



Logix 5000 Controllers EDS AOP Guidelines for Logix Designer

1756 ControlLogix, 1756 GuardLogix, 1769 CompactLogix,
1769 Compact GuardLogix, 1789 SoftLogix, 5069
CompactLogix, 5069 Compact GuardLogix, Studio 5000
Logix Emulate

Rockwell Automation Publication 1756-PM002F-EN-D - November 2023
Supersedes Publication 1756-PM002E-EN-D - July 2022



Important User Information

Read this document and the documents listed in the additional resources section about installation, configuration, and operation of this equipment before you install, configure, operate, or maintain this product. Users are required to familiarize themselves with installation and wiring instructions in addition to requirements of all applicable codes, laws, and standards.

Activities including installation, adjustments, putting into service, use, assembly, disassembly, and maintenance are required to be carried out by suitably trained personnel in accordance with applicable code of practice.

If this equipment is used in a manner not specified by the manufacturer, the protection provided by the equipment may be impaired.

In no event will Rockwell Automation, Inc. be responsible or liable for indirect or consequential damages resulting from the use or application of this equipment.

The examples and diagrams in this manual are included solely for illustrative purposes. Because of the many variables and requirements associated with any particular installation, Rockwell Automation, Inc. cannot assume responsibility or liability for actual use based on the examples and diagrams.

No patent liability is assumed by Rockwell Automation, Inc. with respect to use of information, circuits, equipment, or software described in this manual.

Reproduction of the contents of this manual, in whole or in part, without written permission of Rockwell Automation, Inc., is prohibited.

Throughout this manual, when necessary, we use notes to make you aware of safety considerations.



WARNING: Identifies information about practices or circumstances that can cause an explosion in a hazardous environment, which may lead to personal injury or death, property damage, or economic loss.



ATTENTION: Identifies information about practices or circumstances that can lead to personal injury or death, property damage, or economic loss. Attentions help you identify a hazard, avoid a hazard, and recognize the consequence.

IMPORTANT

Identifies information that is critical for successful application and understanding of the product.

Labels may also be on or inside the equipment to provide specific precautions.



SHOCK HAZARD: Labels may be on or inside the equipment, for example, a drive or motor, to alert people that dangerous voltage may be present.



BURN HAZARD: Labels may be on or inside the equipment, for example, a drive or motor, to alert people that surfaces may reach dangerous temperatures.



ARC FLASH HAZARD: Labels may be on or inside the equipment, for example, a motor control center, to alert people to potential Arc Flash. Arc Flash will cause severe injury or death. Wear proper Personal Protective Equipment (PPE). Follow ALL Regulatory requirements for safe work practices and for Personal Protective Equipment (PPE).

Rockwell Automation recognizes that some of the terms that are currently used in our industry and in this publication are not in alignment with the movement toward inclusive language in technology. We are proactively collaborating with industry peers to find alternatives to such terms and making changes to our products and content. Please excuse the use of such terms in our content while we implement these changes.

This manual includes new and updated information. Use these reference tables to locate changed information.

Grammatical and editorial style changes are not included in this summary.

Global changes

This table identifies changes that apply to all information about a subject in the manual and the reason for the change. For example, the addition of new supported hardware, a software design change, or additional reference material would result in changes to all of the topics that deal with that subject.

No global changes for this release.

New or enhanced features

Change	Topic
Support for the Logix Designer v34 data types is in v36.00.00 and later. If v36.00.00 is installed on a computer with other Logix Designer versions, then support for v34 data types is supported in v34.00.00 and later.	Introduction on page 13
Updated the unsigned parameter description	License key on page 16
Added example to illustrate the EDS elements used to specify a single Logix Designer string.	Supported uses on page 65
Updated topic to note that the result of nested assemblies is now nested structures.	Nested assemblies on page 65
Updated the supported data types	Data type field on page 70

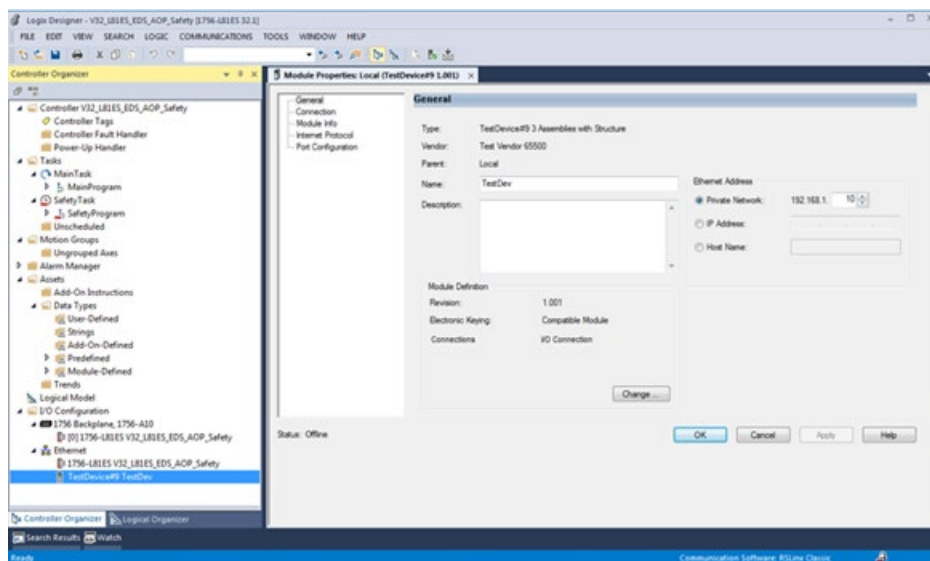
Preface	Studio 5000 environment 10
	Access the attachments 10
	Additional resources 11
	Legal Notices..... 11
Introduction	Chapter 1
	References 14
	Definitions, acronyms, and abbreviations..... 14
	General EDS file information.....14
	EDS file authoring tools 14
	EDS file installation 14
	Icon files.....15
	License key.....16
Create an EDS file	Chapter 2
	Connections 19
	Input and Output data structures and parameters 19
	Input tags..... 20
	Output tags 20
	I/O parameter definition 21
	Configuration data structures and parameters..... 22
	Configuration tags 22
	Configuration parameter definition 23
	Multiple language support 25
Module Configuration dialog boxes	Chapter 3
	General tab 27
	About Module Profile dialog box..... 29
	Module Definition dialog box 30
	Custom Data Type dialog box 32
	Connection tab 34
	Safety tab..... 36
	Module Info tab 37
	Safety Configuration tab 39
	Internet Protocol tab 39
	Port Configuration tab..... 40
	Port Diagnostics dialog box 41
	Network tab 42
	Advanced Network Configuration dialog box 43
	Time Sync tab 44

EDS file sections**Chapter 4**

[File] section	47
Revision, CreateDate, CreateTime, ModDate and ModTime	
keywords	48
[Device] section	48
Catalog and ProdName keywords	48
Catalog keyword.....	48
ProdName keyword	50
VendName keyword.....	51
ProdTypeStr keyword	52
MajRev keyword	52
MinRev keyword	52
[Device Classification] section	53
ClassN keyword.....	53
[Connection Manager] section.....	53
ConnectionN keyword.....	53
ConnectionN fields.....	54
[ParamClass] section.....	64
CfgAssembly keyword.....	64
CfgAssemblyExa keyword	64
Descriptor keyword.....	65
SafetyCfgAssembly keyword	65
[Assembly] section.....	65
AssemN/AssemExaN keyword.....	65
Supported uses.....	65
Nested assemblies.....	65
AssemN/AssemExaN fields	65
VariantN/VariantExaN keyword	67
[Params] section	68
ParamN keyword.....	68
Input/Output/Configuration details uses	68
ParamN fields.....	68
ParamN enumeration, EnumN.....	73
[Groups] section	73
[Internationalization] section	73
ProdName keyword	74
ParamN keyword.....	74
EnumN keyword	74
GroupN keyword.....	74
AssemN/AssemExaN keyword.....	74
ConnectionN keyword.....	74
[Modular] section	74

	[Ethernet Link Class] section	74
	[TCP/IP Interface Class] section	75
	[DLR Class] section	76
	[Time Sync Class] section	77
	Chapter 5	
Description of data types and tags	Module-Defined data type naming	79
	Tag naming	80
	Module-Defined data type member naming	81
	Chapter 6	
Example EDS files	Minimum EDS	83
	Enhanced EDS, single connection	84
	Enhanced EDS, single connection, configuration data	85
	Enhanced EDS, multiple choice of connections and multiple simultaneous connections.....	87
	Enhanced EDS, safety and standard connections.....	89
Index		

The document provides an overview of the Logix Designer application EDS Add-On Profile (AOP) feature, how Electronic Data Sheet (EDS) content is used by the feature, and guidelines for creating EDS content that integrate well within the Logix Designer application development environment. The EDS file format for the Common Industrial Protocol (CIP) networks is defined by ODVA.



Rockwell Automation recognizes that some of the terms that are currently used in our industry and in this publication are not in alignment with the movement toward inclusive language in technology. We are proactively collaborating with industry peers to find alternatives to such terms and making changes to our products and content. Please excuse the use of such terms in our content while we implement these changes.

Studio 5000 environment


The Studio 5000 Automation Engineering & Design Environment® combines engineering and design elements into a common environment. The first element is the Studio 5000 Logix Designer® application. The Logix Designer application is the rebranding of RSLogix 5000® software and will continue to be the product to program Logix 5000™ controllers for discrete, process, batch, motion, safety, and drive-based solutions.

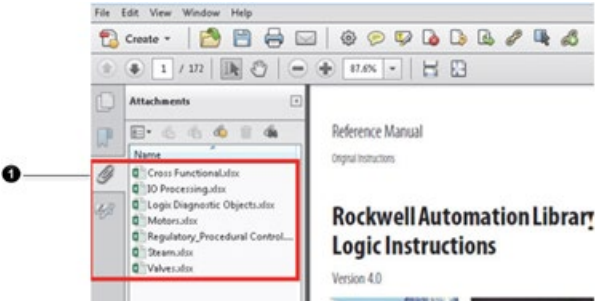


The Studio 5000® environment is the foundation for the future of Rockwell Automation® engineering design tools and capabilities. The Studio 5000 environment is the one place for design engineers to develop all elements of their control system.

Access the attachments

The attachments in this PDF file contain EDS files to assist with configuration.

To use an EDS file, select the attachments link  shown in the left ribbon of the PDF and open the applicable EDS file. If the PDF file opens in a browser and the attachment link does not show, save the PDF file to a computer and then reopen the PDF file to view the attachment symbol.



Number	Description
1	Attachment symbol.

Additional resources

Documents that contain additional information concerning related Rockwell Automation products.

Resource	Description
Logix 5000 Controllers Program Parameters Programming Manual , publication 1756-PM021	Describes how to use program parameters when programming Logix 5000 controllers.
Product Certifications website, http://ab.rockwellautomation.com	Provides declarations of conformity, certificates, and other certification details.

View or download publications at <http://www.rockwellautomation.com/literature>. To order paper copies of technical documentation, contact the local Rockwell Automation distributor or sales representative.

Rockwell Automation recognizes that some of the terms that are currently used in our industry and in this publication are not in alignment with the movement toward inclusive language in technology. We are proactively collaborating with industry peers to find alternatives to such terms and making changes to our products and content. Please excuse the use of such terms in our content while we implement these changes.

Legal Notices

Rockwell Automation publishes legal notices, such as privacy policies, license agreements, trademark disclosures, and other terms and conditions on the [Legal Notices](#) page of the Rockwell Automation website.

Software and Cloud Services Agreement

Review and accept the Rockwell Automation Software and Cloud Services Agreement [here](#).

Open Source Software Licenses

The software included in this product contains copyrighted software that is licensed under one or more open source licenses.

You can view a full list of all open source software used in this product and their corresponding licenses by opening the index.html file located your product's [OPENSOURCE folder](#) on your hard drive.

The default location of this file is:

C:\Program Files (x86)\Rockwell Software\Studio 5000\Logix Designer\ENU\<version number>\relnote\OPENSOURCE\index.htm

You may obtain Corresponding Source code for open source packages included in this product from their respective project web site(s).

Alternatively, you may obtain complete Corresponding Source code by contacting Rockwell Automation via the **Contact** form on the Rockwell Automation website:

<http://www.rockwellautomation.com/global/about-us/contact/contact.page>.

Please include "Open Source" as part of the request text.

Introduction

Creating an EDS is an efficient way to enable a module to operate within the Rockwell Automation integrated architecture, and gives the module developer a better alternative to using the Generic Device AOP.

- The Generic Device AOP requires minimal development time because it uses the generic device user interface, but only specifies the basic connection parameters (assembly instance and the size of the Configuration, Input, and Output arrays). The module developer still creates documentation describing the instance values and tag details. The Generic Device AOP requires the user to manually enter often cryptic configuration parameters.

This document describes the specific parts of the EDS file that affect module configuration, I/O, and the user interface in the Logix Designer application. This document is intended for module developers creating products that integrate with the Logix Designer application who want their module configured by the EDS AOP using EDS content. This document is not intended for end-users applying products. EDS files should only be modified by the product vendor.

Module developers should:

- Be very familiar with the CIP EDS definition; see Chapter 7 in *THE CIP NETWORKS LIBRARY Volume 1, Common Industrial Protocol (CIP™)*, and *THE CIP NETWORKS LIBRARY Volume 2, EtherNet/IP Adaptation of CIP*.
- Be familiar with the Logix Designer application dialog boxes. Dialog boxes are described in *Module Configuration dialog boxes*.
- Be familiar with Module-Defined data types generated by AOPs.

The Logix Designer application EDS AOP feature is limited to devices with a single CIP port that attach to the Ethernet bus.

The EDS AOP feature is supported in the Logix Designer application in v20.00.00 and later. Support for safety devices is in v32.00.00 and later. If v32.00.00 is installed on a PC with other Logix Designer application versions, then safety devices are supported in v24.00.00 and later. Support for v34.00.00 data types is in v36.00.00 or later. If v36.00.00 is installed on a PC with other Logix Designer application versions, then support for Logix Designer v34.00.00 data types are supported in v34.00.00 and later.

Unsupported devices:

- Devices with one or more CIP ports and a non-CIP backplane

- Devices with more than one CIP port that are routable

See also

[Module Configuration dialog boxes](#) on [page 27](#)

[Module-Defined data type naming](#) on [page 79](#)

References

Documents referenced in or related to this document.

- *THE CIP NETWORKS LIBRARY Volume 1, Common Industrial Protocol (CIP™)*
- *THE CIP NETWORKS LIBRARY Volume 2, EtherNet/IP Adaptation of CIP*

Definitions, acronyms, and abbreviations

Definitions, acronyms, and abbreviations used in this document.

Terms	Definition
EDS	Electronic Data Sheet. A text file that contains configuration data for specific device types. The device vendor provides the EDS for a device. The EDS is required for compliance with ODVA standards.
AOP	Add-On Profile. Logix Designer application component separately installed and used for configuring one or more modules.
EDS AOP	EDS Add-On Profile. The Add-On Profile that uses EDS content to construct the profile to support various module types.
Profile	A subsystem of the Logix Designer application. Each supported module type has an associated profile. The profile provides information needed to establish topology and module-defined data types, as well as graphical user interface for configuration (in some cases).

General EDS file information

This section provides general information about EDS tools, EDS installation, and EDS icon files. For example files, see *Example EDS Files*.

See also

[Example EDS files](#) on [page 83](#)

EDS file authoring tools

Create EDS files using any ASCII text editor that is capable of handling the character set specified in *THE CIP NETWORKS LIBRARY Volume 1, Common Industrial Protocol (CIP™)*.



Tip: Rockwell Automation recommends the use of a specialized EDS editing tool, such as the ODVA EZ-EDS tool.

Developers who need to create large families of EDS files of nearly identical modules may choose to use or develop tools that automatically create the EDS files for a set of modules within the product family. If EZ-EDS is not used to construct the file(s), use it to verify the EDS file(s) syntax (ODVA uses the latest version during conformance testing).

EDS file installation

The module vendor provides EDS files. The EDS files deploy in one or more

ways:

- Embedded in the module as described in Section 5A-42 (File Object) of *THE CIP NETWORKS LIBRARY Volume 1, Common Industrial Protocol (CIP™)*.
- Provided on separate media, such as a CD-ROM.
- Downloaded from the module vendor website.

Use the EDS Hardware Installation Tool in the Logix Designer application version 20.00.00 or later to register EDS files. There is no need to shut down the Logix Designer application to register a new EDS, and the EDS is immediately available. The Logix Designer application does not allow the registering of an older version EDS file over a newer version EDS file.

The EDS Hardware Installation Tool issues warnings when it detects problems in the EDS (for example, about discarded entries).

IMPORTANT Consider warnings when registering an EDS. For example, connections may be discarded because of a missing assembly referenced in the connection.

