



**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**

**Hilscher Gesellschaft für Systemautomation mbH**

**[www.hilscher.com](http://www.hilscher.com)**

DOC070203OI07EN | Revision 7 | English | 2010-10 | Released | Public

# Table of Contents

1	INTRODUCTION.....	4
1.1	About this Manual .....	4
1.1.1	Descriptions of the Dialog Panes .....	4
1.1.2	Online Help .....	4
1.1.3	List of Revisions .....	5
1.1.4	Conventions in this Manual .....	6
1.2	Legal Notes.....	7
1.2.1	Copyright .....	7
1.2.2	Important Notes .....	7
1.2.3	Exclusion of Liability .....	8
1.2.4	Warranty .....	8
1.2.5	Export Regulations .....	9
1.2.6	Registered Trademarks.....	9
1.3	About EtherNet/IP Generic Adapter DTM.....	10
1.3.1	Requirements .....	10
1.4	Dialog Structure of the EtherNet/IP Generic Adapter DTM .....	11
1.4.1	General Device Information.....	12
1.4.2	Navigation Area .....	12
1.4.3	Dialog Panes .....	13
1.4.4	OK, Cancel, Apply and Help.....	13
1.4.5	Status Bar .....	14
2	GETTING STARTED AND INSTRUCTIONS STEP BY STEP .....	15
2.1	Overview Configuration Steps .....	15
2.2	Configuring Parameters of the Adapter Device .....	16
3	CONFIGURATION .....	17
3.1	Overview Configuration .....	17
3.2	General .....	18
3.3	Modules (modular DTM) .....	19
3.4	Electronic Keying .....	20
3.5	Connection.....	22
3.6	Assembly .....	26
4	CONFIGURING A MODULAR ETHERNET/IP ADAPTER .....	28
4.1	Requirements .....	28
4.1.1	Overview - Configuring Adapter Module and I/O Modules.....	29
4.2	Configuration Steps for modular EtherNet/IP Devices.....	30
4.3	Modular EDS files .....	31
4.3.1	The Concept for Proxying.....	31

4.3.2	Rehearse a Configuration by means of Genuine Modular EDS files .....	32
4.3.3	Configuring Modules .....	33
4.3.4	Configuring 1794-IB16/A Flex module .....	34
4.3.5	Configuring 1794-OB16/A Flex module .....	45
5	LISTS .....	52
5.1	List of Figures .....	52
5.2	List of Tables .....	52
6	GLOSSARY .....	53
7	APPENDIX .....	55
7.1	User Rights .....	55
7.1.1	Configuration .....	55
7.2	References .....	55
7.3	Contacts .....	56

# 1 Introduction

## 1.1 About this Manual

This manual describes how to use the EtherNet/IP generic Adapter DTM to configure within a FDT Framework the device parameters of an EtherNet/IP Adapter device, which is described with an EDS file. 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
<i>Configuration</i>	<i>Overview Configuration</i>	17
	<i>General</i>	18
	<i>Modules (modular DTM) (for modular DTM only)</i>	19
	<i>Electronic Keying</i>	20
	<i>Connection</i>	22
	<i>Assembly</i>	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

In- dex	Datum	Version	Component	Chapter	Revisions
1	21.02.07	1.0.0.1, 1.0.0.0	ENIPGenericAdapterDTM.dll, ENIPGenericAdapterGUI.ocx	alle	created
2	24.05.07	0.9.2.0, 0.9.2.0	ENIPGenericAdapterDTM.dll, ENIPGenericAdapterGUI.ocx	3.5 3.6	Section <i>Connection</i> revised Section <i>Assembly</i> revised
3	10.07.07	0.9.2.0, 0.9.2.0	ENIPGenericAdapterDTM.dll, ENIPGenericAdapterGUI.ocx	4	revised
4	08.08.07	0.9.2.1, 0.9.2.1	ENIPGenericAdapterDTM.dll, ENIPGenericAdapterGUI.ocx	2 5 5.2	Chapter ‚GETTING STARTED‘ added, Chapter ‚CONFIGURING A MODULAR ETHERNET/IP ADAPTER‘ added, Section ‚Contacts‘ actualized
5	16.04.08	1.0.0.1, 1.0.0.0	ENIPGenericAdapterDTM.dll, ENIPGenericAdapterGUI.ocx	1 4.3.4.7	Chapter ‚Introduction‘ actualized. Section ‚Specifying O2T Data Size‘ adapted.
6	29.10.08	1.0.2.0, 1.0.2.0	ENIPGenericAdapterDTM.dll, ENIPGenericAdapterGUI.ocx	1 2	Chapter ‚Introduction‘ revised, Chapter ‚GETTING STARTED‘ revised
7	27.10.10	1.5.x.x, 1.5.x.x	ENIPGenericAdapterDTM.dll, ENIPGenericAdapterGUI.ocx	1.4.5, 3.5, 3.6	Section ‚Status Bar‘ actualized, Section <i>Connection</i> – connection type NULL omitted, Abschnitt <i>Assembly</i> Import of the data segment configuration added.

Table 2: List of Revisions

## 1.1.4 Conventions in this Manual

Operation instructions, a result of an operation step or notes are marked as follows:

### **Operation Instructions:**

➤ <instruction>

Or

1. <instruction>

2. <instruction>

### **Results:**

↪ <result>

### **Notes:**



**Important:** <important note>

---



**Note:** <note>

---



<note, were to find further information>

---

### **Designation "Scanner" and "Adapter"**



**Note:** The EtherNet/IP specification defines the designation "Scanner" instead of "Master" and "Adapter" instead of "Slave".

