Off->On= 32 ms";
```

- Set for these 3 bits EDS default 0, marked red.
- Then `"Points 12-15 Input Filter Time"` is set to `0.25ms`.

© Hilscher, 2007-2017

The next 6 bits aren't specified.

➢ Set next 6 bits to 0.

```
Param3 =
  0,          $ first field shall equal 0
  ,,          $ path size,path
  0x0000,     $ descriptor - read only
  4,          $ data type : 8-bit Unsigned Integer
  1,          $ data size in bytes
  "Counter Enable/Disable", $ name
  "",            $ units
  "",            $ help string
  0,1,0,         $ min,max,default data values
  0,0,0,0,    $ mult,dev,base,offset scaling not used
  0,0,0,0,    $ mult,dev,base,offset link not used
  0;          $ decimal places not used
Enum3 =
  0,"Enabled",
  1,"Disabled";
```

➢ Set the following 1 bit to default 0, marked red.

➢ Then `"Counter Enable/Disable"` is set to `„Enabled"`.

```
Param4 =
  0,          $ first field shall equal 0
  ,,          $ path size,path
  0x0000,     $ descriptor - read only
  4,          $ data type : 8-bit Unsigned Integer
  1,          $ data size in bytes
  "Filter Enable/Disable", $ name
  "",            $ units
  "",            $ help string
  0,1,0,         $ min,max,default data values
  0,0,0,0,    $ mult,dev,base,offset scaling not used
  0,0,0,0,    $ mult,dev,base,offset link not used
  0;          $ decimal places not used
Enum4 =
  0,"Enabled",
  1,"Disabled";
```


© Hilscher, 2007-2017

- Set the following 1 bit to default 0, marked red.
- Then `"Filter Enable/Disable"` is set to `"Enabled"`.

The last 2 bits of the 16 bit word aren't specified, we set them to zero.

- Set the last 2 bits of the 16 bit word to 0.

Note: `ProxyAssem6` is module's "safe state" data, which is empty. Do not configure it.

- Set *module config.* 16 bits as follows: …

```
00 00 Cfg.#2, module cfg
```

(in the example here).

- Set the Cfg.#2 data segment value to "**0100 0000**".

© Hilscher, 2007-2017

4.3.5 Configuring 1794-OB16/A Flex module

4.3.5.1 Selecting Connection

For the 1794-OB16 module the EDS entry pair "**ProxyConnect1**+**ProxiedConnect1**" has been chosen.

"**ProxyConnect1**" is described in the 1794-AENT Adapter EDS, see corresponding listing (EDS snippet) above, in the section for the 1794-IB16 module.

"**ProxiedConnect1**" is described in the 1794-OB16 Adapter EDS file:

```
ProxiedConnect1 = 0x00000000,
                  0x00000000,
                  , , , $ O=>T
                  , , , $ T=>O
                  , ,   $ Config #1
                  , ,   $ Config #2
                  "",
                  "",
                  "";
```

4.3.5.2 Trigger & Transport Configuration

Note: Scanner firmware supports only **cyclic** Trigger Mode.

In the "**ProxyConnect1**" key definition the entry

```
0x04010002,    $ trigger & transport
```

defines in the bits 24-27 the transport type as `exclusive owner`, bit 26 is set.

➢ Set the Transport Type to `"exclusive owner"`.

For Trigger & Transport details, refer to [2], Chapter 7-3.6.9.1.

© Hilscher, 2007-2017

4.3.5.3 Connection Type Configuration

In the "**ProxyConnect1**" key definition the entry

```
0x44240405,     $ point/multicast & priority & realtime format
```

defines in the bits 16-19, O2T connection type as POINT2POINT, bit 18 is set.

- So in the DTM chose in the configuration **POINT2POINT** for the O2T connection.

The bits 20-23 define T2O connection type as MULTICAST, bit 21 is set.

- In the DTM set T2O connection type to **MULTICAST**.

For Connection Type details, refer to [2], Chapter 7-3.6.9.2.

4.3.5.4 Real-time Transfer Format Configuration

In the "**ProxyConnect1**" key definition the entry