If an EDS file is not available for a module that exists in the Logix Designer application I/O tree, Logix Designer application displays a message that the module is not registered. This can happen when Logix Designer application detects a new module, or when copying a Logix Designer application project ACD file to a different computer that does not have the correct EDS files registered. Logix Designer application also displays a message if the device configuration has a newer version of the EDS than the one registered on the computer. In both cases, the Logix Designer application cannot display the module's **Module Properties** dialog box until the correct EDS file is installed. The Logix Designer application can still download the project to the controller.

Icon files

If there is no module icon file, Logix Designer application assigns a default icon. The icon allows a graphical representation of the module in the FactoryTalk® Linx and Logix Designer application environment. Embed icons in modules using the method described in Section 5A-42 (File Object) of *THE CIP NETWORKS LIBRARY Volume 1, Common Industrial Protocol (CIP™)*.



Tip: Module icon files are highly recommended.

The EZ-EDS tool version 3.9, or later, supports embedding icons inside the EDS file. If icons are embedded inside the EDS using the method described in Chapter 7 of *THE CIP NETWORKS LIBRARY Volume 1, Common Industrial Protocol (CIP™)*, they are extracted automatically from the EDS.

Logix Designer application automatically assigns icons provided as separate files if their file names match the ones specified by the Icon keyword in the Device Description section, [Device] Section, of the EDS. The icon files must reside in the same directory as the EDS. If these conditions are not met, the EDS Hardware Installation Tool allows for manually assignment of an icon file when registering the EDS.

To create icon files, use these guidelines:

- At a minimum, create a 32x32 pixel, 16-color icon in the Microsoft format.

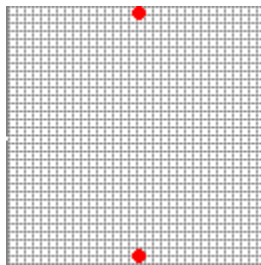
For best results, define a:

- 16x16 pixel, 16 color icon
- 32x32 pixel, 16 color icon
- 48x48 pixel, 256 color icon

Store all three icons in one icon file.

- Center the icon in both the horizontal and vertical directions.
 - Use transparent pixels for the pixels that are not part of the product.
- Do not use solid colors for backgrounds.

The red dots in the graphic below indicate the points at which the icon should be centered horizontally. This insures that the icon appears correctly.



In addition, make sure the Logix Designer application overlays (for example, inhibited device, error indication) are visible and evident in the Logix Designer application I/O tree.

See also

[\[Device\] section](#) on [page 48](#)

License key

The standard form of the EDS AOP does not create named I/O tags. As with the generic EtherNet/IP module profile, the inputs, outputs, and configuration tags are displayed in the form of an array.

For the Logix Designer application to display named tags, the EDS file needs to include a license key. These license keys are issued by RA Technologies Inc. on a case-by-case basis. To obtain an EDS file with a license key, please email RA Technologies Inc. at RATech@RA.Rockwell.com.

Once the license key has been included in the file, there are three further requirements - specific to the I/O structure - that need to be met in order for named tags to be created correctly. These are:

1. The assemblies must have a structure that is built of parameters, as defined in this publication.

2. The parameter data types in the EDS file, and their respective minimum and maximum values need to match the data types that are supported in the Logix Designer application.
3. The assembly structure must follow alignment rules. Parameters that are a 2-byte data type must start on a 2-byte boundary and those that are a 4-byte data type must start on a 4-byte boundary. Parameters must be rearranged, or pad parameters added to meet the alignment rules. These changes must be made in both the EDS and the firmware of the device.

If these requirements are not met, the resulting tag structure will be an array - same as if no EDS license key is present.

A common problem is the use of unsigned data types as the data type for the EDS parameter. An unsigned parameter is omitted from the resulting tag structure if its maximum value is greater than the maximum value of its signed counterpart. For example, a USINT will be omitted in the resulting tag structure if its maximum value is 128 or greater. However in v36 (and in v34 and v35 with AOP Core v30 installed), unsigned parameters are now converted to unsigned tag members with support for the full range of the unsigned data type; therefore, this constraint does not apply.

Create an EDS file

This section gives an overview of the process of creating an EDS file for a module.

See also

[Connections](#) on [page 19](#)

[Input and Output data structures and parameters](#) on [page 19](#)

[Configuration data structures and parameters](#) on [page 22](#)

[Multiple language support](#) on [page 25](#)

Connections

A connection identifies a CIP connection supported by the module and allows for setting some connection parameters that define controller-to-module behavior. Different connections may have different purposes. For example, optimize one connection for speed, and another connection for carrying more data. The connection also defines its Input, Output, and Configuration assemblies. See the sections in *Input and Output data structure and parameters* and *Configuration data structures and parameters*.

Connections are defined in the *[Connection Manager]* section of the EDS file. The connection information appears in the *Module Definition dialog box* and the *Connection tab*.

See also

[Input and Output data structures and parameters](#) on [page 19](#)

[Configuration data structures and parameters](#) on [page 22](#)

[\[Connection Manager\] Section](#) on [page 53](#)

[Module Definition dialog box](#) on [page 30](#)

[Connection tab](#) on [page 34](#)

Input and Output data structures and parameters

This section covers defining the module Input and Output data structures and parameters, and how they are displayed in the Logix Designer application. In the EDS file, the *[Assembly]* section defines the data structures. The EDS file details are described later in this document, beginning with the *EDS File Sections*.

The Input and Output data structures define the parameters whose values are produced by the target module and consumed by the target module.

- The Input and Output tags include descriptive member names, types, and display styles. Always define detailed Input and Output assemblies to take advantage of all the EDS AOP features in Logix Designer.
- In the EDS file, define a detailed assembly or a non-detailed assembly.
 - The detailed assembly results in a detailed data type (highly recommended).
 - The non-detailed assemblies only defines the size of the assembly, and result in a non-detailed data type. See *[Assembly] section* and *[Params] section*.

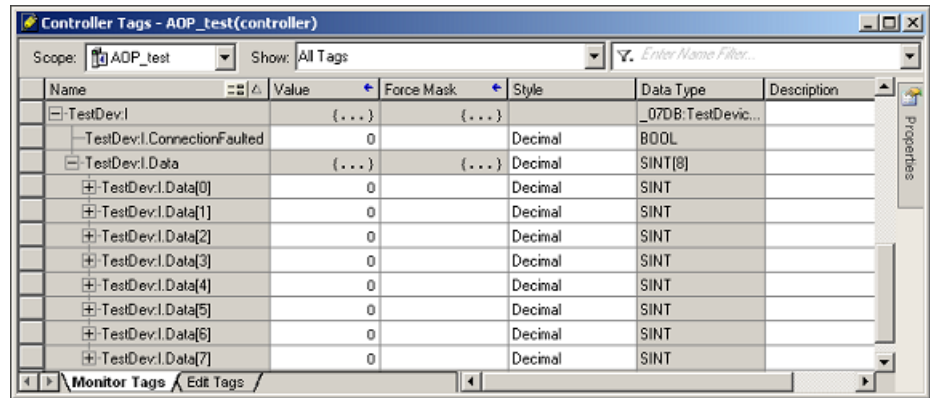
See also

[EDS File sections](#) on [page 47](#)

[\[Assembly\] section](#) on [page 65](#)

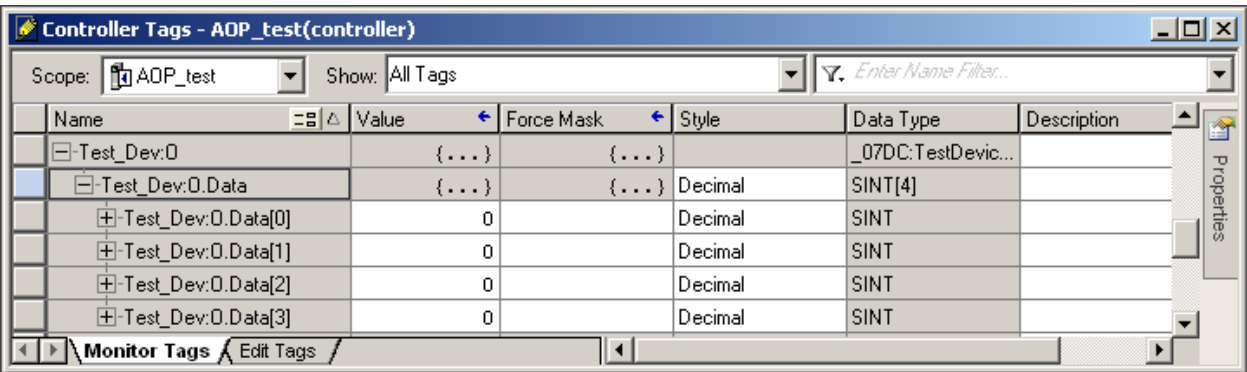
Input tags

The values for Input tags are sent from the target module to the controller.



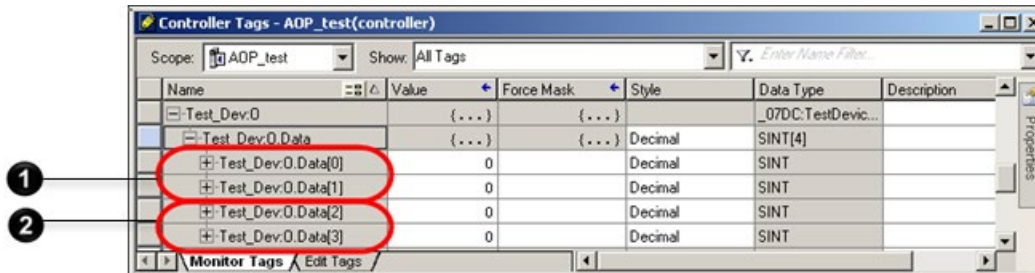
Output tags

The values for Output tags are sent from the controller to the target module.



I/O parameter definition

Define I/O parameters in the *[Params]* section of the EDS file. This example shows two circled Output parameters (the first bit- enumerated).



Number	Description
①	Param2 corresponds to Data[0] and Data[1]
②	Param3 corresponds to Data[2] and Data[3]

The parameters are included in an assembly in the *[Assembly]* section.

```

Assem101 =
    "Output Data",
    ,
    ,
    0x0000,
    ,,
    ,Param2,      $ Output Commands parameter
    ,Param3;      $ Speed Reference parameter
  
```

The parameters are defined in the *[Params]* section.

```

Param2 =
    0,          $ reserved, shall equal 0
    ,,          $ Link Path Size, Link Path
    0x0000,     $ Descriptor
    0xD2,      $ Data Type
    2,          $ Data Size in bytes
    "Output Commands", $ name
    "",         $ units
    "",         $ help string
    ,,0,       $ min, max, default data values
    ,,,,      $ mult, div, base, offset scaling
    ,,,,      $ mult, div, base, offset links
    ;          $ decimal places

Enum2 =
    0, "Stop",
    1, "Run",
    2, "Reverse";

Param3 =
    0,          $ reserved, shall equal 0
    ,,          $ Link Path Size, Link Path
    0x0000,     $ Descriptor
  
```

```

0xC7, $ Data Type
2,      $ Data Size in bytes
"Speed Reference", $ name
"rpm", $ units
"",      $ help string
0,32767,0, $ min, max, default data values
,,, $ mult, div, base, offset scaling
,,, $ mult, div, base, offset links
;      $ decimal places

```

See also

[\[Params\] section](#) on [page 68](#)

[\[Assembly\] section](#) on [page 65](#)

Configuration data structures and parameters

This section covers defining the module configuration data structures and parameters, and how they display in the Logix Designer application. In the EDS file, the *[Assembly]* section defines the data structures. The EDS file details are described later in this document, beginning with the *EDS File sections*.

See also

[EDS File sections](#) on [page 47](#)

[\[Assembly\] section](#) on [page 65](#)

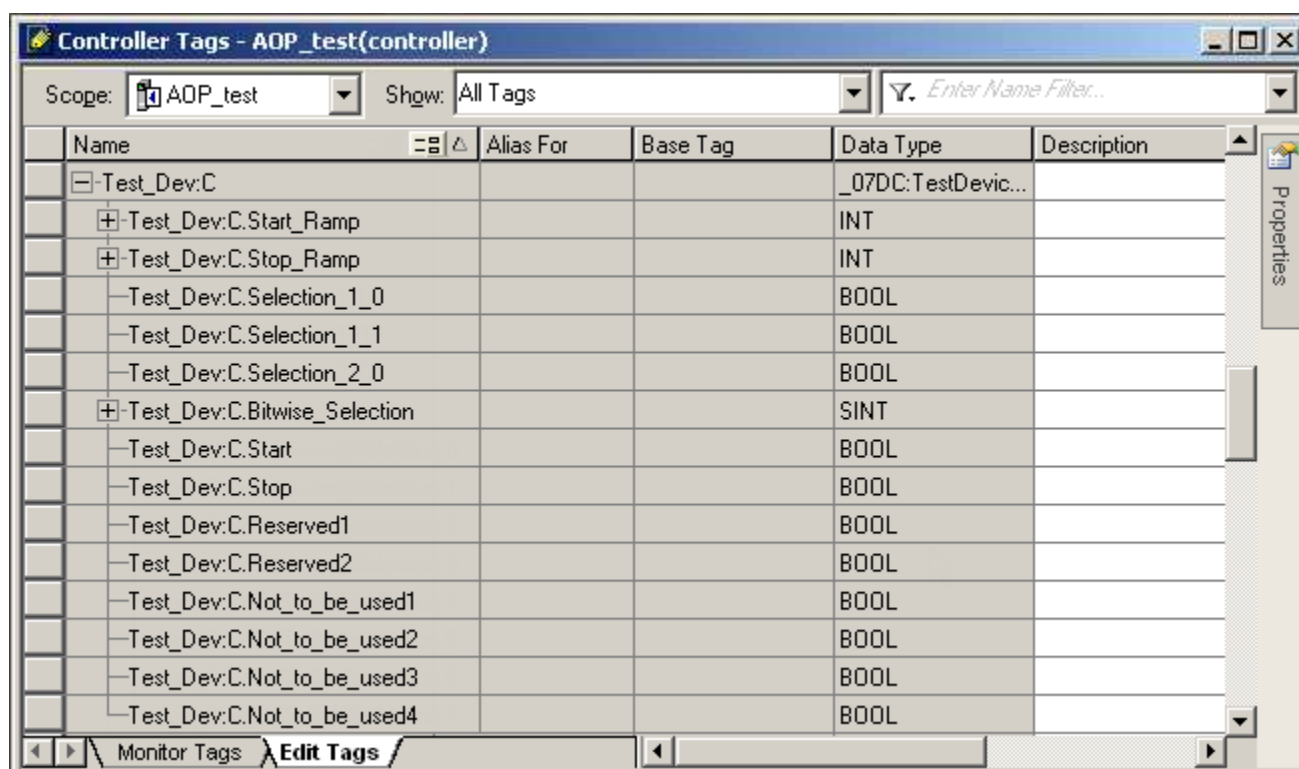
Configuration tags

Configuration tags are parameters that are sent from a controller to a target module when an I/O connection is established. The tags configure and verify how a target module is configured.

The configuration data structure defines the configurable parameters for a module.

- Configuration tags may include descriptive tag names, as well as individual tag data types that define individual configuration parameters.
- In the EDS file, define a detailed assembly or a non-detailed assembly.
 - The detailed assembly results in a detailed data type. (This is highly recommended.)
 - The non-detailed assembly only defines the size of the assembly, and results in a non-detailed data type. See *[Assembly]* in *[Assembly section]* and *[Params]* in *[Params section]*.

- Type of configuration: The configuration is delivered using the Forward_Open service in the Data Segment. The configuration data is described by the ConnectionN entry.



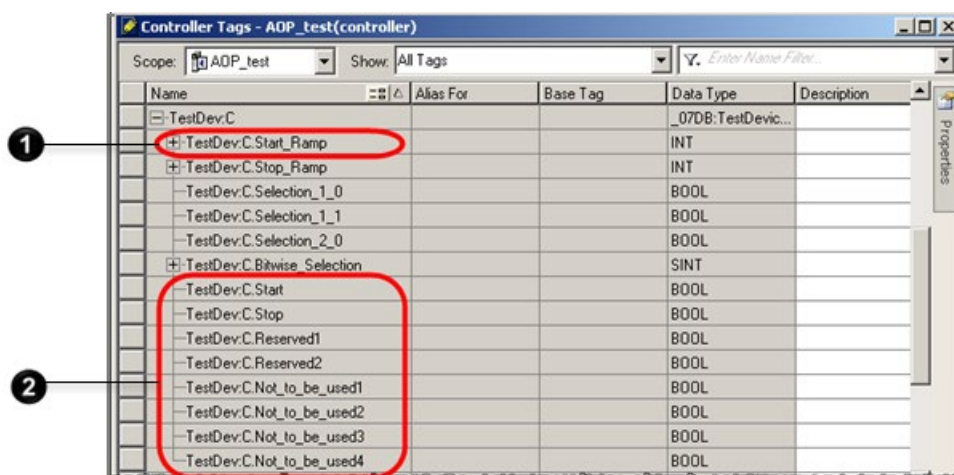
See also

[\[Assembly\] section](#) on [page 65](#)

[\[Params\] section](#) on [page 68](#)

Configuration parameter definition

Define configuration parameters in the *[Params]* section of the EDS file. The example shows two circled configuration items. Safety configuration parameters do not show as tags in the Logix Designer application.