---

## 1.2 Legal Notes

### 1.2.1 Copyright

© 2008-2010 Hilscher Gesellschaft für Systemautomation mbH

All rights reserved.

The images, photographs and texts in the accompanying material (user manual, accompanying texts, documentation, etc.) are protected by German and international copyright law as well as international trade and protection provisions. You are not authorized to duplicate these in whole or in part using technical or mechanical methods (printing, photocopying or other methods), to manipulate or transfer using electronic systems without prior written consent. You are not permitted to make changes to copyright notices, markings, trademarks or ownership declarations. The included diagrams do not take the patent situation into account. The company names and product descriptions included in this document may be trademarks or brands of the respective owners and may be trademarked or patented. Any form of further use requires the explicit consent of the respective rights owner.

### 1.2.2 Important Notes

The user manual, accompanying texts and the documentation were created for the use of the products by qualified experts, however, errors cannot be ruled out. For this reason, no guarantee can be made and neither juristic responsibility for erroneous information nor any liability can be assumed. Descriptions, accompanying texts and documentation included in the user manual do not present a guarantee nor any information about proper use as stipulated in the contract or a warranted feature. It cannot be ruled out that the user manual, the accompanying texts and the documentation do not correspond exactly to the described features, standards or other data of the delivered product. No warranty or guarantee regarding the correctness or accuracy of the information is assumed.

We reserve the right to change our products and their specification as well as related user manuals, accompanying texts and documentation at all times and without advance notice, without obligation to report the change. Changes will be included in future manuals and do not constitute any obligations. There is no entitlement to revisions of delivered documents. The manual delivered with the product applies.

Hilscher Gesellschaft für Systemautomation mbH is not liable under any circumstances for direct, indirect, incidental or follow-on damage or loss of earnings resulting from the use of the information contained in this publication.

### 1.2.3 Exclusion of Liability

The software was produced and tested with utmost care by Hilscher Gesellschaft für Systemautomation mbH and is made available as is. No warranty can be assumed for the performance and flawlessness of the software for all usage conditions and cases and for the results produced when utilized by the user. Liability for any damages that may result from the use of the hardware or software or related documents, is limited to cases of intent or grossly negligent violation of significant contractual obligations. Indemnity claims for the violation of significant contractual obligations are limited to damages that are foreseeable and typical for this type of contract.

It is strictly prohibited to use the software in the following areas:

- for military purposes or in weapon systems;
- for the design, construction, maintenance or operation of nuclear facilities;
- in air traffic control systems, air traffic or air traffic communication systems;
- in life support systems;
- in systems in which failures in the software could lead to personal injury or injuries leading to death.

We inform you that the software was not developed for use in dangerous environments requiring fail-proof control mechanisms. Use of the software in such an environment occurs at your own risk. No liability is assumed for damages or losses due to unauthorized use.

### 1.2.4 Warranty

Although the hardware and software was developed with utmost care and tested intensively, Hilscher Gesellschaft für Systemautomation mbH does not guarantee its suitability for any purpose not confirmed in writing. It cannot be guaranteed that the hardware and software will meet your requirements, that the use of the software operates without interruption and that the software is free of errors. No guarantee is made regarding infringements, violations of patents, rights of ownership or the freedom from interference by third parties. No additional guarantees or assurances are made regarding marketability, freedom of defect of title, integration or usability for certain purposes unless they are required in accordance with the law and cannot be limited. Warranty claims are limited to the right to claim rectification.

## 1.2.5 Export Regulations

The delivered product (including the technical data) is subject to export or import laws as well as the associated regulations of different countries, in particular those of Germany and the USA. The software may not be exported to countries where this is prohibited by the United States Export Administration Act and its additional provisions. You are obligated to comply with the regulations at your personal responsibility. We wish to inform you that you may require permission from state authorities to export, re-export or import the product.

## 1.2.6 Registered Trademarks

Windows® 2000 and Windows® XP 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.

## 1.3 About EtherNet/IP Generic Adapter DTM

You can use the EtherNet/IP generic Adapter DTM to configure within a FDT Framework the device parameters of an EtherNet/IP Adapter device, which is described with an EDS file. Modular EtherNet/IP Devices are configured using the modular EtherNet/IP generic Adapter DTM.

To perform the configuration insert the EtherNet/IP generic Adapter DTM in the network project to the Master busline of the EtherNet/IP Scanner DTM.

### 1.3.1 Requirements

#### **System Requirements**

- Windows® 2000 and Windows® XP
- CD ROM drive
- Graphic resolution: min. 1024 x 768 pixel
- Keyboard and Mouse

#### **Requirements EtherNet/IP Generic Adapter DTM**

To configure an EtherNet/IP Adapter device with the EtherNet/IP Generic Adapter DTM or an modular EtherNet/IP Adapter device with the modular EtherNet/IP Generic Adapter DTM the following requirements must be fulfilled:

- Installed FDT/DTM V 1.2 compliant frame application,
- Installed EtherNet/IP Scanner DTM,
- EDS file of the devices to configure (for future application). Currently the parameters must be set manually.
- Loaded DTM in the Device Catalog of the FTD Framework. E. g. the user needs to reload the Device Catalog.