```
0x44240405,     $ point/multicast & priority & realtime format
```

defines in the bits 8-10, the O2T header, value 100h = 4 bytes. The modules "**ProxiedConnect1**" key definition doesn't add any information to it.

- Configure **Run/Idle header** for the O2T connection.

In the bits 12-14, T2O header, value 000h = 0 bytes is set and no additional info is available in "**ProxiedConnect1**".

- So for the T2O connection set RTT format to "**connection is pure data and is modeless**".

For RTT Format details, refer to [2], Chapter 7-3.6.9.2.

4.3.5.5 Specifying Cfg.,O2T & T2O Instance IDs

The Adapter EDS defines the connection path in **ProxyConnect1** as:

```
    "01 SLOT_MINUS_ONE 20 04 24 03 2C 01 2C 02"; $ exclusive
owner path
```

`20 04` is an Assembly Object Class, followed by the Instance for configuration `24 03`, Instance ID **3**, optional O2T Connection Point `2C 01`, Instance ID **1**, and optional T2O Connection Point `2C 02`, Instance ID **2**.

- Set these IDs for the selected module on the dialog page **Assembly** in the DTM.

For Module connections the connection path is extended with `SLOT_MINUS_ONE`, where the `SLOT` is the slot number of the module.

© Hilscher, 2007-2017

4.3.5.6 Specifying O2T & T2O Data Rate

RPI in this "**ProxyConnect1**" connection for both directions is not specified. The I/O module EDS does not add any further restrictions in "**ProxiedConnect1**" either. This means that the full range of RPI is available.

RPIs are set in the Scanner DTM, **Scanlist** page, for each connection separately.

4.3.5.7 Specifying O2T Data Size

In **ProxyConnect1** there's an entry

```
,ProxyParam7,ProxyAssem3, $ O=>T default,description
```

Adapter EDS	I/O Module EDS
ProxyParam7 = 0, ,, 0x0000, 0xC7,**2**, "output size","","", Module,Module,Module, 0,0,0,0, ,,,, 0;	ProxiedParam7 = 0, ,, 0x0000, ,, "output size","","", **2**,**2**,**2**, ,,,, ,,,, ;

Table 20: Adapter and I/O Module EDS

"**ProxyParam7**"+"**ProxiedParam7**" defines the O2T data size as exactly 2 bytes. There's only one channel and it is 2 bytes long.

➢ Set the O2T data size (Output Assembly Instance size) to 2.

The ForwardOpen Message O2T data size (packet size) is larger than the I/O data size because the O2T packet always includes the CIP Sequence Count (2 bytes) and if configured, the 32-bit Run/Idle Header (4 bytes).

The ForwardOpen Message O2T data size so could be 4 or 8 bytes depending on run/idle header configuration.

In this example here the ForwardOpen Message O2T data size is 8 bytes as the Run/Idle header (4 bytes) is used.

© Hilscher, 2007-2017

4.3.5.8 Specifying T2O Data Size

In **ProxyConnect1** there's an entry:

```
,ProxyParam1,Assem5, $ T=>O default,description
```

Adapter EDS	I/O Module EDS
ProxyParam1 = 0, ,, 0x0000, 0xC7,**2**, "input size", "","", Module,Module,Module, 0,0,0,0,,,,, 0;	ProxiedParam1 = 0, ,, 0x0000, ,, "input size", "","", 2,2,2, ,,,, ,,,, ;

Table 21: Adapter and I/O Module EDS

The range of the T2O data size is defined in "**ProxiedParam1**" as exactly 2 bytes. So there's only one channel that is 2 bytes long, see "**ProxyParam1**".

In the adapter EDS file *size adder* is defined - 4 bytes minimum, 4 bytes maximum and 4 bytes default, see "**ProxyParamSizeAdder1**".

- Use always the default value, marked red in this EDS snippet.
- In the DTM on the **Connection** page set T2O *size adder* exactly to 4 Bytes.