Number	Description
--------	-------------

Number	Description
❶	Corresponds to Param7
❷	Corresponds to Param11

The parameters are included in an assembly in the *[Assembly]* section:

```
Assem100 =
    "Configuration Assembly",
    ,
    ,
    0x0000,
    ,,
    16,Param7,          $ Start Ramp parameter
    16,Param8,
    2,Param9,          $ Selection1 parameter
    1,Param10,
    5,,
    8,Param11;
```

The parameters are defined in the *[Params]* section:

```
Param7 =
    0,                  $ reserved, shall equal 0
    ,,                  $ Link Path Size, Link Path
    0x0000,             $ Descriptor
    0xC7,               $ Data Type
    2,                  $ Data Size in bytes
    "Start Ramp",       $ name
    "ms",               $ units
    "This is the value for the start ramp", $ help string
    1,10000,1000 $ min, max, default data values
    ,,,,               $ mult, div, base, offset scaling
    ,,,,               $ mult, div, base, offset links
    ;                  $ decimal places
```

Parameters can include bit-enumeration:

```
Param11 =
    0,                  $ reserved, shall equal 0
    ,,                  $ Link Path Size, Link Path
    0x0000,             $ Descriptor
    0xD1,               $ Data Type
    1,                  $ Data Size in bytes
    "Bitwise Selection", $ name
    "",                 $ units
    "New Help String", $ help string
    ,,0,               $ min, max, default data values
    ,,,,               $ mult, div, base, offset scaling
    ,,,,               $ mult, div, base, offset links
```



```

;                                $ decimal places
Enum11 =
    0, "Start",
    1, "Stop",
    2, "Reserved",
    3, "Reserved",
    4, "Not to be used",
    5, "Not to be used",
    6, "Not to be used",
    7, "Not to be used";

```



Tip: Parameters used in a Configuration Assembly should not include a link path. A link path makes it available to be written to separately by the Logix Designer application, which can result in configuration inconsistencies between the device and the Logix Designer application.

See also

[\[Assembly\] section](#) on [page 65](#)

[\[Params\] section](#) on [page 68](#)

Multiple language support

Including support for multiple languages in the EDS file is strongly recommended. The Logix Designer application is translated into English, French, German, Spanish, Portuguese, Italian, Korean, Japanese, and Chinese. Names to translate include connections, configuration parameters, and assemblies. Tag member names in data structures always use the default string from the original definition in the EDS file.

When the Logix Designer application displays the **Module Properties** dialog box for a device with an EDS file, these rules control which text shows:

- For non-English languages, the Logix Designer application automatically uses the translated strings from the [Internationalization] section of the EDS file to match the current language setting.
- For English, or if there is no matching translated string, the Logix Designer application uses the English ("eng") string from the [Internationalization] section.
- If there is no English string in the [Internationalization] section, the Logix Designer application uses the default string from the original definition in the EDS file.

```

[Internationalization]
Param7 =
{
    2,
    {"eng", 0xD0, 4, "Start Ramp"},
    {"deu", 0xD0, 4, "Start-Rampe"}
}

```

```
    },  
    {  
    2,  
    {"eng", 0xD0, 4, "ms"},  
    {"deu", 0xD0, 4, "ms"}  
    },  
    {  
    2,  
    {"eng", 0xD0, 4, "The value for the start ramp"},  
    {"deu", 0xD0, 4, "Der Wert für die Start-Rampe"}  
    }  
};
```

The first group of EDS keywords support internationalization. Always translate:

- ProdName
- ConnectionN

The second group of EDS keywords support internationalization. Only translate if they represent items that appear in the **Module Definition** dialog box:

- ParamN
- EnumN
- GroupN
- AssemN, AssemExaN

See *[Internationalization]* section for more information.

See also

[Module Definition dialog box](#) on [page 30](#)

[\[Internationalization\] section](#) on [page 73](#)

Module Configuration dialog boxes

The Logix Designer application uses tabbed dialog boxes with a common look and feel to display the information from an EDS file and from the corresponding module when online. This simplifies device configuration in the integrated architecture. This section lists the dialog boxes, describes their functions, and references the EDS file sections that affect the dialog boxes in the Logix Designer application. Use this section to understand how the EDS AOP displays the information from the EDS in the Logix Designer user interface.

See also

[General tab](#) on [page 27](#)

[Connection tab](#) on [page 34](#)

[Module Info tab](#) on [page 37](#)

[Port Configuration tab](#) on [page 40](#)

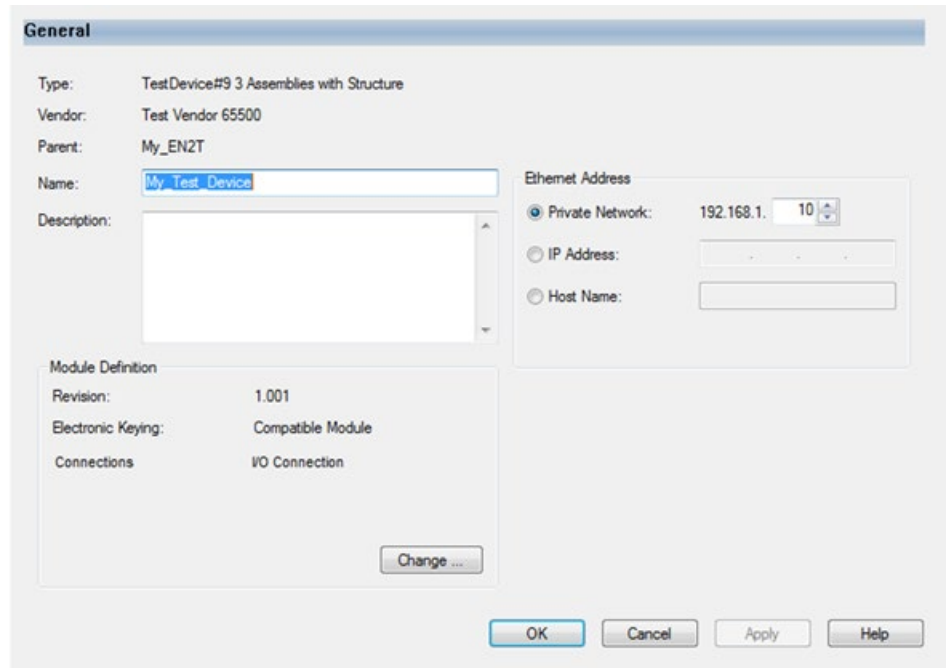
[Network tab](#) on [page 42](#)

General tab

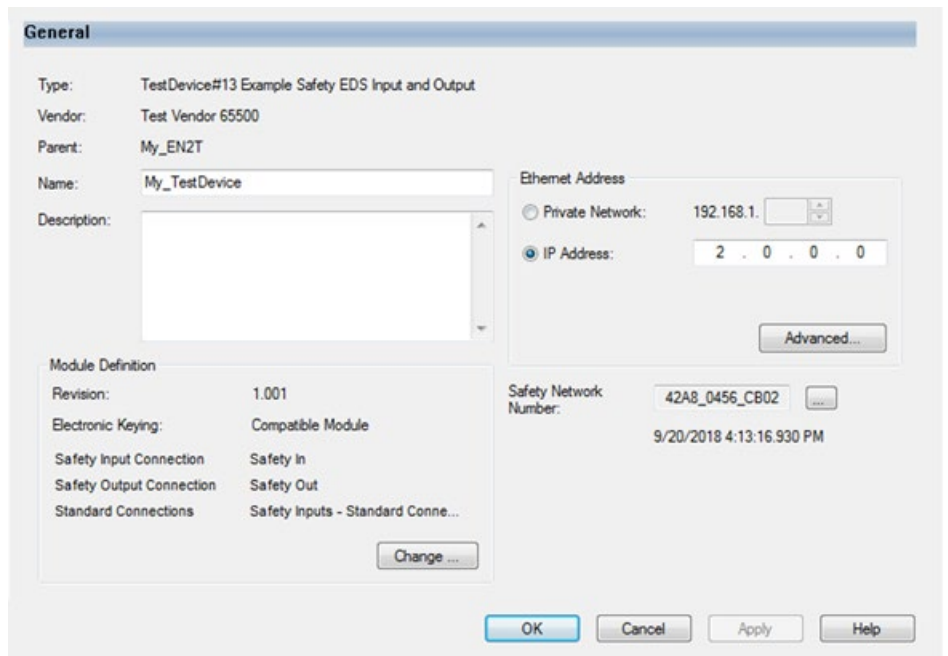
Use the **General** tab to view and configure module properties.

- View module type and vendor name. See *ProdTypeStr* keyword and *VendName* keyword.
- View the name of the parent module in the Logix Designer I/O tree.
- Enter the name and description of the device as it appears in the Logix Designer application.
- Configure network addressing. See *[Device Classification]* section and *[Modular]* section.
- View revision, electronic keying, and connection type information. See *Module Definition dialog box*.
- If a safety connection is enabled, then the **Safety Network Number** displays and can be edited.
- If a safety connection is enabled, an **Advanced** button displays, which allows the configuration of the IP addresses when the module and controller communicate through a Network Address Translation (NAT) device.

Select the module in the Logix Designer I/O tree and press **Enter**, or right-click the module and select **Properties** to open the **Module Properties** dialog box.



The **Module Properties** dialog box, **General** tab when safety is enabled:



See also

[ProdTypeStr keyword](#) on [page 52](#)

[VendName keyword](#) on [page 51](#)

[\[Device Classification\] section](#) on [page 53](#)

[\[Modular\] section](#) on [page 74](#)

[Module Definition dialog box](#) on [page 30](#)

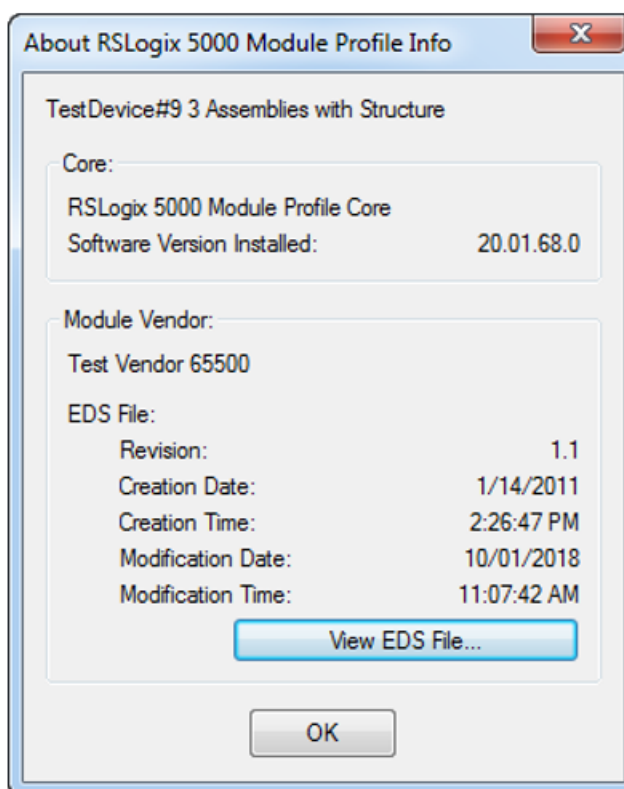
About Module Profile dialog box

The **About Module Profile Info** dialog box displays version and file information about the Logix Designer and the EDS file. In addition, select to view the EDS file in a text editor. Use this dialog box to view:

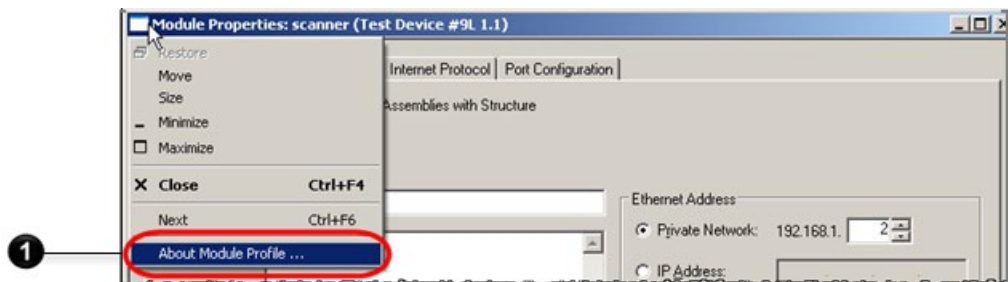
- Logix Designer Module Profile Core version.
- Module vendor. See *VendName* keyword.
- EDS file version, creation date and time, and modification date and time. See *Revision*, *CreateDate*, *CreateTime*, *ModDate*, *EDSFileCRC*, and *ModTime* keywords.



Tip: Select **View EDS file** to view the EDS file in a text editor.



Select the icon in the upper-left corner of the **Module Properties** dialog box to display the context menu, and then select **About Module Profile** to display this dialog box.



Number	Description
1	From the Module Properties dialog box menu, select About Module Profile .

See also

[VendName keyword](#) on [page 51](#)

[Revision, CreateDate, CreateTime, ModDate, EDSFileCRC, and ModTime keywords](#) on [page 48](#)

Module Definition dialog box

Configure the module details and the Input/Output data to transfer using the **Module Definition** dialog box. Typically, when defining the properties for a standard module, select:

- The module revision (major and minor). See *MajRev Keywords* and *MinRev keywords*. Note that the module revision number is not the same as the EDS file version number.
- The electronic keying option.
- Connections that the controller opens (see *ConnectionN keywords*).

Optionally, for modules that support configuring module definition properties, (as defined by the module developer) also:

- Configure connection settings, for example, remote data, selectable assemblies, configurable sizes, or custom data types.

- Select multiple, simultaneous connections, including CIP Safety connections (for those modules that support them).

The **Module Definition** dialog box shows the following configuration:

- Revision:** 1 (dropdown) and 1 (spin box)
- Electronic Keying:** Compatible Module (dropdown)
- Connections:**

Name	Input:	Output:	Tag Suffix
I/O Connection	1		TestDev2:I1
			TestDev2:O1
Select a connection			

Buttons: OK, Cancel, Help

The **Module Definition** dialog box when a safety connection is enabled:

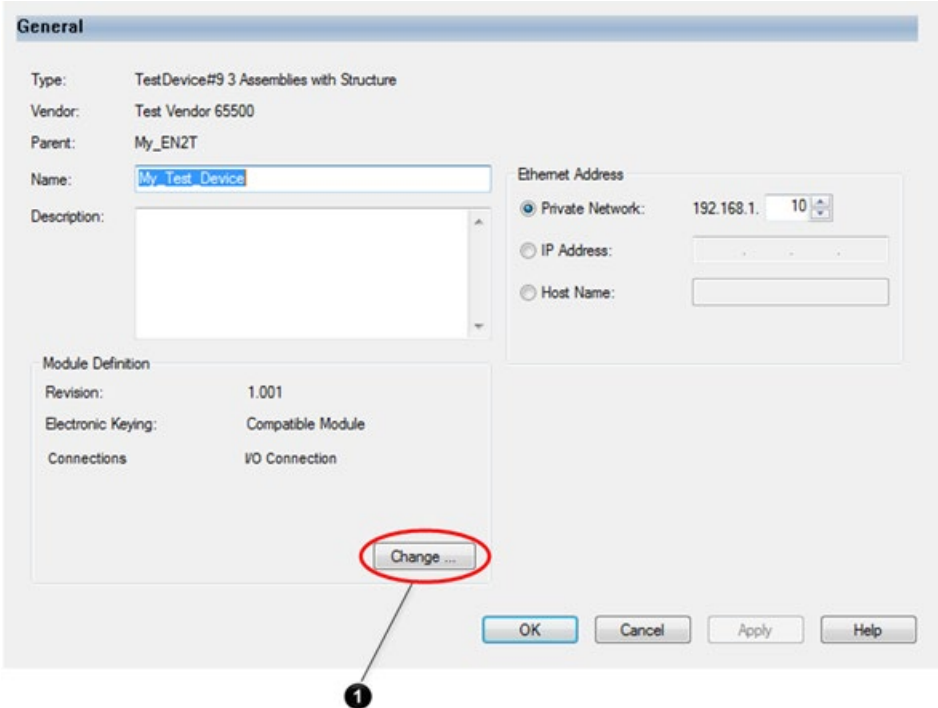
The **Module Definition** dialog box shows the following configuration when a safety connection is enabled:

- Revision:** 1 (dropdown) and 001 (spin box)
- Electronic Keying:** Compatible Module (dropdown)
- Connections:**

Name	Input:	Output:
Safety In		Safety Input:
Safety Out		Safety Output:
Safety Inputs - Standard Connection	Input:	Output:

Buttons: OK, Cancel, Help

In the **Module Properties** dialog box, select **Change** to display the **Module Definition** dialog box.