## 1.4 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.

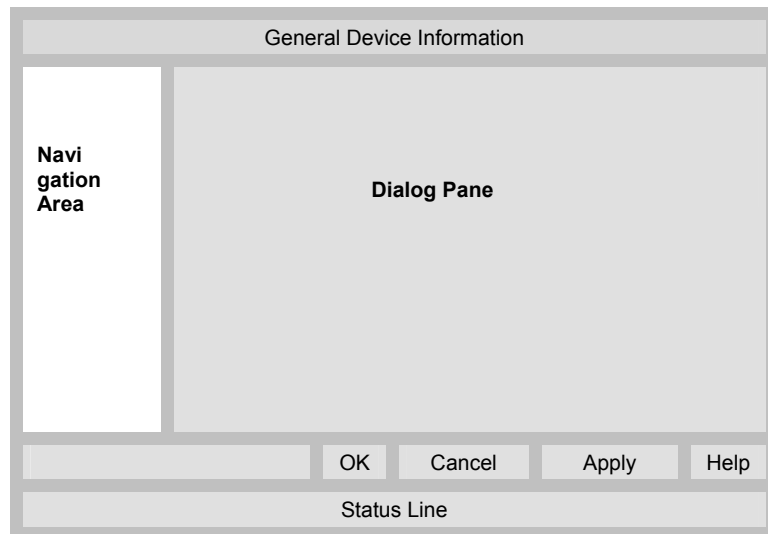


Figure 1: Dialog Structure of EtherNet/IP Generic Adapter DTM

## 1.4.1 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

## 1.4.2 Navigation Area

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

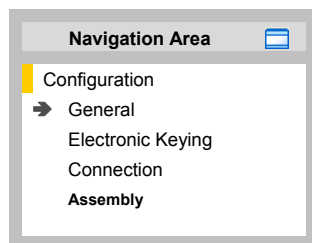


Figure 2: Navigation Area

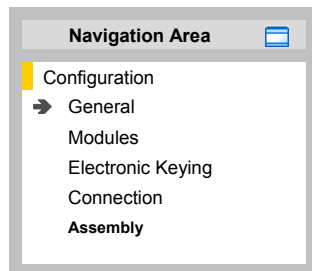




Figure 3: Navigation Area (modulare DTM)

- 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	<b>Opening</b> the navigation area (below left side).

### 1.4.3 Dialog Panes

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

Configuration	
General	On the pane <b>General</b> EtherNet/IP Adapter information is displayed. For further information, refer to section <i>General</i> on page 18.
Modules (modular DTM only)	At the <b>Modules</b> page the modules can be configured. For further information, refer to section <i>Modules (modular DTM)</i> on page 19.
Electronic Keying	At the <b>Electronic Keying</b> 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 <i>Electronic Keying</i> on page 20.
Connection	At the <b>Connection</b> pane the connection can be parameterized. For further information, refer to section <i>Connection</i> on page 22 .
Assembly	At the <b>Assembly</b> pane the configured input / output connections of the EtherNet/IP adapter are displayed. For further information, refer to section <i>Assembly</i> on page 26.

Table 4: Overview Dialog Panes

### 1.4.4 OK, Cancel, Apply and Help

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

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

Table 5: OK, Cancel, Apply and Help

## 1.4.5 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.

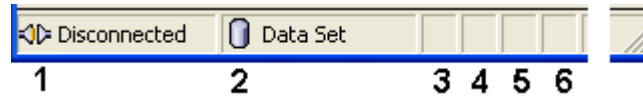


Figure 4: Status Bar – Status Fields 1 to 6

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

Table 6: Status Bar Icons [1]

Offline State

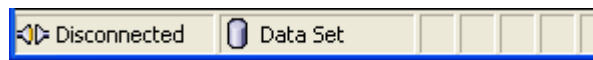


Figure 5: Status Bar Display Example

## 2 Getting Started and Instructions Step by Step

### 2.1 Overview Configuration Steps

The following table describes the steps on how to configure an EtherNet/IP Adapter device with the EtherNet/IP generic Adapter DTM as it is typical for many cases.

#	Step	Short Description	For detailed information see section	Page
1	Add EtherNet/IP generic Adapter DTM in the Device Catalog	Add the adapter DTM in the Device Catalog by importing the device description file to the Device Catalog. Depending of the FDT Container. For netDevice: - <b>Network &gt; Import Device Descriptions.</b>	(See User Manual <i>netDevice</i> and <i>netProject</i> )	-
2	Load device catalog	Depending of the FDT Container: For netDevice: - select <b>Network &gt; Device Catalog</b> , - select button <b>Reload Catalog</b> .	(See User Manual <i>netDevice</i> and <i>netProject</i> )	-
3	Create new project / Open existing project	Depending of the frame application. For the configuration software: - select <b>File &gt; New</b> or <b>File &gt; Open</b> .	(See User Manual of the <i>Frame Application</i> )	-
4	Insert Scanner or Adapter device icon into configuration	For netDevice: - in the Device Catalog click to the Scanner icon, - and insert the device icon via drag and drop <b>to the line</b> 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 <b>Scanner bus line</b> in the network view.	(See User Manual <i>netDevice</i> and <i>netProject</i> )	-
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 <b>Configuration &gt; General</b> , - set the IP settings for the adapter device, - select <b>Configuration &gt; Modules</b> (for modular DTM only), - configure the modules of the modular EtherNet/IP Adapter, - select <b>Configuration &gt; Electronic Keying</b> , - select the keying method and configure it if necessary, - select <b>Configuration &gt; Connection</b> , - configure the connection points, - select <b>Configuration &gt; Assembly</b> , - configure Instance ID and Data length, - configure a <i>modular device</i> according to the description <i>Configuring a modular EtherNet/IP Adapter</i> , - close the Adapter DTM configuration dialog via <b>OK</b> .	<i>Configuring Parameters of the Adapter Device</i>  <i>General</i>  <i>Modules (modular DTM) (for modular DTM only)</i>  <i>Electronic Keying</i>  <i>Connection</i>  <i>Assembly</i>  <i>Configuration Steps for modular EtherNet/IP Devices</i>	16  18 19  20  22 26 30
6	Configure Scanner device	Configure the Scanner device via EtherNet/IP Scanner DTM. Important: Enter the IP settings of the EtherNet/IP Adapter device.	(See User Manual <i>DTM for EtherNet/IP Scanner devices</i> )	-
7	Save project	Depending of the frame application. For the configuration software: - select <b>File &gt; Save</b> .	(See User Manual of the <i>Frame Application</i> )	-