```
ProxyParam1 = . . . ;
ProxyParamSizeAdder1 = 4,4,4;
```

The Adapter configuration should indicate to the Scanner that the module input data will consist not only of configured 2 input bytes, but there will be additional 4 bytes the adapter adds to the input packet.

- In the **Connection** Page set T2O *Size Adder* to 4.

The Input Data Section (Packet) is 8 bytes long as 2 bytes input data, 4 bytes size adder and 2 bytes CIP sequence counter are available.

© Hilscher, 2007-2017

4.3.5.9 Specifying Cfg.#1,#2

In "**ProxyConnect1**" Assem3 defines the format of Cfg.#1.

…

```
,Assem3,                    $ config part 1 (dynamic assemblies)
```

…

The Cfg Data Segment sections have been described above. A listing for Assem3 is also available above, see section *Specifying Cfg.#1,#2* on page 39.

Cfg.#1 is composed of:

```
01 00 Configuration Revision (Param29)

02 00 Size of Output Data Section (in 16-bit words)    ::next 2 words
10 00 Output Assembly Width(bits) (Param30=O->T(Output) data size: 16 bits)
7D 0C Output Assembly Class/Attrib (class/attrib ids of data consumer)

04 00 Size of Input Data Section (in 16-bit words)     ::next 4 words
00 00 Input Assembly Width (Param31 = T->O (Input) data size: 0 bits)
7D 0A Input Assembly Class/Attrib (class/attrib ids of data producer)
30 00 Status Assembly Width (Param32+SizeAdder = 16 bits + 32 bits = 48)
7D 0B Status Assembly Class/Attrib (class/attrib ids of data producer)

08 00 Size of Config data (Param33 = Config Data Size) ::next 8 words
91 01 Module Key:         (From EDS=ExternalID=ProxiedParam2)
20 00 Module Flags:       (ProxyAssem7 = Module Flag Data)
00 00 Input Class Size    (ProxyParam3 = Module Input Size  = 0 words)
01 00 Status Class Size   (ProxyParam4 = Module Status Size = 1 words)
01 00 Output Class Size   (ProxyParam5 = Module Output Size = 1 words)
02 00 Config Class Size   (ProxyParam6 = Module Config Size = 2 words)
00 00 Cfg.#2, module cfg
00 00 Cfg.#2, module flags
```

- Set Cfg.#1 data segment value to "**0100 0200 1000 7D0C 0400 0000 7D0A 3000 7D0B 0800 9101 2000 0000 0100 0100**".

© Hilscher, 2007-2017

The Cfg.#2 is defined as

```
,ProxyAssem5,     $ config part 2 (module configuration)
```

Adapter EDS	I/O Module EDS
ProxyAssem5 = "module config and safe state", ,, Module,, , ,ProxyAssem1, ,ProxyAssem6;	ProxiedAssem5 = "module config and safe state", ,, ,,;
ProxyAssem1 = "module config", "20 7D 24 SLOT 30 0D", ,Module,,, ModuleMemberList;	ProxiedAssem1 = "module config",, ,0x0001,,, 16,;
ProxyAssem6 = "safe state", "20 7D 24 SLOT 30 0E", ,,,, ModuleMemberList;	ProxiedAssem6 = "safe state", , ,,,, 1, Param19, 1, Param20, 1, Param21, 1, Param22, 1, Param23, 1, Param24, 1, Param25, 1, Param26, 1, Param27, 1, Param28, 1, Param29, 1, Param30, 1, Param31, 1, Param32, 1, Param33, 1, Param34;

Table 22: Adapter and I/O Module EDS

© Hilscher, 2007-2017

ProxyAssem1 is “module config”, which is 16 bits long. It is not specified in detail.

- Set “module config” to 0.

ProxyAssem6 is module’s “safe state” data. It consists of the same size, specified as a bit-field of 16 Bits. For all these bits we take default settings, they’re all zeros. That does mean that "Point ‘i’ Safe State On/Off" default setting is “off”.

Consult 1794-OB16/A users manual for the details.

- Set in Cfg.#2 *module config*. 16 Bits and *module safe state* 16 Bits as follows:

```
…
00 00 Cfg.#2, module cfg
00 00 Cfg.#2, module flags
```

- In the example here the Cfg.#2 data segment value then is “**0200 0000 0000**”.

© Hilscher, 2007-2017

5 Appendix

5.1 User Rights

User rights are set within the FDT-container. Depending on the level the configuration is accessible by the user or read-only.

To access the **Configuration** panes of the EtherNet/IP Generic Adapter DTM you do not need special user rights. Also all users can select the decimal or hexadecimal Display mode or sort table entries.

Note: To edit, set or configure the parameters of the **Configuration** panes, you need user rights for *Maintenance,* for *Planning Engineer* or for *Administrator*.

The following tables give an overview of the user right groups and which user rights you need to configure the single parameters.

5.1.1 Configuration

	Observer	Operator	Maintenance	Planning Engineer	Administrator
Configuration	D	D	X	X	X
General	D	D	X	X	X
Modules (modular DTM)	D	D	X	X	X
Electronic Keying	D	D	X	X	X
Connection	D	D	X	X	X
Assembly	D	D	X	X	X

Table 23: User Rights Configuration (D = Displaying, X = Editing, Configuring)

5.2 References

[1] Device Type Manager (DTM) Style Guide, Version 1.0 ; FDT-JIG - Order No. <0001-0008-000>

[2] EtherNetIP Adapter Protocol API Manual, Revision 12, Hilscher GmbH 2013

 © Hilscher, 2007-2017

5.3 List of Figures

5.4 List of Tables

© Hilscher, 2007-2017

5.5 Glossary

Assembly

Connection Point

CIP

Common Industrial Protocol (Control and Information Protocol)

DHCP

Dynamic Host Configuration Protocol

DNS

Domain Name Service.

DTM

Device Type Manager.

The Device Type Manager (DTM) is a software module with graphical user interface for the configuration or for diagnosis of device.

EDS

An Electronic Data Sheet (EDS) provides information necessary to access and alter the configurable parameters of a device. An Electronic Data Sheet (EDS) is an external file that contains information for the device.

EtherNet/IP

EtherNet/Industrial Protocol (CIP on Ethernet)

EtherNet/IP Scanner

A Scanner exchanges real-time I/O data with Adapters and Scanners. This type of node can respond to connection requests and can also initiate connections on its own.

EtherNet/IP-Adapter

An Adapter emulates functions provided by traditional rack-adapter products. This type of node exchanges real-time I/O data with a Scanner Class product. It does not initiate connections on its own.

FDT

Field Device Tool

FDT specifies an interface, in order to be able to use DTM (Device Type Manager) in different applications of different manufacturers.

TCP/IP

Transmission Control Protocol / Internet Protocol

UDP

User Datagram Protocol

© Hilscher, 2007-2017

5.6 Contacts

Headquarters

Germany
Hilscher Gesellschaft für
Systemautomation mbH
Rheinstrasse 15
65795 Hattersheim
Phone: +49 (0) 6190 9907-0
Fax: +49 (0) 6190 9907-50
E-Mail: info@hilscher.com

Support
Phone: +49 (0) 6190 9907-99
E-Mail: de.support@hilscher.com

Subsidiaries

China
Hilscher Systemautomation (Shanghai) Co. Ltd.
200010 Shanghai
Phone: +86 (0) 21-6355-5161
E-Mail: info@hilscher.cn

Support
Phone: +86 (0) 21-6355-5161
E-Mail: cn.support@hilscher.com

France
Hilscher France S.a.r.l.
69500 Bron
Phone: +33 (0) 4 72 37 98 40
E-Mail: info@hilscher.fr

Support
Phone: +33 (0) 4 72 37 98 40
E-Mail: fr.support@hilscher.com

India
Hilscher India Pvt. Ltd.
Pune, Delhi, Mumbai
Phone: +91 8888 750 777
E-Mail: info@hilscher.in

Italy
Hilscher Italia S.r.l.
20090 Vimodrone (MI)
Phone: +39 02 25007068
E-Mail: info@hilscher.it

Support
Phone: +39 02 25007068
E-Mail: it.support@hilscher.com

Japan
Hilscher Japan KK
Tokyo, 160-0022
Phone: +81 (0) 3-5362-0521
E-Mail: info@hilscher.jp

Support
Phone: +81 (0) 3-5362-0521
E-Mail: jp.support@hilscher.com

Korea
Hilscher Korea Inc.
Seongnam, Gyeonggi, 463-400
Phone: +82 (0) 31-789-3715
E-Mail: info@hilscher.kr

Switzerland
Hilscher Swiss GmbH
4500 Solothurn
Phone: +41 (0) 32 623 6633
E-Mail: info@hilscher.ch

Support
Phone: +49 (0) 6190 9907-99
E-Mail: ch.support@hilscher.com

USA
Hilscher North America, Inc.
Lisle, IL 60532
Phone: +1 630-505-5301
E-Mail: info@hilscher.us

Support
Phone: +1 630-505-5301
E-Mail: us.support@hilscher.com

© Hilscher, 2007-2017