Number	Description
1	From the Module Properties dialog box, select Change to open the Module Definition dialog box.

See also

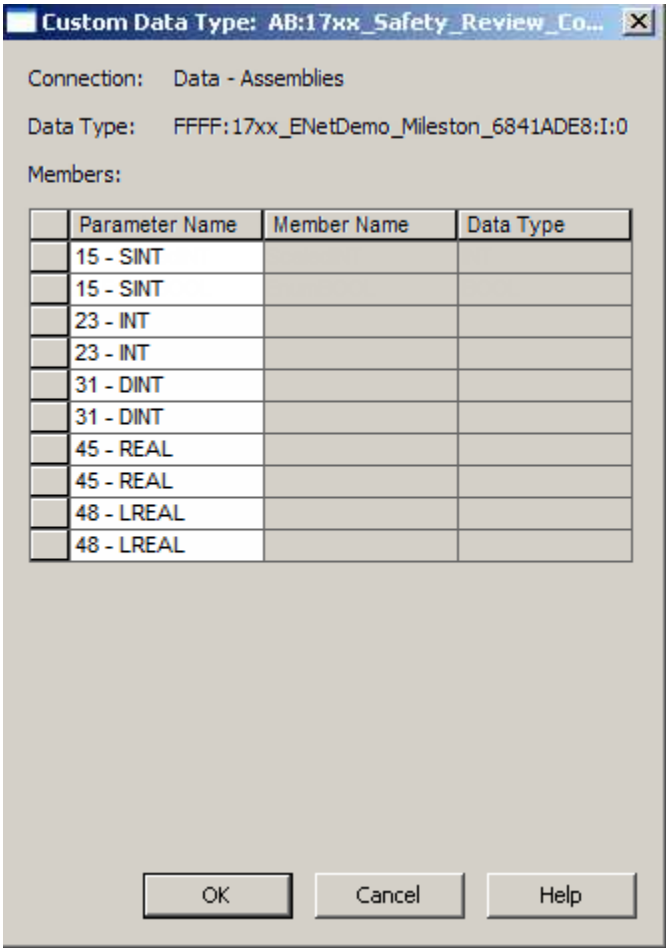
- [MajRev keyword](#) on [page 52](#)
- [MinRev keyword](#) on [page 52](#)
- [ConnectionN keyword](#) on [page 74](#)
- [Enhanced EDS, Multiple Choice of Connections and Multiple Simultaneous Connections](#) on [page 87](#)

Custom Data Type dialog box

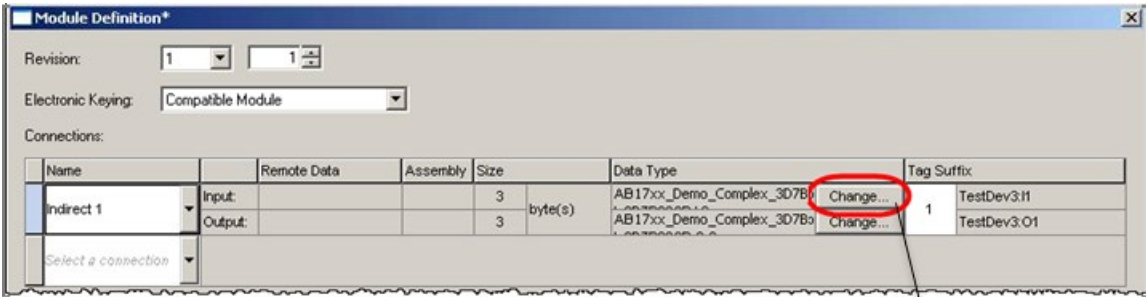
Use the **Custom Data Type** dialog box to configure indirect Input/Output parameters. The dialog box shows:

- Connection name. See *ConnectionN keyword*.

- I/O module-defined data type information. See *T->O* and *O->T* *Format fields* and *Indirect Parameter reference* under *Descriptor field*.



In the **Module Definition** dialog box, select **Change** to display the **Custom Data Type** dialog box. **Change** only appears if indirect parameters exist in Input or Output assemblies associated with the connection.



Number	Description
1	In the Module Definition dialog box, select Change to open the Custom Data Type dialog box. Change only appears if indirect parameters exist in Input or Output assemblies associated with the connection.

See also

[ConnectionN keyword](#) on [page 74](#)

[T->O and O->T Format fields](#) on [page 58](#)

[Descriptor field](#) on [page 66](#)

Connection tab

The **Connection** tab displays the set of configured connections (see *ConnectionN keyword*) and allows setting some connection parameters to define controller-to-module behavior. The **Connection** tab also shows the status of the connection(s) between the controller and the target module.



Tip: The data on the **Connection** tab comes directly from the controller.

n the **Connection** tab:

- Select a **Requested Packet Interval**.
- Specify an **Input Type**.
- Select an **Input Trigger**.
- Select to inhibit the module.

- Configure fault handling and view module faults.

Module Properties: scanner (Test Device #9L 1.1)

General | Connection | Module Info | Internet Protocol | Port Configuration

Name	Requested Packet Interval (RPI) (ms)	Input Type	Input Trigger
I/O Connection	10.0 4.0 - 100.0	Unicast	Cyclic

☐ Inhibit Module

☐ Major Fault On Controller If Connection Fails While in Run Mode

Module Fault

Status: Offline

OK Cancel Apply Help

The **Connection** tab when a safety connection is enabled.

Connection

Name	Requested Packet Interval (RPI) (ms)		Connection over EtherNet/IP	Input Trigger
Safety In	10	Set on Safety Page	Unicast	Application
Safety Out	30	Set by Safety Task	Unicast	Application
Safety Inputs - Standard Connection	20.0	1.0 - 3200.0	Unicast	Cyclic

☐ Inhibit Module

☐ Major Fault On Controller If Connection Fails While in Run Mode

Module Fault

OK

Cancel

Apply

Help

See also

[ConnectionN keyword](#) on [page 74](#)

Safety tab

The **Safety** tab only displays if a safety connection is enabled. Use the **Safety** tab to:

- Select a requested packet interval.
- View reaction time.
- View and reset configuration ownership.
- View and copy the configuration signature.

The **Safety** tab if the EDS contains a safety configuration.

The screenshot shows the 'Safety' tab with a table of connection parameters and configuration options.

Connection Type	Requested Packet Interval (RPI) (ms)	Connection Reaction Time Limit (ms)	Max Observed Network Delay (ms)
Safety Input	10	40.1	Reset
Safety Output	30	90.1	Reset

Configuration Ownership:

Configuration Signature:

ID: (Hex)

Date:

Time: ms

The Safety tab if the EDS does not contain a safety configuration.

The screenshot shows the 'Safety' tab with a table of connection parameters and configuration options, similar to the previous one but with different values.

Connection Type	Requested Packet Interval (RPI) (ms)	Connection Reaction Time Limit (ms)	Max Observed Network Delay (ms)
Safety Input	20	80.0	Reset
Safety Output	30	90.1	Reset

Configuration Ownership:

☒ Configuration Signature:

ID: (Hex)

Date:

Time: ms

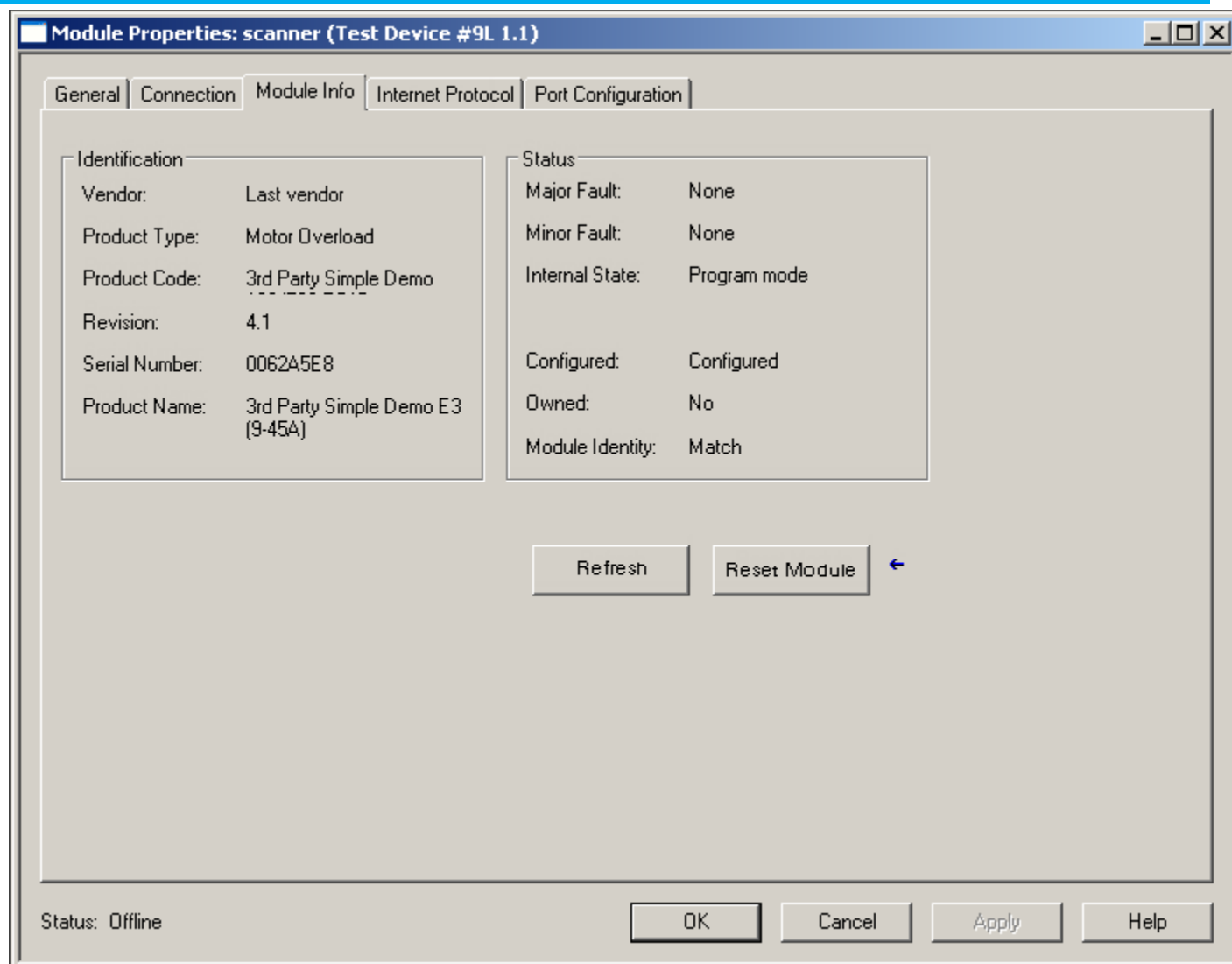
Module Info tab

The **Module Info** tab displays module and status information. Use this tab to:

- Display the module identity. See [Device] section.
- Display the module status.
- Reset the module to its power-up state.



Tip: The information on the **Module Info** tab comes directly from the module, and is not displayed if Logix Designer application is offline, or if currently creating a module. When online, the Logix Designer application indicates if there is a mismatch between the configured module in Logix Designer application and the online module.



See also

[\[Device\] section](#) on [page 48](#)

Safety Configuration tab

The Safety Configuration tab only displays if a safety connection is enabled. The Safety Configuration tab displays the parameters that are sent from a controller to a target module when an I/O connection is established. Use the **Safety Configuration** tab to configure and verify how a target module is configured. See *Target Config Format field*, *CfgAssembly keyword*, and *CfgAssemblyExa keyword* for more information.

ID	Name	Value	Units	Style	Description
1	Test Output 00 Mode	Pulse Test			Type of test output point.
2	Test Output 01 Mode	Pulse Test			Type of test output point.
3	Test Output 02 Mode	Not Used			Type of test output point.
4	Test Output 03 Mode	Not Used			Type of test output point.
5	Test Output 04 Mode	Not Used			Type of test output point.
6	Test Output 05 Mode	Not Used			Type of test output point.
7	Test Output 06 Mode	Not Used			Type of test output point.
8	Test Output 07 Mode	Not Used			Type of test output point.
9	Output Error Latch Time	100	x10 ms	Decimal	Latch any Safety Output error
10	Output 00 Mode	Not Used			Type of safety output point.
11	Output 01 Mode	Not Used			Type of safety output point.
12	Output 02 Mode	Not Used			Type of safety output point.
13	Output 03 Mode	Not Used			Type of safety output point.
14	Output 04 Mode	Not Used			Type of safety output point.
15	Output 05 Mode	Not Used			Type of safety output point.

See also

- [Target Config Format field](#) on [page 59](#)
- [CfgAssembly keyword](#) on [page 64](#)
- [CfgAssemblyExa keyword](#) on [page 64](#)

Internet Protocol tab

Use the **Internet Protocol** tab to view the TCP/IP object attribute values in the online module and make changes to the online module. See *[TCP/IP Interface Class] section*.

Module Properties: scanner (Test Device #9L 1.1)

General | Connection | Module Info | **Internet Protocol** | Port Configuration

Internet Protocol (IP) Settings
 IP settings can be manually configured or can be automatically configured if the network supports this capability.

☒ Manually configure IP settings
☐ Obtain IP settings automatically using DHCP

IP Settings Configuration

Physical Module IP Address: 10 . 88 . 46 . 49 Subnet Mask: 0 . 0 . 0 . 0
 Gateway Address: 0 . 0 . 0 . 0

Domain Name: Primary DNS Server Address: 0 . 0 . 0 . 0
 Host Name: Secondary DNS Server Address: 0 . 0 . 0 . 0

[Refresh communication.](#) ←

Status: Offline

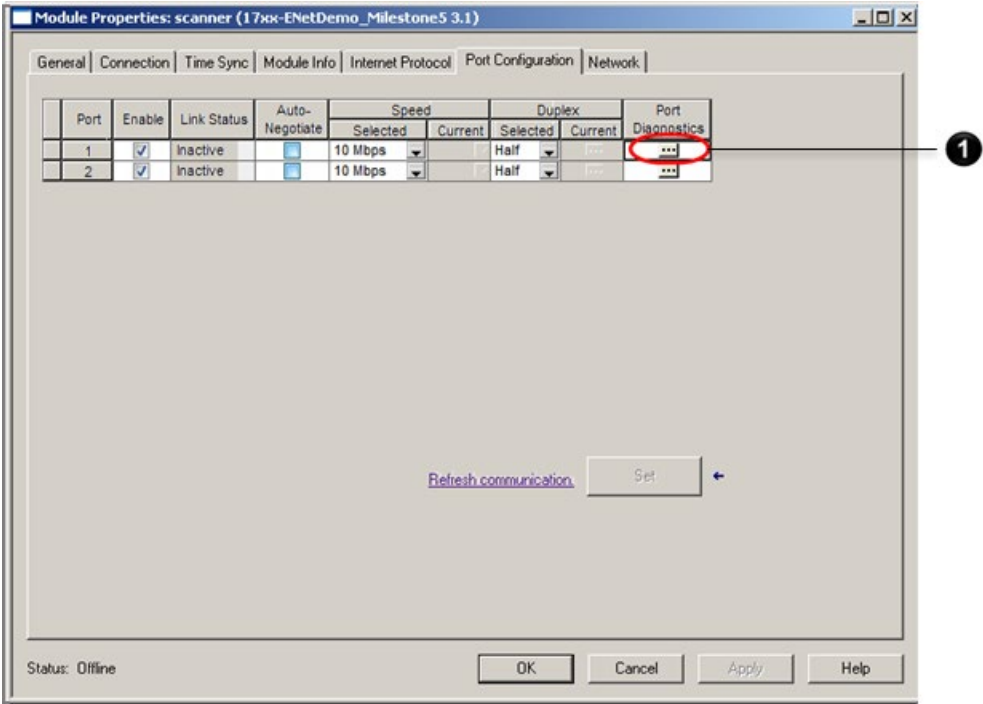
See also

[\[TCP/IP Interface Class\] section](#) on [page 75](#)

Port Configuration tab

Use the **Port Configuration** tab to view the Ethernet Link object attribute values in the online module and to change the online module. See *[Ethernet Link Class] section*.

In the **Port Configuration** tab, select the browse icon to open the **Port Diagnostics** dialog box.



Number	Description
1	Select the browse icon to open the Port Diagnostics dialog box.