Table 7: Getting Started - Configuration Steps

## 2.2 Configuring Parameters of the Adapter Device

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

In the EtherNet/IP Scanner DTM:

1. Make the IP settings for the Adapter device.

In the generic EtherNet/IP Adapter DTM:

2. Configure the modules of the modular EtherNet/IP Adapter (for modular DTM only):
  - Select **Configuration > Modules** in the navigation area.
3. Select the electronic keying method and configure if necessary:
  - Select **Configuration > Electronic Keying** in the navigation area.
4. Configure the connection points:
  - Select **Configuration > Connetcion** in the navigation area.
5. Configure Instance ID and Data length:
  - Select **Configuration > Assembly** in the navigation area.
6. Configure a *modular device* according to the description *Configuring a modular EtherNet/IP Adapter*.



---

For more information refer to section *Electronic Keying* on page 20, to section *Modules (modular DTM)* on page 19t, to section *Connection* on page 22 , to section *Assembly* on page 26 and to section *Configuration Steps for modular EtherNet/IP Devices* on page 30t.

---

### 3 Configuration

#### 3.1 Overview Configuration

##### Configuration Dialog Panes

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

Section	Page
General	18
Modules (modular DTM) (for modular DTM only)	19
Electronic Keying	20
Connection	22
Assembly	26

Table 8: Dialog Panes Configuration

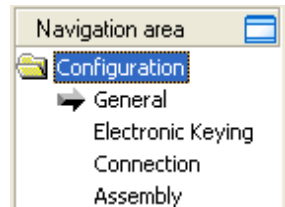


Figure 6: Navigation Area - Configuration

EtherNet/IP Modular Generic Adapter DTM:

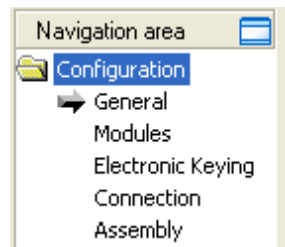


Figure 7: Navigation Area - Configuration (modular DTM)



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

### 3.2 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.

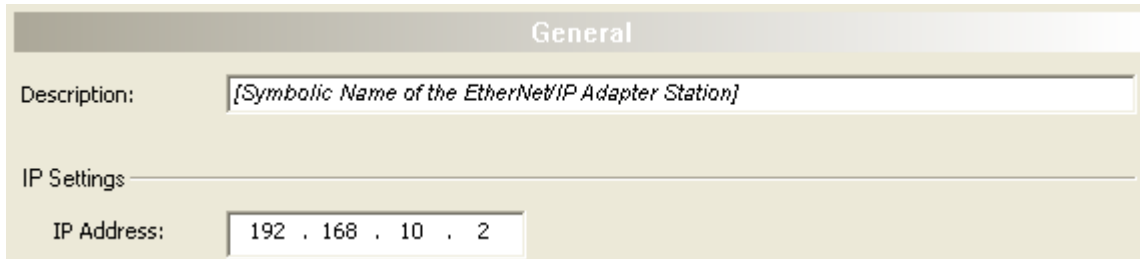


Figure 8: Configuration > General

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

### 3.3 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.

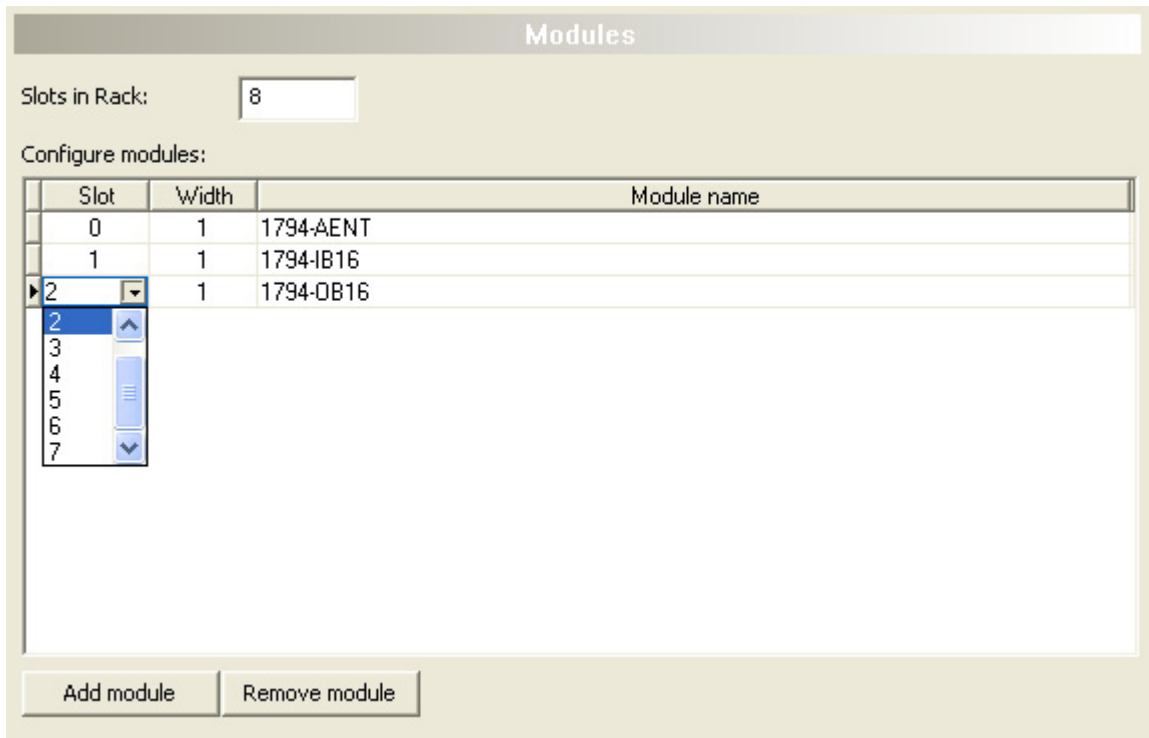


Figure 9: Configuration > Modules Pane (modular DTM)

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

Table 10: Modules Pane Parameters

### 3.4 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.

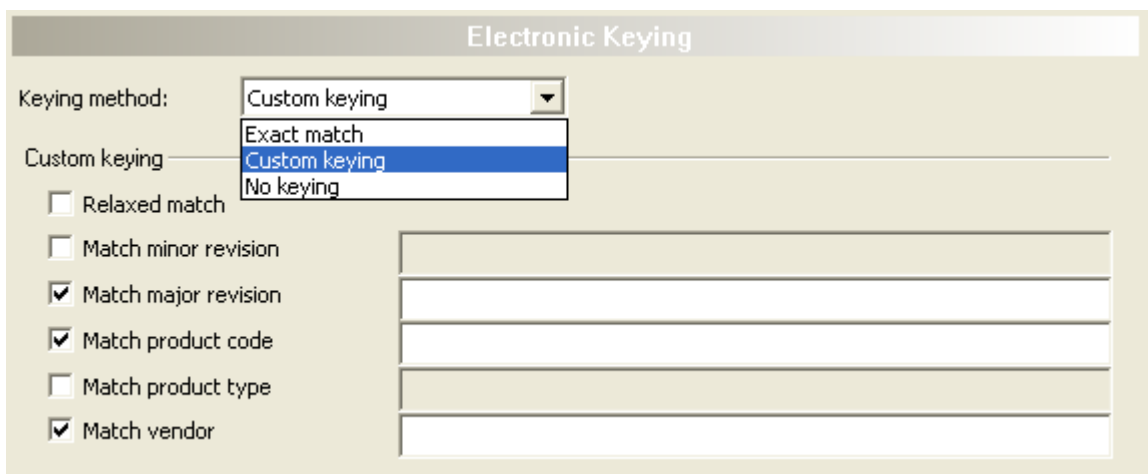


Figure 10: Configuration > Electronic Keying

EtherNet/IP Modular Generic Adapter DTM:

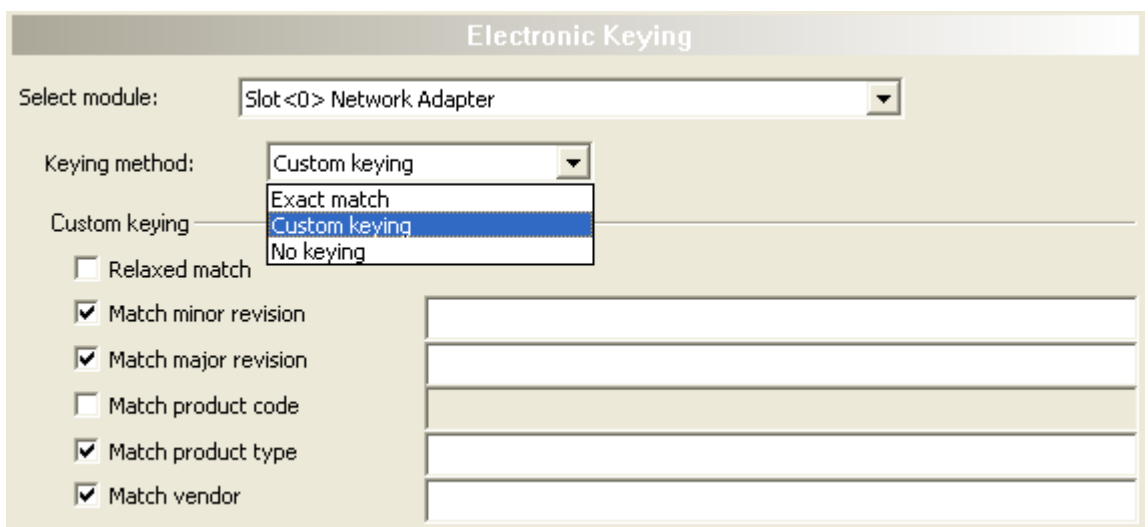


Figure 11: Configuration > Electronic Keying (modular DTM)

- Select a Module (modular DTM only).
- Select a **Keying method**.

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

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.