See also

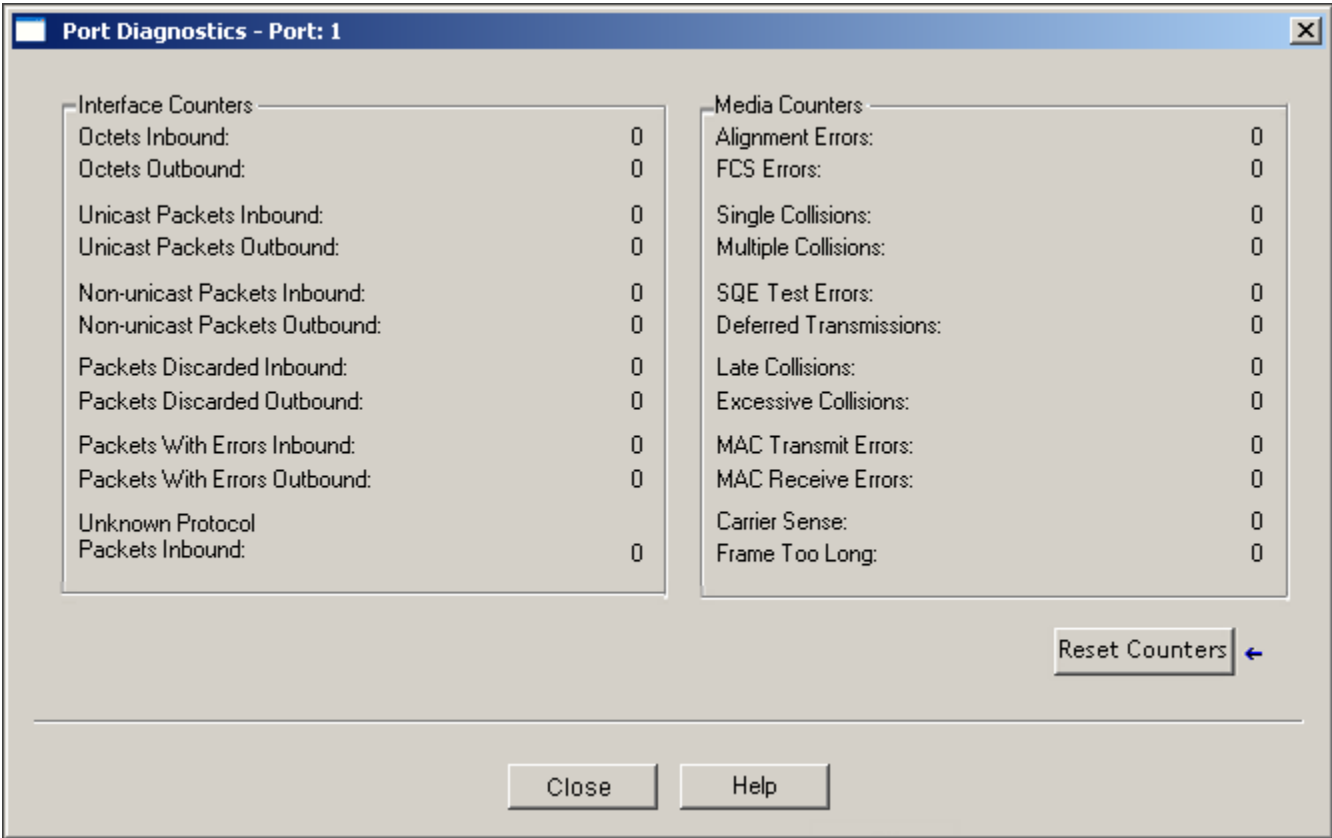
[\[Ethernet Link Class\] section](#) on [page 74](#)

[Port Diagnostics dialog box](#) on [page 41](#)

Port Diagnostics dialog box

Use the **Port Diagnostics** dialog box to view diagnostic information in the online module for the selected port and to reset the diagnostic counters. See *Port Configuration tab*.

Select [...] on the **Port Configuration** tab to display the **Port Diagnostics** dialog box.




See also

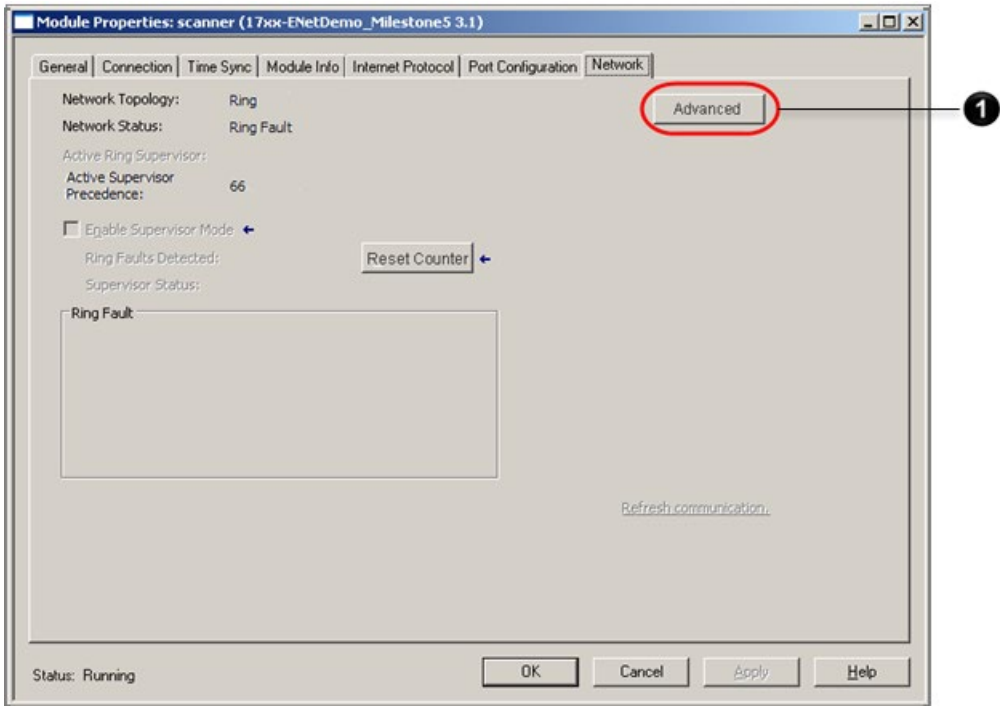
[Port Configuration tab](#) on [page 40](#)

Network tab

Use the **Network** tab to view the Device Level Ring object attribute values in the online module and to make changes to the online module. If the module supports the **Supervisor Mode** function, the **Network** tab includes a control to enable **Supervisor Mode**.

 Tip: The **Network** tab displays only for EtherNet/IP modules that support the Device Level Ring object. See *[DLR Class] section*.

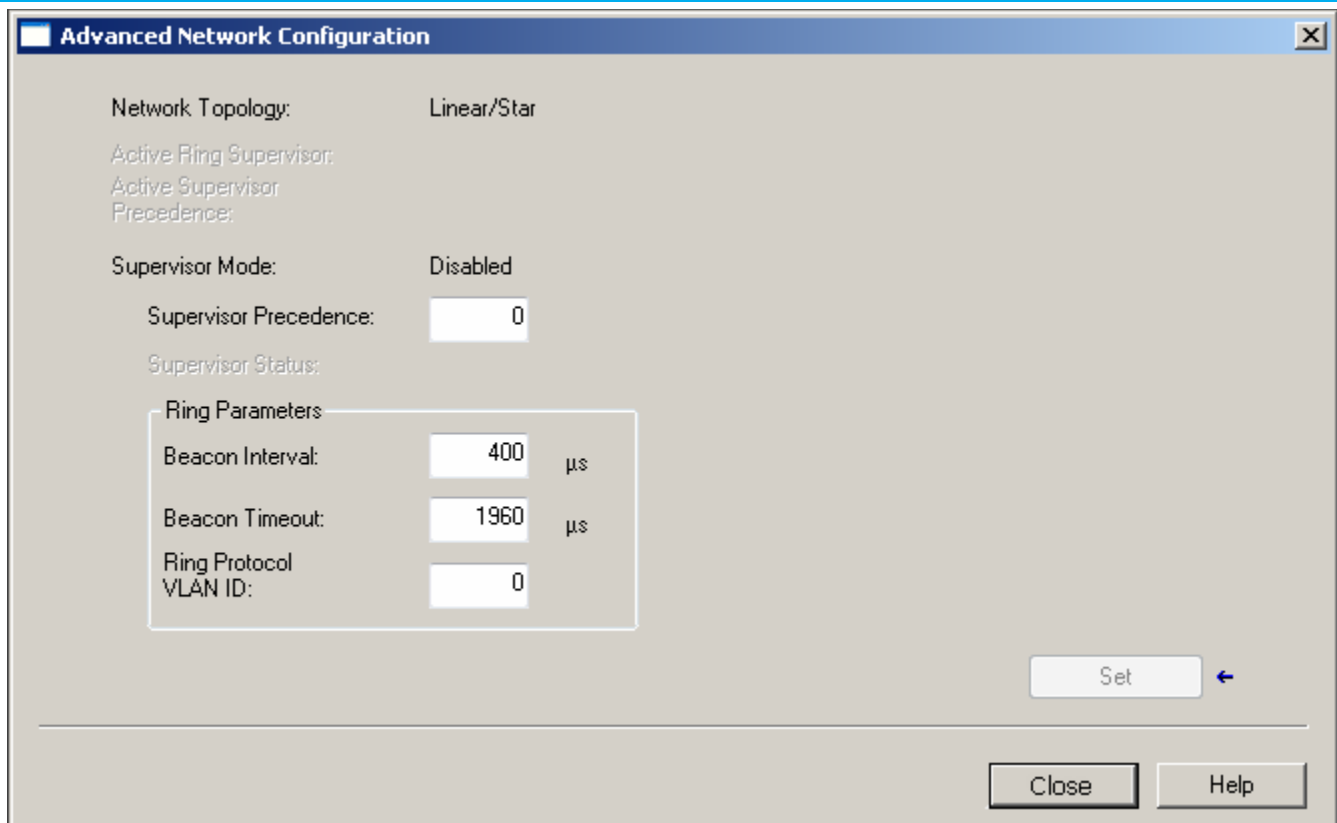
In the **Module Properties** dialog box, **Network** tab, select **Advanced** to display the **Advanced Network Configuration** dialog box.



Number	Description
1	In the Module Properties dialog box, Network tab, select Advanced to display the Advanced Network Configuration dialog box.

Advanced Network Configuration dialog box

The **Advanced Network Configuration** dialog box displays the network **Supervisor Mode** settings when the module is a Network Supervisor.



The **Advanced Network Configuration** dialog box is shown with the following settings:

- Network Topology:** Linear/Star
- Active Ring Supervisor:** Active Supervisor Precedence:
- Supervisor Mode:** Disabled
- Supervisor Precedence:** 0
- Supervisor Status:**
- Ring Parameters:**
 - Beacon Interval:** 400 μ s
 - Beacon Timeout:** 1960 μ s
 - Ring Protocol VLAN ID:** 0

Buttons at the bottom right: **Set** (with a left arrow), **Close**, and **Help**.

Select **Advanced** on the **Network** tab to display the **Advanced Network Configuration** dialog box.

See also

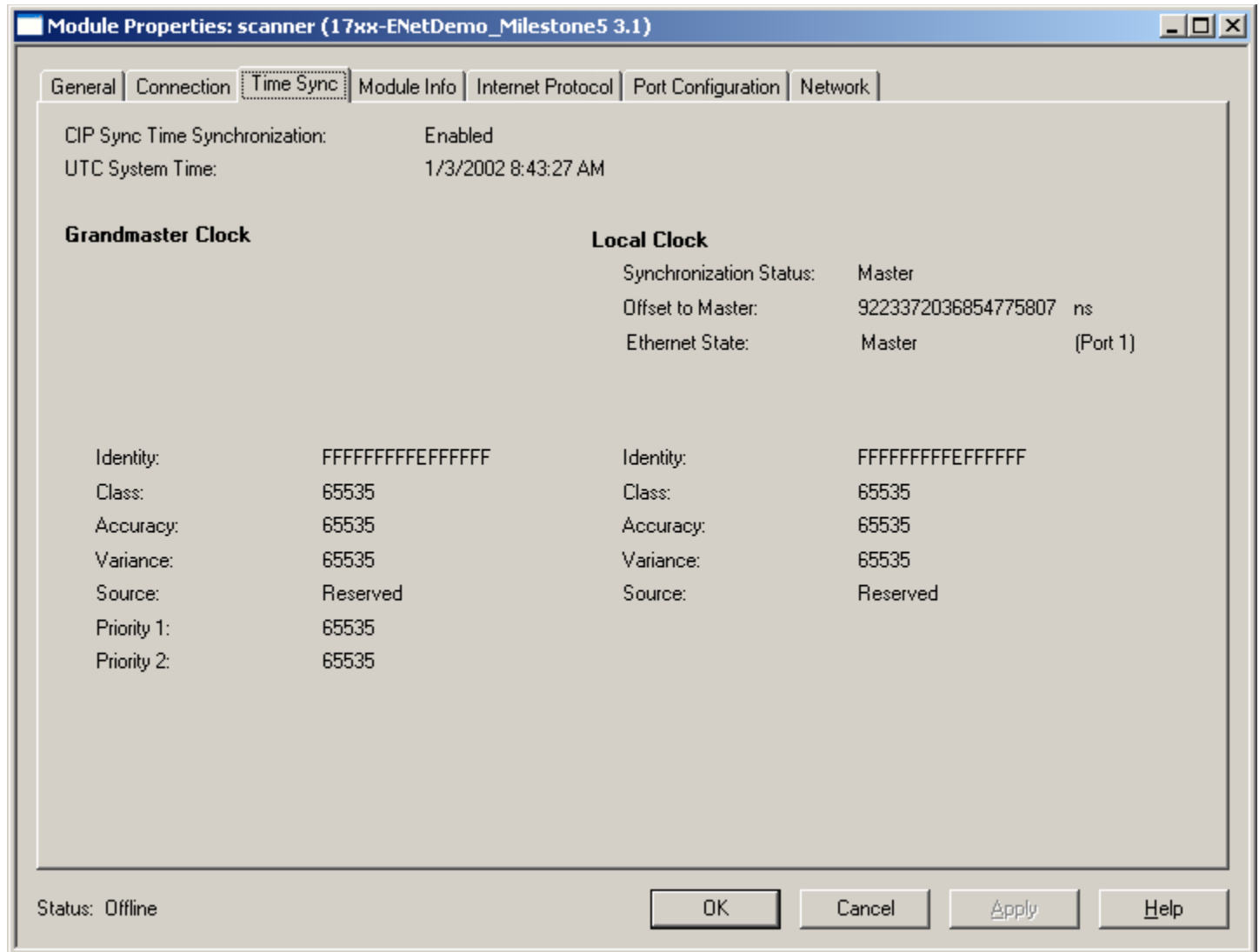
[Network tab](#) on [page 42](#)

Time Sync tab

Use the **Time Sync** tab to display the Time Sync object attribute values in the online module and to make changes to the online module. See *[Time Sync Class] section*.



Tip: The **Time Sync** tab is only displayed for modules that support the Time Sync object.



See also

[\[Time Sync Class\] section](#) on [page 77](#)

EDS file sections

This section describes the EDS sections and entry keywords used by the Logix Designer EDS AOP feature. This section covers how the keyword affects a CIP service (for example, the Forward_Open service) or affects the **Module Properties** dialog box. The *Example EDS Files* section describe several examples of EDS files.

EDS sections and keywords that are not mentioned in this document are not supported in the Logix Designer EDS AOP and are ignored.

The sections are:

- [\[File\] section](#) on [page 47](#)
- [\[Device\] section](#) on [page 48](#)
- [\[Device Classification\] section](#) on [page 53](#)
- [\[Connection Manager\] section](#) on [page 53](#)
- [\[ParamClass\] section](#) on [page 64](#)
- [\[Assembly\] section](#) on [page 65](#)
- [\[Params\] section](#) on [page 68](#)
- [\[Internationalization\] section](#) on [page 73](#)
- [\[Ethernet Link Class\] section](#) on [page 74](#)
- [\[TCP/IP Interface Class\] section](#) on [page 75](#)
- [\[DLR Class\] section](#) on [page 76](#)
- [\[Time Sync Class\] section](#) on [page 77](#)

See also

[Example EDS Files](#) on [page 83](#)

[File] section

The [File] section lists the File entry keywords and the keyword value.

```
[File]
DescText = "Test EDS Description";
CreateDate = 01-14-2011;
CreateTime = 14:26:47;
ModDate = 12-23-2011;
ModTime = 11:55:27;
Revision = 1.0;
EDSFileCRC = 0xn timer; $ number is different for every
EDS file
```

Revision, CreateDate, CreateTime, ModDate and ModTime keywords

The revision, creation, and modification information displays on the **About Module Profile** dialog box. The EDSFileCRC is a value calculated by the product developer and inserted in this entry. The value is unique to each EDS file. See CIP NETWORKS LIBRARY, Volume 5, Chapter 7 for details about this keyword.