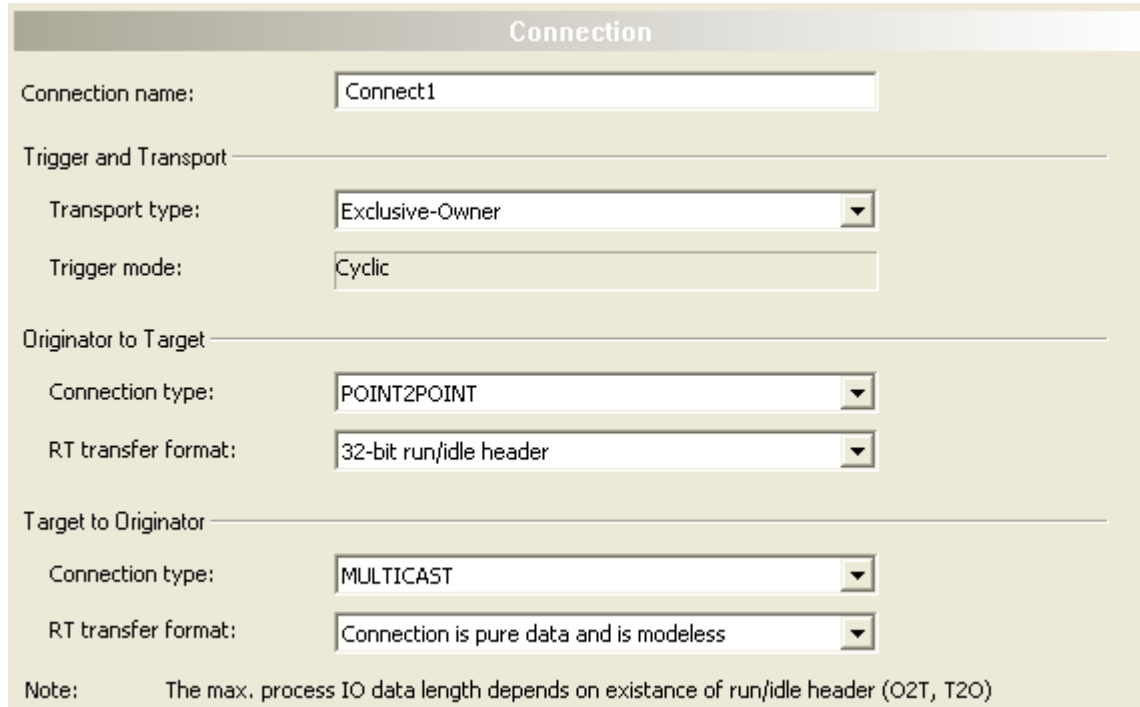
Parameter	Meaning
Relaxed Match	If checked: Restricted validation of the electronic identity for devices. To indicate relaxed match to an adapter, the scanner sets bit 7 in major revision.
Match minor Revision	If checked: For electronic keying consistency to minor revision is relevant and gets verified.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

## 3.5 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.



The screenshot shows a configuration window titled "Connection". It is divided into several sections:

- Connection name:** A text input field containing "Connect1".
- Trigger and Transport:**
  - Transport type:** A dropdown menu set to "Exclusive-Owner".
  - Trigger mode:** A text input field containing "Cyclic".
- Originator to Target:**
  - Connection type:** A dropdown menu set to "POINT2POINT".
  - RT transfer format:** A dropdown menu set to "32-bit run/idle header".
- Target to Originator:**
  - Connection type:** A dropdown menu set to "MULTICAST".
  - RT transfer format:** A dropdown menu set to "Connection is pure data and is modeless".

**Note:** The max. process IO data length depends on existence of run/idle header (O2T, T2O)

Figure 12: Configuration > Connection

EtherNet/IP Modular Generic Adapter DTM:

Connection

Select module: Slot <1> I/O Module1 ▼

Connection name: Connect1

---

Trigger and Transport

Transport type: Exclusive-Owner ▼

Trigger mode: Cyclic

---

Originator to Target

Connection type: POINT2POINT ▼

RT transfer format: 32-bit run/idle header ▼

---

Target to Originator

Connection type: MULTICAST ▼

RT transfer format: 32-bit run/idle header ▼

Size Adder: 4

Note: The max. process IO data length depends on existence of run/idle header (O2T, T2O) and SizeAdder byte length (T2O only)

Figure 13: 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
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.

---

### 3.6 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.

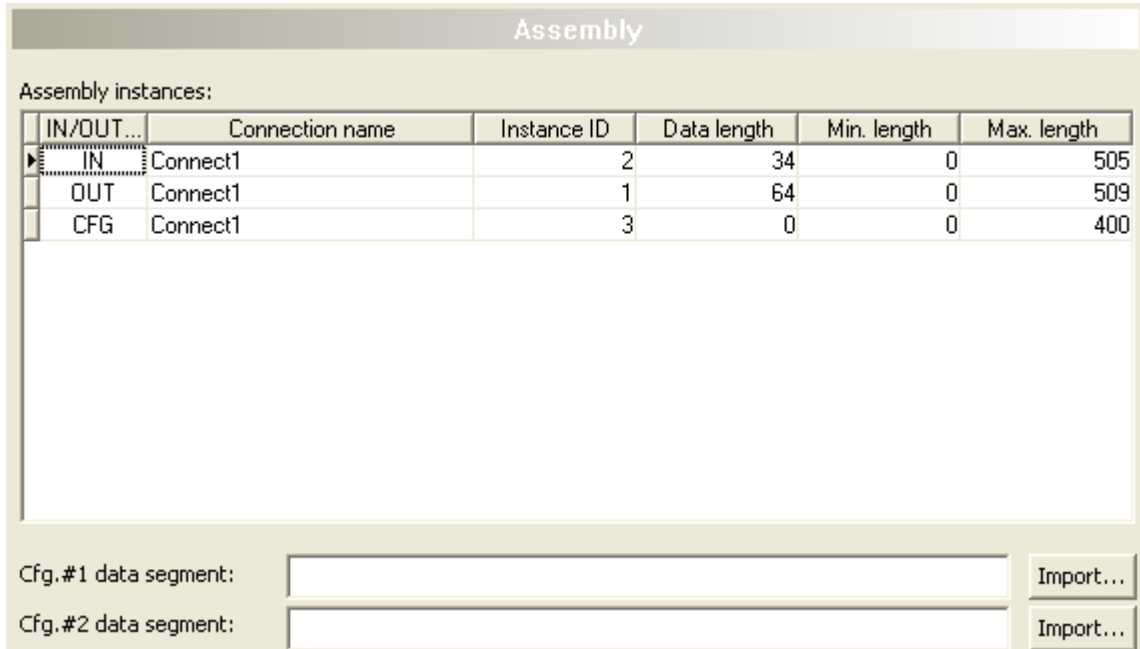


Figure 14: Configuration > Assembly

EtherNet/IP Modular Generic Adapter DTM:

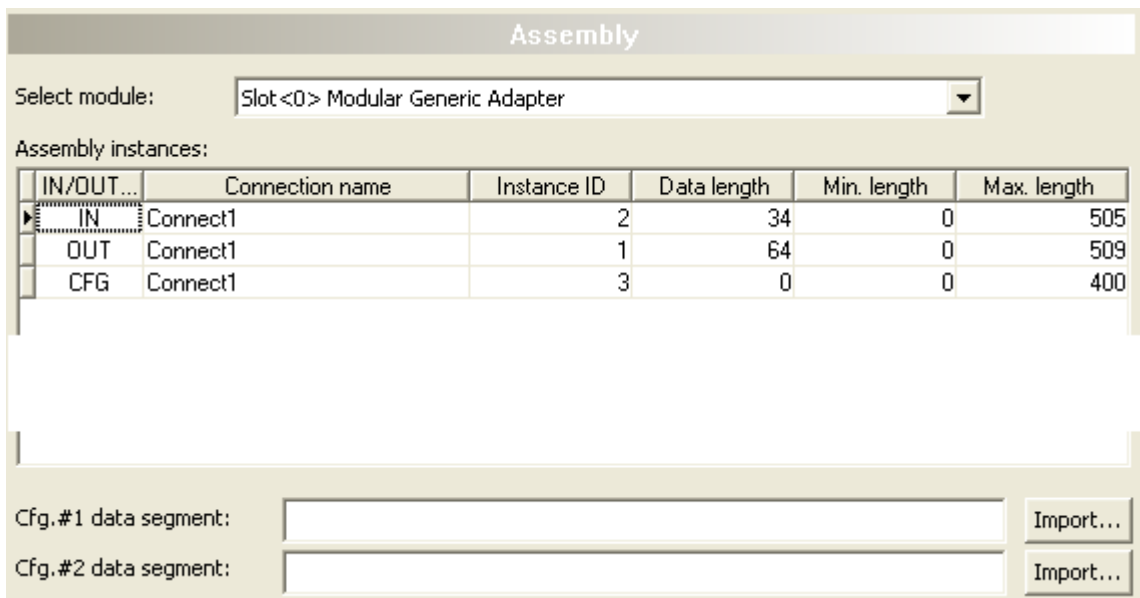


Figure 15: Configuration > Assembly (modular DTM)

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)	<p>Maximum data length in Bytes</p> <p>For the IO messaging there are 511 Bytes max., 9-Bit CIP Container Length, available.</p> <p>Apart from process IO data, the CIP message contains:</p> <ul style="list-style-type: none"> <li>-1- CIP Counter, 2 Bytes, always</li> <li>-2- Real-Time Header (32-bit Run/Idle Header) 4 Bytes, if configured (for each direction)</li> <li>-3- SizeAdder Bytes (according to module EDS) in modular Adapters, for T2O direction, if available.</li> </ul> <p>Hence, the <b>max. Assembly Instance data length</b> is calculated as follows:</p> <p><b>Max_Data_Len</b> = 509 - sizeof( run_time_header) - sizeof ( module_SizeAdder)</p> <p>509 Bytes = 511 Bytes (Max. avail.) - 2 Bytes CIP Counter (always)</p> <p>sizeof( run_time_header) = 4 Bytes if configured, 0 Bytes else.</p> <p>sizeof( module_SizeAdder) = n Bytes (EDS) if available for T2O direction. For O2T it is always 0 Bytes.</p>	
Max. length (CFG)	Maximum data length in Bytes	400
Cfg.#1 data segment	<p>Configuration data segment 1</p> <p>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.</p>	
Cfg.#2 data segment	<p>Configuration data segment 2</p> <p>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.</p>	
Import ...	The data segment configuration file can be imported.	

Table 15: Parameters Dialog Page Assembly

## 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.

## 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 files.

---

## 4.2 Configuration Steps for modular EtherNet/IP Devices

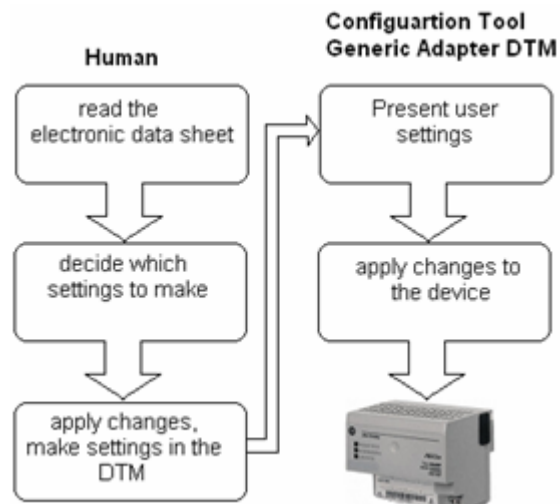


Figure 16: Parameterizing a modular EtherNet/IP Device

## 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 Module 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 its 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.

## 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 RackN 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 configuration 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.

### 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 row 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 row *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 row *Slot* to **2**, *Width* to **1** and the *Module name* to "**1794 - 16 Point 24V DC Output, Source**".

## 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
```

“**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.

#### 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.

### 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
<pre>ProxyParam7 = 0,     ''     0x0000,     0xC7,2,     "output size","", "",     Module,Module,Module,     0,0,0,0,     ''''     0;</pre>	<pre>ProxiedParam7 = 0,     ''     0x0000,     ''     "output size","", "",     0,0,0,     ''''     ;</pre>

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.

### 4.3.4.8 Specifying T2O Data Size

In ProxyConnect1 there's an entry:

```
,ProxyParam1,Assem5, $ T=>0 default,description
```

Adapter EDS	I/O Module EDS
<pre>ProxyParam1 = 0,     '     0x0000,     0xC7,2,     "input size",     "", "",     Module,Module,Module,     0,0,0,0,,,,,     0;</pre>	<pre>ProxiedParam1 = 0,     '     0x0000,     '     "input size",     "", "",     2,4,4,     '     '     ;</pre>

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 than 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.

#### 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".

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”**.

The Cfg.#2 is defined as:

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

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

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
```

```
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.

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";
```

➤ 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".

## 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.

### 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.

### 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
<pre>ProxyParam7 = 0,     ''     0x0000,     0xC7,2,     "output size","", "",     Module,Module,Module,     0,0,0,0,     ''''     0;</pre>	<pre>ProxiedParam7 = 0,     ''     0x0000,     ''     "output size","", "",     2,2,2,     ''''     ''''     ;</pre>

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.

### 4.3.5.8 Specifying T2O Data Size

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

```
,ProxyParam1,Assem5, $ T=>0 default,description
```

Adapter EDS	I/O Module EDS
<pre>ProxyParam1 = 0,     '     0x0000,     0xC7,2,     "input size",     "", "",     Module,Module,Module,     0,0,0,0,,,,,     0;</pre>	<pre>ProxiedParam1 = 0,     '     0x0000,     '     "input size",     "", "",     2,2,2,     '     '     ;</pre>

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.

### 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**”.



`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**".

## 5 Lists

### 5.1 List of Figures

Figure 1: Dialog Structure of EtherNet/IP Generic Adapter DTM	11
Figure 2: Navigation Area	12
Figure 3: Navigation Area (modulare DTM)	12
Figure 4: Status Bar – Status Fields 1 to 6	14
Figure 5: Status Bar Display Example	14
Figure 6: Navigation Area - Configuration	17
Figure 7: Navigation Area - Configuration (modular DTM)	17
Figure 8: Configuration > General	18
Figure 9: Configuration > Modules Pane (modular DTM)	19
Figure 10: Configuration > Electronic Keying	20
Figure 11: Configuration > Electronic Keying (modular DTM)	20
Figure 12: Configuration > Connection	22
Figure 13: 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	23
Figure 14: Configuration > Assembly	26
Figure 15: Configuration > Assembly (modular DTM)	26
Figure 16: Parameterizing a modular EtherNet/IP Device	30

### 5.2 List of Tables

Table 1: Descriptions Dialog Panes	4
Table 2: List of Revisions	5
Table 3: General Device Information	12
Table 4: Overview Dialog Panes	13
Table 5: OK, Cancel, Apply and Help	13
Table 6: Status Bar Icons [1]	14
Table 7: Getting Started - Configuration Steps	15
Table 8: Dialog Panes Configuration	17
Table 9: General Pane Parameters	18
Table 10: Modules Pane Parameters	19
Table 11: Electronic Keying > Select module	21
Table 12: Electronic Keying > Keying Method	21
Table 13: Electronic Keying > Custom Keying	21
Table 14: Parameter Configuration > Connection	24
Table 15: Parameters Dialog Page Assembly	27
Table 16: Example Adapter Data	32
Table 17: Adapter and I/O Module EDS	37
Table 18: Adapter and I/O Module EDS	38
Table 19: Adapter and I/O Module EDS	41
Table 20: Adapter and I/O Module EDS	47
Table 21: Adapter and I/O Module EDS	48
Table 22: Adapter and I/O Module EDS	50
Table 23: User Rights Configuration (D = Displaying, X = Editing, Configuring)	55

## 6 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

## 7 Appendix

### 7.1 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.

#### 7.1.1 Configuration

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

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

### 7.2 References

- [1] Device Type Manager (DTM) Style Guide, Version 1.0 ; FDT-JIG - Order No. <0001-0008-000>
- [2] EtherNet/IP-Adapter; Protocol API (Hilscher), Revision 9

## 7.3 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](mailto:info@hilscher.com)

#### Support

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

### Subsidiaries

#### China

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

#### Support

Phone: +86 (0) 21-6355-5161  
E-Mail: [cn.support@hilscher.com](mailto: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](mailto:info@hilscher.fr)

#### Support

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

#### India

Hilscher India Pvt. Ltd.  
New Delhi - 110 025  
Phone: +91 11 40515640  
E-Mail: [info@hilscher.in](mailto:info@hilscher.in)

#### Italy

Hilscher Italia srl  
20090 Vimodrone (MI)  
Phone: +39 02 25007068  
E-Mail: [info@hilscher.it](mailto:info@hilscher.it)

#### Support

Phone: +39 02 25007068  
E-Mail: [it.support@hilscher.com](mailto:it.support@hilscher.com)

#### Japan

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

#### Support

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

#### Korea

Hilscher Korea Inc.  
Suwon, 443-810  
Phone: +82-31-204-6190  
E-Mail: [info@hilscher.kr](mailto:info@hilscher.kr)

#### Switzerland

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

#### Support

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

#### USA

Hilscher North America, Inc.  
Lisle, IL 60532  
Phone: +1 630-505-5301  
E-Mail: [info@hilscher.us](mailto:info@hilscher.us)

#### Support

Phone: +1 630-505-5301  
E-Mail: [us.support@hilscher.com](mailto:us.support@hilscher.com)