See also

[About Module Profile dialog box](#) on [page 29](#)

[Device] section

The [Device] section lists the Device entry keywords and the keyword value.

```
[Device]
VendCode = 65500;
VendName = "Test Vendor 65500";
ProdType = 0xFFFF;
ProdTypeStr = "Generic Device";
ProdCode = 9;
MajRev = 1;
MinRev = 1;
ProdName = "3 Assemblies with Structure";
Catalog = "TestDevice#9";
```

Catalog and ProdName keywords

The Catalog string, followed by the ProdName string, are concatenated for display in the **Module Properties** dialog box, **General** tab, **Type** box. If the optional Catalog keyword is not present, only the value of the ProdName keyword shows.



Tip: Use a unique combination strings for Catalog and ProdName for all modules from a vendor to distinguish them from each other.

See also

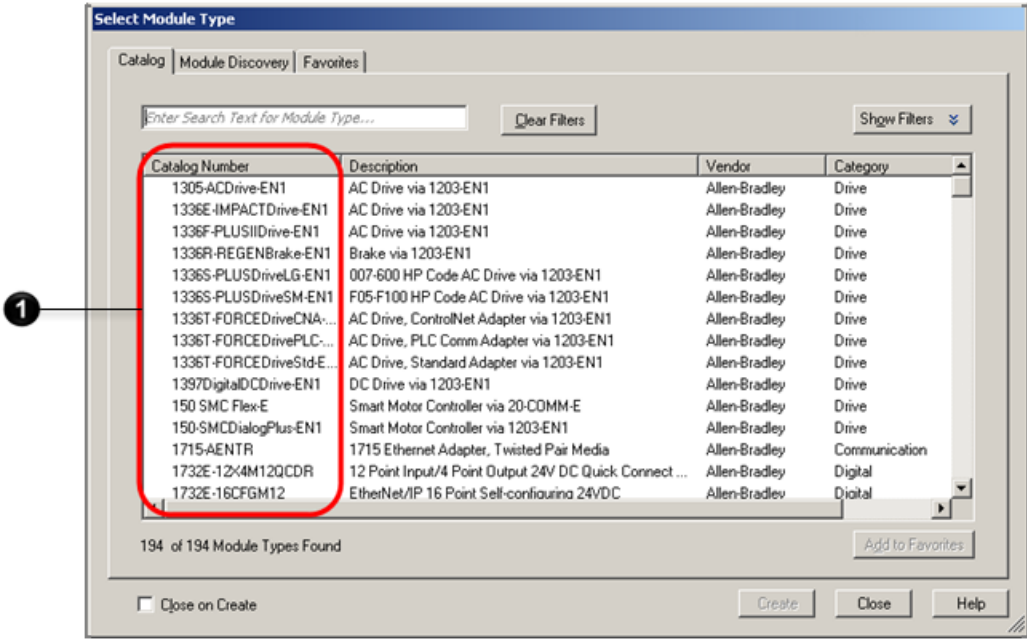
[General tab](#) on [page 27](#)

[Catalog keyword](#) on [page 48](#)

[ProdName keyword](#) on [page 74](#)

Catalog keyword

The Catalog keyword value displays in the **Select Module Type** dialog box, **Catalog** and **Favorites** tabs, **Catalog Number** column. If the optional Catalog keyword is omitted, a string of the form xxxx_yyyy_zzzz (xxxx is hex VendCode, yyyy is hex ProdType, zzzz is hex ProdCode) displays instead.



Number	Description
1	Select Module Type dialog box, Catalog Number column contains a list of catalog keyword values.

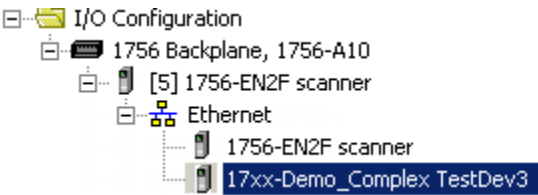
Important:

Module developers should include the Catalog keyword.

Do not use spaces in the Catalog keyword. Space characters cause problems with the **Path** control found in the **Message Configuration** dialog box, **Communication** tab (the **Message Configuration** dialog box is invoked from the MSG instruction).

The Catalog keyword is included in Catalog Organizer, Data Types tree, Module- Defined data type name. Use alpha-numeric characters and the underscore (.) character. Other characters are removed, except dash (-), which is converted to underscore (.).

The Catalog keyword value displays in the **Controller Organizer, I/O Configuration** tree module.

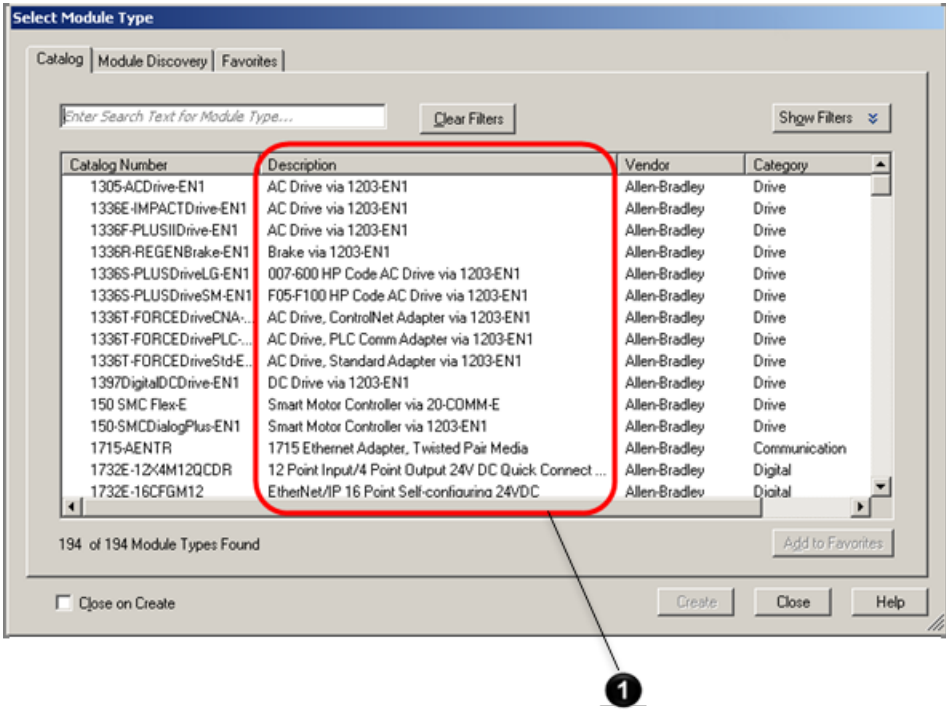


See also

[Catalog and ProdName keywords](#) on [page 48](#)

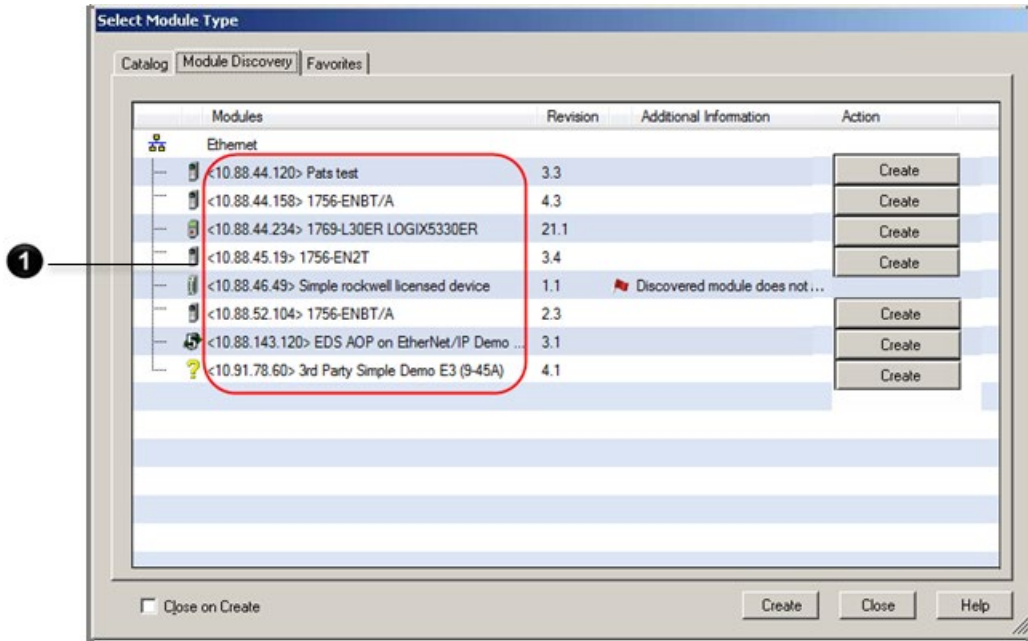
ProdName keyword

The ProdName keyword value displays in the **Select Module Type** dialog box, **Catalog** and **Favorites** tabs, **Description** column.



Number	Description
1	Select Module Type dialog box, Description column contains a list of ProdName keyword values.

The ProdName keyword value also displays in the **Select Module Type** dialog box, **Module Discovery** tab, **Modules** column.



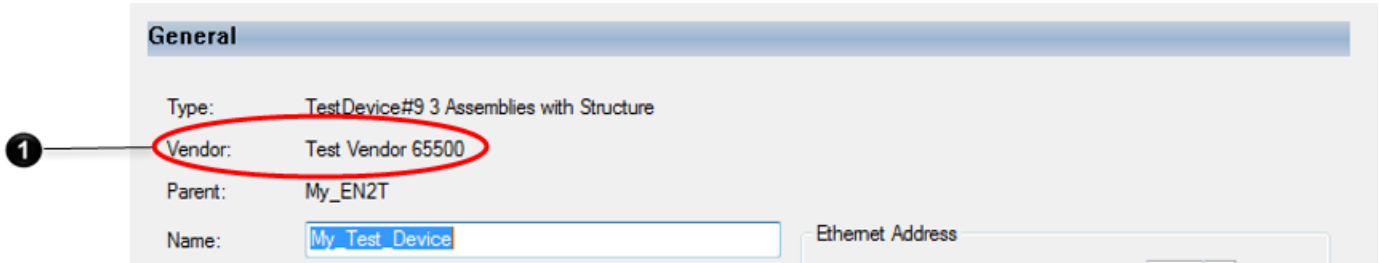
Number	Description
1	Select Module Type dialog box, Module Discovery tab, Modules column contains a list of ProdName keyword values.

VendName keyword

If an ODVA-defined vendor string is not known for the VendCode, the VendName from the first EDS file registered for the VendCode is used; otherwise, the ODVA-defined vendor string is used.

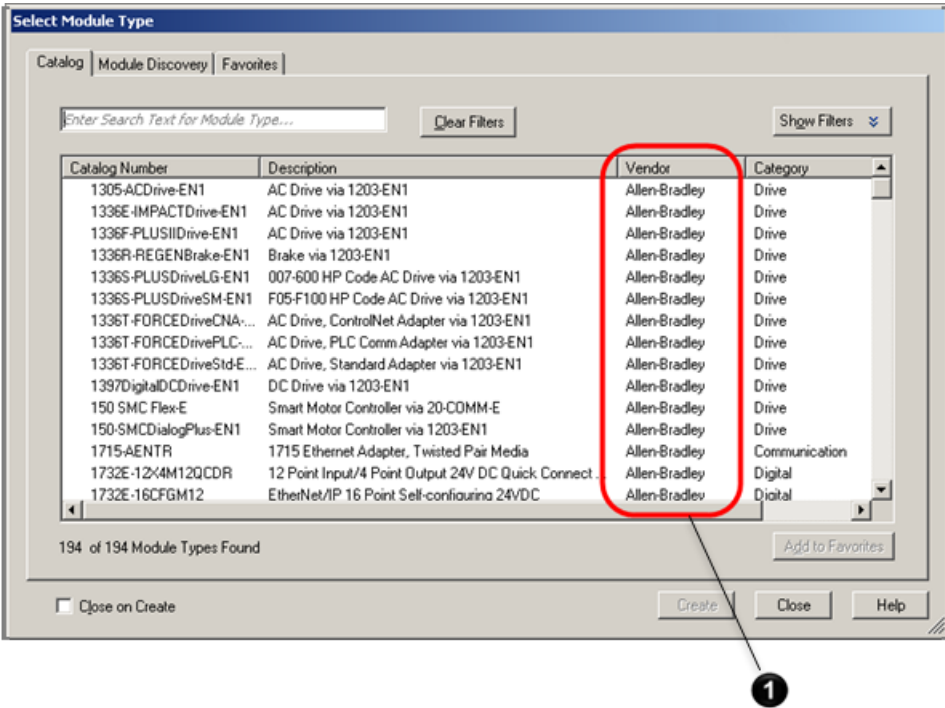
IMPORTANT Once the VendName is set by registering the first EDS file with that VendName, there is no mechanism to change the VendName.

The Rockwell Automation-defined VendName keyword value displays in the **Module Properties** dialog box, **General** tab, **Vendor** area.



Number	Description
1	Module Properties dialog box, General tab, Vendor area lists the VendName keyword value.

The Rockwell Automation-defined VendName keyword value also displays in the **Select Module Type** dialog box, **Catalog** and **Favorites** tabs, **Vendor** column.



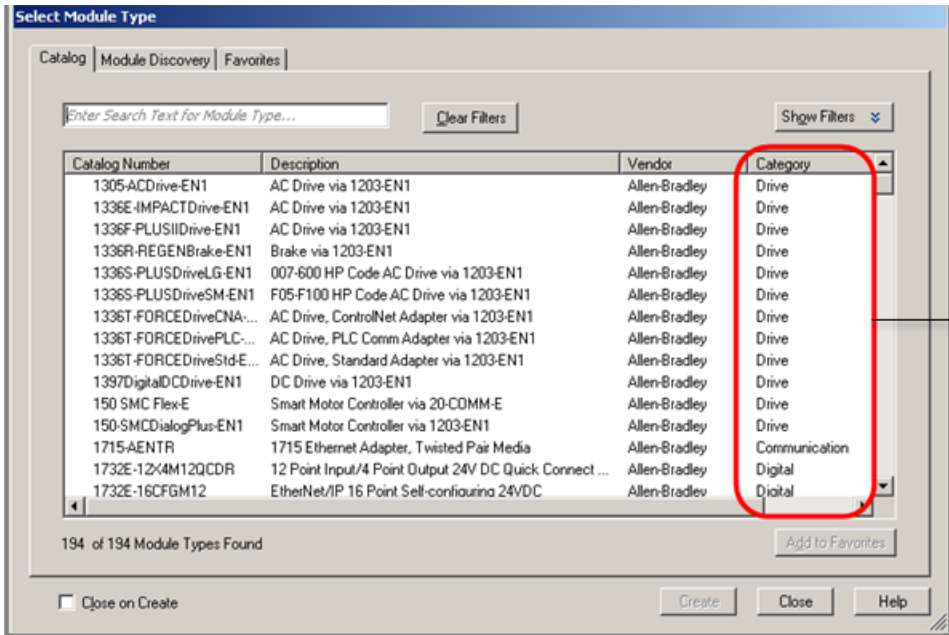
Number	Description
1	Select Module Type dialog box, Catalog and Favorites tabs, Vendor column also lists the VendName keyword values.

See also

[General tab](#) on [page 27](#)

ProdTypeStr keyword

The ProdTypeStr keyword value displays in the **Select Module Type** dialog box, **Catalog** and **Favorites** tabs, **Category** column.



Number	Description
1	Select Module Type dialog box, Catalog and Favorites tabs, Category column lists the ProdTypeStr keyword value.

If an ODVA-defined profile string is not known for publicly defined ProdType values, the ProdTypeStr from the first EDS registered for the ProdType is used; otherwise, the ODVA-defined profile name is used. See *THE CIP NETWORKS LIBRARY Volume 1, Common Industrial Protocol (CIP™)*, table 6- 7.1

For vendor-specific ProdType values, the ProdTypeStr from the first EDS registered for the VendCode/ProdType is used.

IMPORTANT Once ProdTypeStr is set by registering the first EDS with that ProdTypeStr, there is no mechanism to change ProdTypeStr.

MajRev keyword

The MajRev keyword value is available for selection in the **Module Definition** dialog box, **Major Revision** control.

See also

[Module Definition dialog box](#) on [page 30](#)

MinRev keyword

The MinRev value is used as the default value of **Minor Revision** in the **Module Definition** dialog box.

See also

[Module Definition dialog box](#) on [page 30](#)

**[Device Classification]
section**

The [Device Classification] section includes the ClassN keyword.

```
[Device Classification]
Class1 = EtherNetIP;
Class2 = Safety;
```

ClassN keyword

If the module is an EtherNet/IP module, a ClassN keyword must exist in the [Device Classification] section. A keyword example is:

```
Class1 = EtherNetIP;
```

**[Connection Manager]
section**

The *CIP NETWORKS LIBRARY, Volume 5, Chapter 7*, has restrictions and requirements for some of the values used in this section when the connection entry is describing a CIP Safety connection. Please refer to the specification for these restrictions and requirements. Any further restrictions and requirements imposed by EDS AOP are noted where appropriate.

```
[Connection Manager]
Connection1 =
0x04030002,
0x44640405,
Param1,,Assem101,          $ O->T RPI, size, format
Param1,,Assem102,          $ T->O RPI, size, format
,,                          $ proxy config size,
format
,Assem100,                  $ target config size,
format
"I/O Connection",          $ Connection Name
"",                          $ help string
"20 04 24 64 2C 65 2C 66";  $ Path
MaxSafetyConnections = 4;
MaxSafetyInputCnxns = 4;
MaxSafetyOutputCnxns = 1;
DefaultSafetyConnections = 3,6;
```

Extra comment lines have been removed.

See also

[ConnectionN keyword](#) on [page 74](#)

[ConnectionN fields](#) on [page 54](#)

ConnectionN keyword

The ConnectionN keyword identifies a CIP connection supported by the module.

Connections are added by default if no variables (for example, ParamN) exist in the **Path** field and the connection does not conflict with an existing

connection. See *Path field* for rules describing connections that can exist together. All Exclusive- Owner connections are added, then Input-Only connections are added, and then Listen-Only connections are added.

ConnectionN entries that do not meet the criteria listed in this section are ignored.



Tip: A "Ping" ConnectionN entry is also supported. See *THE CIP NETWORKS LIBRARY, Volume 1*. A Ping Connection allows adding a module with no connections configured. The controller periodically sends a Null Forward_Open service. See Null Forward Open in *THE CIP NETWORKS LIBRARY, Volume 1, Chapter 3* to verify the correct module is present.

These keywords are required for devices that support CIP Safety connections. The values used for these keywords are dependent on device features and are shown for example only. See *CIP NETWORKS LIBRARY, Volume 5, Chapter 7* for further information.

```
MaxSafetyConnections,
MaxSafetyInputCnxns,
MaxSafetyOutputCnxns,
DefaultSafetyConnections,
```

See also

[ParamN keyword](#) on [page 68](#)

[ParamN fields](#) on [page 68](#)

[Path field](#) on [page 63](#)

ConnectionN fields

The ConnectionN fields include:

- Trigger and Transport
- Connection Parameters
- T->O and O->T RPI,
- T->O and O->T Size
- T->O and O->T Format
- Proxy Config Size and Format
- Target Config Size
- Target Config Format
- Connection Name String
- Help String
- Path

Trigger and Transport field

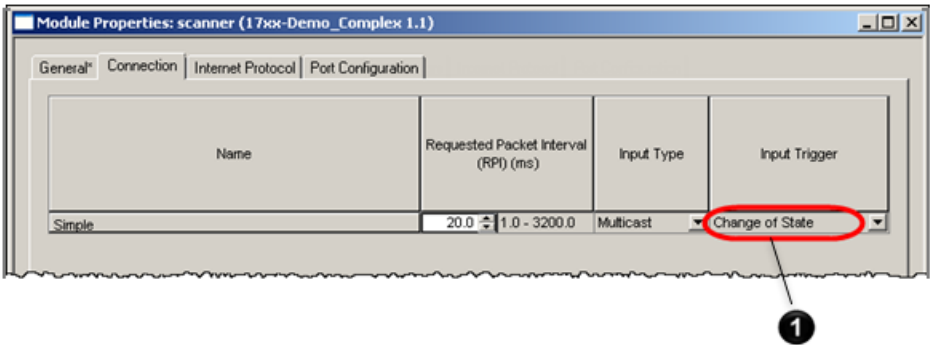
The Trigger and Transport field consist of Transport class, Trigger, Application type, and Direction.

- Transport class

Must indicate Class 1 support (may indicate others).

The Transport Class field value of the Transport Class and Trigger parameter sent in the Forward_Open request is always set to 1.

- Trigger
The Trigger displays on the **Modules Properties** dialog box, **Connection** tab, **Input Trigger** column. Select the appropriate trigger (**Cyclic**, **Change of State**, or **Application**).



Number	Description
1	Module Properties dialog box, Connection tab, Input Trigger column lists the Trigger keyword values to choose.

The user-selected Input Trigger value is the Forward_Open Transport Class and Trigger parameter.

- Application type
Must indicate one of **Listen-Only**, **Input-Only** or **Exclusive-Owner**. Redundant-Owner is not supported.
- Direction
The Direction value is the Forward_Open Transport Class and Trigger parameter, Dir field value. Set this to Client.

See also

[Connection tab](#) on [page 34](#)

Connection Parameters field

T->O and O->T fixed/variable size

O->T Real time transfer format

T->O Real time transfer format

The Connection Parameters field consist of T→O and O→T fields, Proxy Config and Format field, Target Config Size field, Target Config Format field, Connection Name String field, Help String field, and Path field.

Only supports fixed size. The T→O and O→T fixed size supported bits must be set. The T→O and O→T variable size supported bits are ignored.

Must indicate one:

- Connection is pure data and modeless – the controller does not include a 32 bit run/idle header.
- Heartbeat – no application data.
- 32 bit run/idle header – the controller includes a 32 bit run/idle header.
- Safety (for CIP Safety connections)

Must indicate one:

- Connection is pure data and modeless – a 32 bit run/idle header is not

included in the Input Module-Defined data type/tag definition.

- Heartbeat – no application data.
- 32 bit run/idle header – a 32 bit run/idle header is included in the Input Module-Defined data type or tag definition.
- Safety (for CIP Safety connections)

O->T Connection type

Must indicate POINT2POINT. May also specify NULL. For more information, see below for T→O and O→T Connection types with NULL type selected.

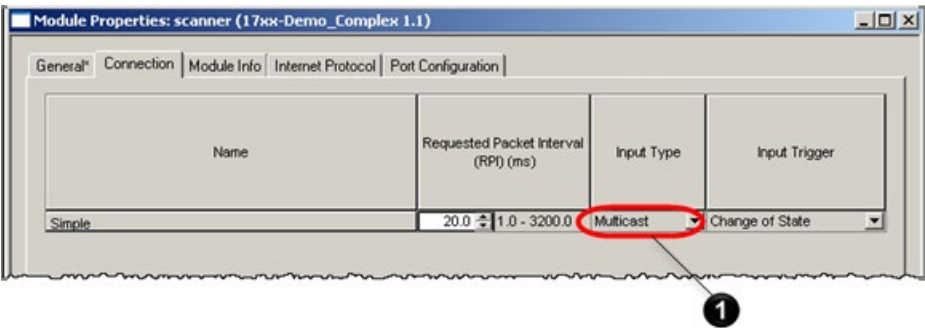
Point to Point is the Forward_Open O→T Network Connection parameter, Connection Type field value.

T->O Connection type

The T→O Connection type must indicate POINT2POINT, MULTICAST, or both; may also specify NULL. For more information, see below for T→O and O→T Connection types with NULL type selected.

The T→O Connection type displays in the **Input Type** column of the **Connection** tab. Users select **Unicast** or **Multicast** as the T→O Connection type.

The user selection of Point to Point or Multicast is the Forward_Open T→O Network Connection parameter, Connection Type field value.



Number	Description
1	Module Properties dialog box, Connection tab, Input Type column contain the T→O Connection type selections Unicast and Multicast.

T->O and O->T Connection types with NULL type selected

The module can be reconfigured by a NULL Forward_Open service (while the I/O connection is open) if these conditions are true:

- Both T→O and O→T NULL connection types are indicated.
- The Write Allowed bit group in the configuration assembly descriptor has a value of 0 (assembly may or may not be written when I/O connection is open) or the Write Allowed bit group in the configuration assembly descriptor has a value of 1 (assembly can be written when any I/O connection is open).

A NULL Forward_Open service is sent when a MSG instruction, Module Reconfiguration Message Type is executed.

T→O and O→T Priority

The T→O and O→T Priority must indicate either both T→O and O→T HIGH priority or both T→O and O→T SCHEDULED priority. Both HIGH and SCHEDULED can be indicated.

High or Scheduled (Scheduled if both are specified) is the Forward_Open T→O and O→T Network Connection parameters, Priority field values.

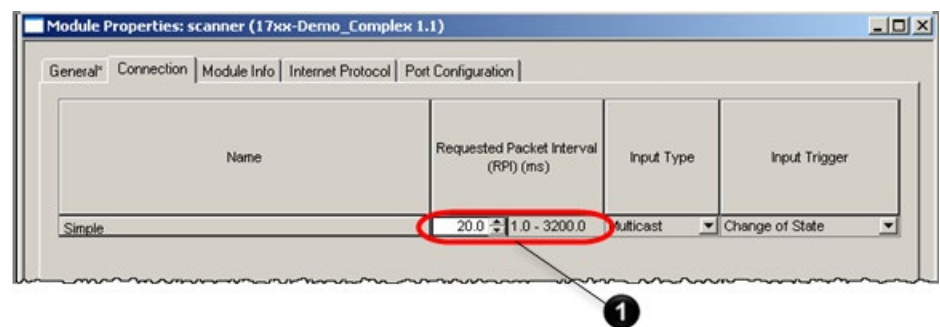
T→O and O→T RPI fields

If both fields are empty, the default RPI for the bus type will be selected and the default range for the bus type will be enabled. For Ethernet bus – minimum RPI is 1.0ms, maximum RPI is 3200.0ms, and default RPI is 20.0ms.

If a UDINT value or ParamN is specified, the intersection of the T→O and O→T value(s) and bus minimum and maximum values are used to generate the minimum and maximum RPI values. If no intersection exists, cannot choose an RPI value and as a consequence, cannot add the connection via EDS AOP.

Important: Enumerated ParamNs are not supported for RPIs.

The RPI displays in the **Modules Properties** dialog box, **Connection** tab, **Requested Packet Interval (RPI) (ms)** column.



Number	Description
1	Module Properties dialog box, Connection tab, Requested Packet Interval (RPI) (ms) column contains the RPI value.

The user selected RPI value is the Forward_Open T→O and O→T RPI parameter values.



Tip: If the O→T production is heartbeat, then a longer O→T RPI value is calculated and used to reduce unneeded traffic.

See also

[Connection tab](#) on [page 34](#)

T→O and O→T Size fields

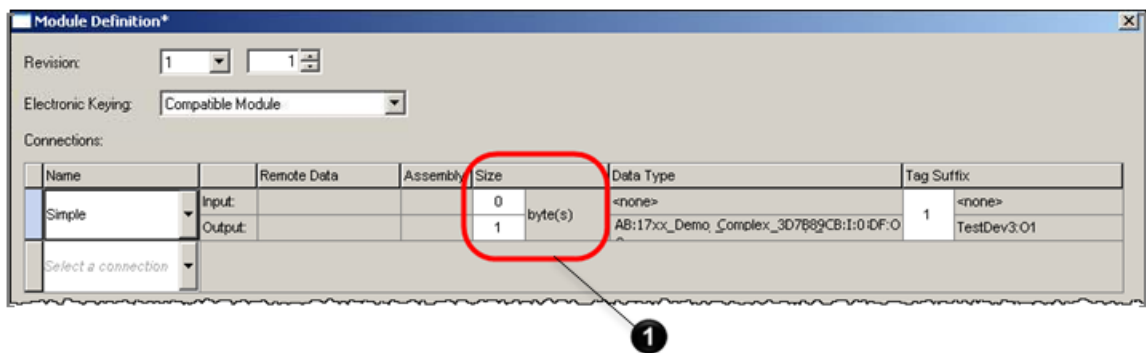
The T→O and O→T Size field must be either UINT or empty.

If the size field is empty, the size of the item specified in the corresponding

format field size plus sequence count size (2) and optional 32 bit run/idle header size (4) is the Forward_Open T→O and O→T Network Connection parameter, Connection Size values.

If the size field is a UINT, then the size field value plus sequence count size (2) and optional 32 bit run/idle header size (4) is the Forward_Open T→O and O→T Network Connection parameter, Connection Size values.

The T→O and O→T Size displays in the **Module Definition** dialog box, **Size** column.



Number	Description
1	Module Definition dialog box, Size column contains T→O and O→T Size value.

If the size field is a ParamN, then the user selected size plus sequence count size (2) and optional 32 bit run/idle header size (4) are the Forward_Open T→O and O→T Network Connection parameter, Connection Size values. If the size field is a parameter, either range or enumeration can be supported, but not both.

See also

[Module Definition dialog box](#) on [page 30](#)

T->O and O->T Format fields

If the format field is empty, no Input or Output Module-Defined data type or tag is created.

IMPORTANT

To create Input/Output Module-Defined data types and tags, specify the format field even when the data format is dynamic.
In this case:

- An AssemN must be specified as the format with the Allow Value Edit bit of the Descriptor of this AssemN set, and
- The size of the AssemN must be specified by the AssemN Size field, or
- A single Member Size/Member Reference pair with an empty Member Reference field.

If the format specifies one or more ParamN entries, a Module-Defined data type and tag is created if the format results in one or more ParamNs.

IMPORTANT	<p>The Input Module-Defined data types created for T→O includes a BOOL member named ConnectionFaulted in the first 32 bits.</p> <p>The Input Module-Defined data type created for T→O includes a BOOL member named RunMode in the first 32 bits if the T→O Real Time transfer format is 32-bit run/idle header.</p> <p>The Module-Defined data types created have a data member of type SINT, INT, DINT or REAL array as chosen by the user in the Module Definition dialog box, Size column.</p> <ul style="list-style-type: none"> • The type choice of INT is only allowed when the I/O sizes (in bytes) is divisible by 2. • The type choices of DINT and REAL are allowed when the I/O sizes (in bytes) are divisible by 4. • If the size fields are ParamNs, then the maximum values of the ParamN are used to determine if INT, DINT and REAL choices are available. • If the maximum values are divisible by 2, the choice of SINT and INT are available. • If the maximum values are divisible by 4 then the choices of SINT, INT, DINT and REAL are available.
------------------	---

See also

[Member Size field](#) on [page 67](#)

[Member Reference field](#) on [page 67](#)

[Module Definition dialog box](#) on [page 30](#)

Proxy Config Size and Format fields

Target Config Size field

If not empty, the connection is ignored.

If the size field is empty, the size of the item specified in the corresponding format field size is the Forward_Open Connection_Path parameter data segment size value (converted from bytes to words and rounded up).

If the size field is a UINT, then the size field is the Forward_Open Connection_Path parameter data segment size value (converted from bytes to words and rounded up).

If the size field is a ParamN, the default value of the ParamN is the Forward_Open Connection_Path parameter data segment size value (converted from bytes to words and rounded up).

Target Config Format field

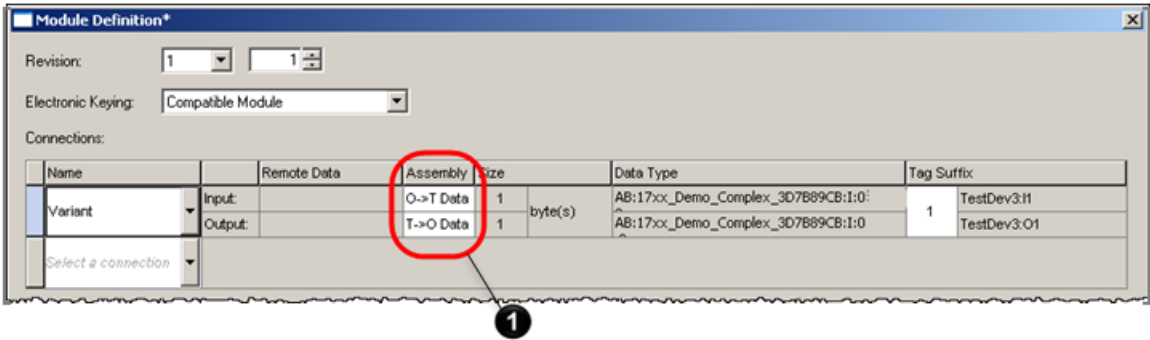
If the format field is empty and the Target Config Size field contains a non-null size, the Forward_Open Connection_Path parameter data segment is filled with zeros.

If the format field is a ParamN or AssemN, the value of the ParamN or AssemN is the Forward_Open Connection_Path parameter data segment data values.

If the format specifies one or more ParamN entries, a Module-Defined data type and tag is created if the format results in one or more ParamNs.

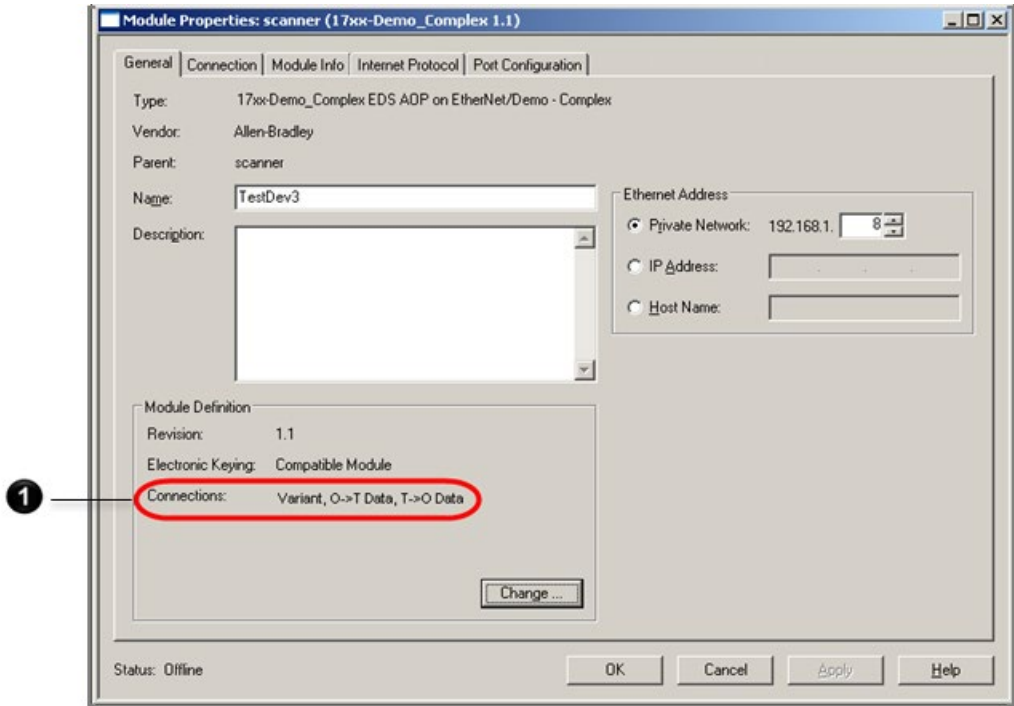
Some configuration parameters are not included in the configuration Module- Defined data type and tag. The parameters not included are:

- I/O assembly variant selectors – if a configuration parameter is an I/O variant selector, the variant selections are displayed on the **Module Definition** dialog box, **Assembly** column.



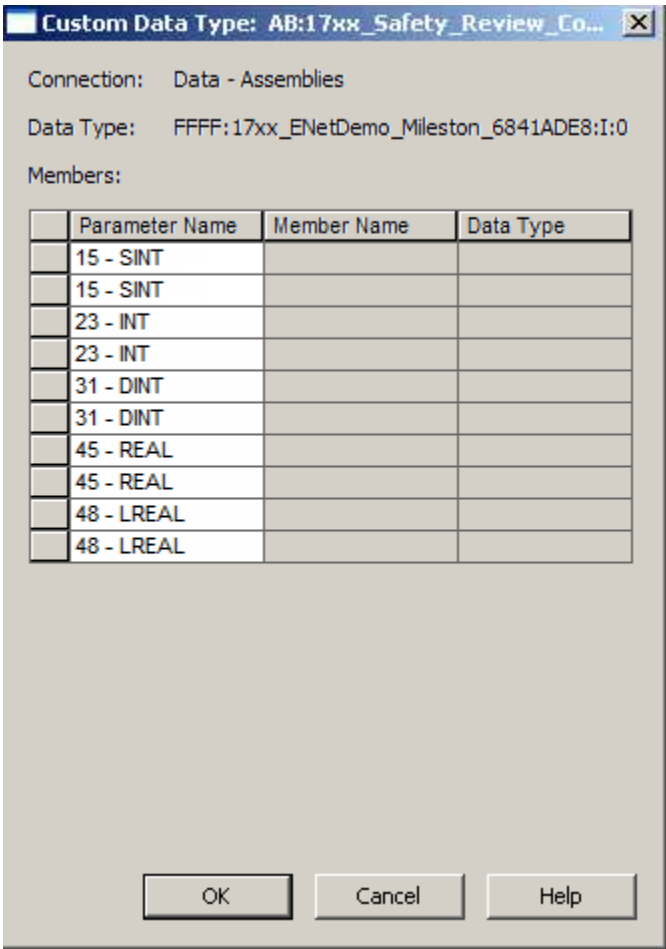
Number	Description
1	If a configuration parameter is an I/O variant selector, the variant selections are shown in the Module Definition dialog box, Assembly column.

- I/O assembly variant selectors – if a configuration parameter is an I/O variant selector, the variant selections show in the **Module Properties** dialog box, **General** tab, **Module Definition** group, **Connections** area.

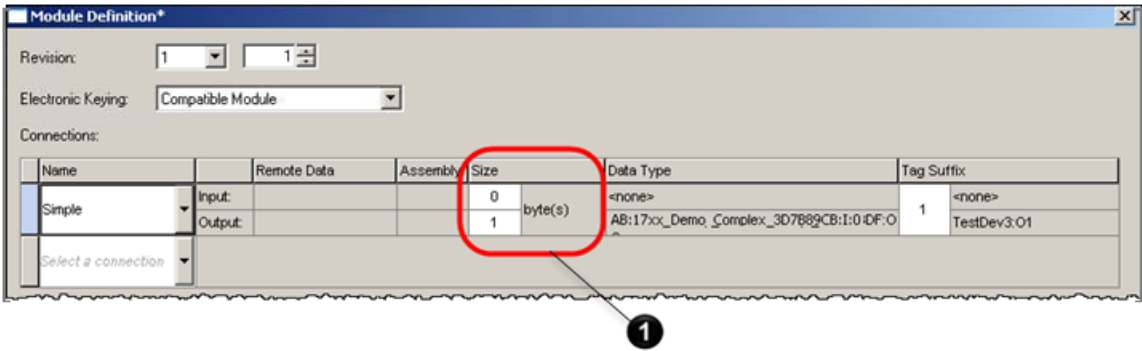


Number	Description
1	If a configuration parameter is an I/O variant selector, the variant selections show in the Module Properties dialog box, General tab, Module Definition group, Connections area.

- Indirect I/O parameter – if a configuration parameter is an indirect I/O parameter, the pointed to name displays on the **Custom Data Type** dialog box (launched from **Module Definition** dialog box).



- I/O size parameter – if a configuration parameter is used to specify a variable I/O size, it displays on the **Module Definition** dialog box, **Size** column.



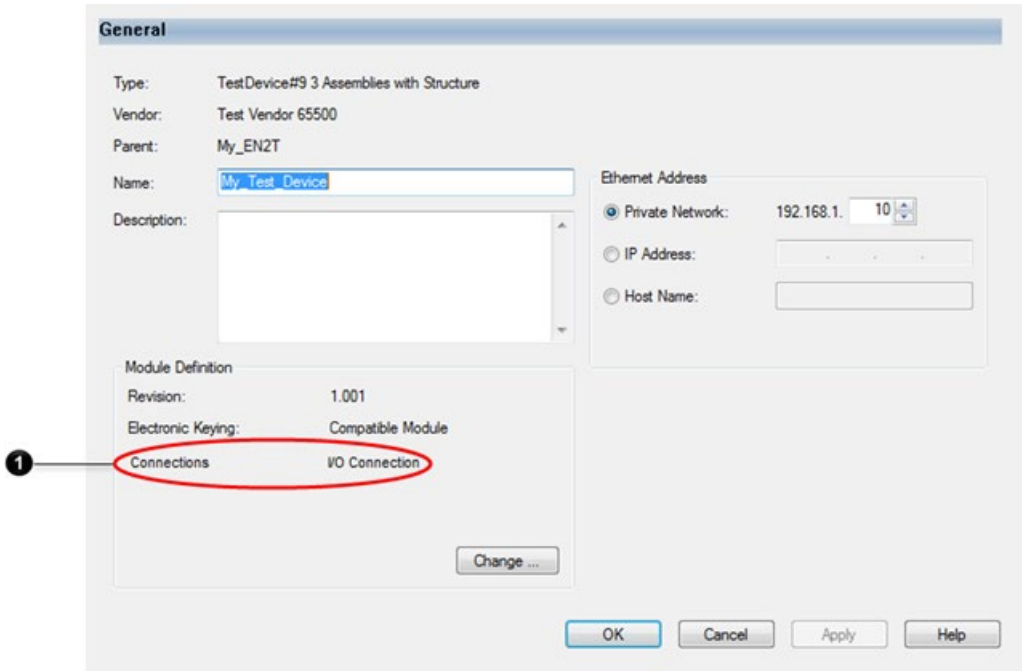
Number	Description
1	If a configuration parameter is used to specify a variable I/O size, it displays on the Module Definition dialog box, Size column.

See also

- [Target Config Size field](#) on [page 59](#)
- [Module Definition dialog box](#) on [page 30](#)
- [Custom Data Type dialog box](#) on [page 32](#)

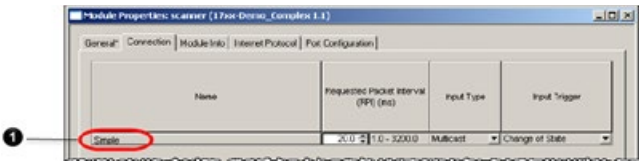
Connection Name String field

The Connection Name String displays in the **Module Properties** dialog box, **General** tab, **Module Definition** group, **Connections** area.



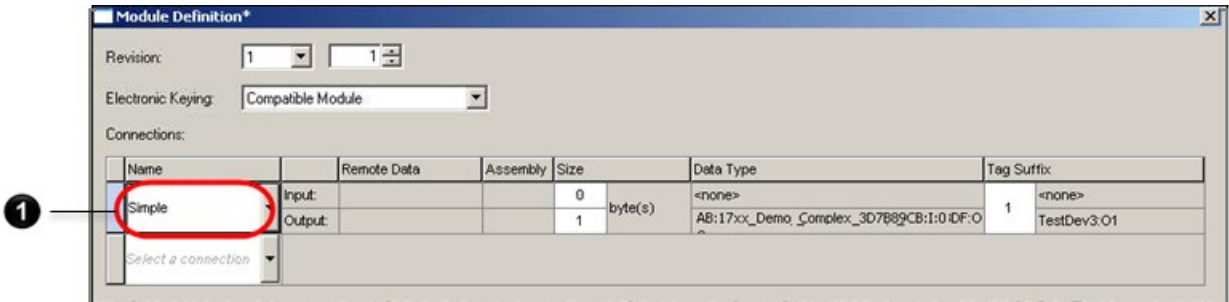
Number	Description
1	The Connection Name String displays in the Module Properties dialog box, General tab, Module Definition group, Connections area.

The Connection Name string displays in the **Module Properties** dialog box, **Connection** tab, **Connections** grid.




Number	Description
1	The Connection Name string displays in the Module Properties dialog box, Connection tab, Connections grid.

The Connection Name string displays in the **Module Definition** dialog box, **Connections** grid.



Number	Description
1	The Connection Name string displays in the Module Definition dialog box, Connections grid.

 Tip: Limiting the name strings to 22 characters is recommended. This allows displaying the chosen connection name without resizing the dialog box.

See also

- [General tab](#) on [page 27](#)
- [Connection tab](#) on [page 34](#)
- [Module Definition dialog box](#) on [page 30](#)

Help String field
Path field

Ignored.

The ConnectionN path field is the part of the Forward_Open Connection_Path parameter immediately prior to the data segment (if a data segment exists), or at the end of the Connection_Path parameter (if the data segment does not exist).

If multiple connections exist with the same connection path, only the ConnectionN entry with the lowest N will be available in the EDS AOP. All other connections with identical paths are discarded during the EDS install process. This is indicated as a warning when using the EDS Hardware Installation Tool to register the EDS file. See *EDS file installation*.

If two connections are available that are identical except for their configuration (one with configuration, one without), they must have different paths; it is recommended that the connection without configuration does not include a configuration path.

The Path field contains up to three application paths. The three application paths are used to limit the set of connections that can be configured simultaneously.

IMPORTANT If the module has additional rules that limit the set of connections that can open simultaneously, the module developer must document these rules for the user.

- Configuration application path
If a configured ConnectionN entry includes non-empty Target Config size/format fields, then all other ConnectionN entries, which include a different Configuration application path and non-empty Target Config size/format fields, are not available to add.
- Originator to Target application path
If a configured ConnectionN entry specifies an O→T Real time transfer format that is not heartbeat, then all other ConnectionN entries, which include the same Originator to Target application path, are not available to add.
- Target to Originator application path
If a configured ConnectionN entry specifies a T→O Real time transfer format that is not heartbeat, then all other ConnectionN entries, which include the same Target to Originator application path, are not available to add.

The ConnectionN Path field can contain variables ([ParamN] entries) in the O→T and T→O application path. When the connection is configured in the Logix Designer application **Module Definition** dialog box, set the ParamN value in the **Remote Data** column.

The ConnectionN path field can contain symbols (ANSI_SYMBOL entries) in the O→T and T→O application path. When the connection is configured in the Logix Designer application **Module Definition** dialog box, set the tag value in the **Remote Data** column.

The ConnectionN Path field can contain a Safety ASYNC field. This field must be empty for consuming connections and is ignored for producing connections.

The ConnectionN Path field can contain Max Consumers. This field is not used.

See also

[EDS file installation](#) on [page 14](#)

[Target Config Size field](#) on [page 59](#)

[Target Config Format field](#) on [page 59](#)

[Module Definition dialog box](#) on [page 30](#)

[ParamClass] section

The [ParamClass] section includes the CfgAssembly, CfgAssemblyExa, Descriptor, and SafetyCfgAssembly keywords.

CfgAssembly keyword

Ignored.

CfgAssemblyExa keyword

Ignored.

Descriptor keyword SafetyCfgAssembly keyword

Ignored.

Required.

See *CIP NETWORKS LIBRARY, Volume 5, Chapter 7*.

[Assembly] section

The [Assembly] section includes the AssemN/AssemExaN and VariantN/VariantExaN keywords.

AssemN/AssemExaN keyword

See this section for:

- Supported uses.
- Nested assemblies.
- AssemN/AssemExaN fields.

Supported uses

AssemN/AssemExaN keywords have these supported uses:

- ConnectionN O→T Format
- ConnectionN T→O Format
- ConnectionN Target Config Format

For more information about these supported uses, See *ConnectionN keyword*.

See also

[ConnectionN keyword](#) on [page 53](#)

Nested assemblies

Assemblies can be nested. The result of nested assemblies is nested structures.

AssemN/AssemExaN fields

The AssemN/AssemExaN fields include:

- Name
- Path
- Size
- Descriptor
- Member Size
- Member Reference

Name field

If this AssemN/AssemExaN is specified as a VariantExaN Selection Entry for a variant specified for a ConnectionN T→O or O→T Format, then the name displays on the **Module Definition** dialog box, **Assembly** column.

If this AssemN/AssemExaN is specified as a VariantExaN Selection Entry for a variant specified for a ConnectionN T→O or O→T Format, then the name displays on the **Module Properties** dialog box, **General** tab, **Module Definition** group, **Connections** area.

See also

[T->O and O->T Format fields](#) on [page 58](#)

[ConnectionN keyword](#) on [page 53](#)

[Module Definition dialog box](#) on [page 30](#)

[General tab](#) on [page 27](#)

Path field

Ignored.

Size field

The Size field defines the Assembly size. The Size field is empty if the Member Size/Member Reference is provided for all members.

Rockwell Automation recommends leaving this field blank when the Member Size/Member Reference is provided.

See also

[Member Size field](#) on [page 67](#)

[Member Reference field](#) on [page 67](#)

Descriptor field

The Descriptor field consists of the Allow Value Edit bit, Assem Type bit, Not Addressable bit, Write Allowed bit, and Indirect Parameter Handling bit fields.

- Allow Value Edit bit field
 - If this AssemN/AssemExaN is specified as the ConnectionN T→O or O→T Format and all AssemN/AssemExaN Member Reference fields are empty, then the Module-Defined data type created has a single data member of type SINT, INT, DINT, or REAL array as chosen by the user in the **Module Definition** dialog box.
 - If this AssemN/AssemExaN is specified as the ConnectionN Target Config Format and all AssemN/AssemExaN Member Reference fields are empty, then the Module-Defined data type created has a single data member of type SINT if the Allow Value Edit bit is set; otherwise, the data member is hidden.
- Assem Type bit field

Not supported.
- Not Addressable bit field

Not supported.
- Write Allowed bit field

If the AssemN/AssemExaN is specified as the ConnectionN Target Config Format and the Write Allowed value is either 0 (write with I/O connection may be allowed) or 1 (write with I/O connection is allowed) and the ConnectionN specifies a NULL Connection Type for both O→T and T→O, then:

 - A NULL forward open is sent to the module when a MSG instruction specifies the Reconfigure option, with this module as the target.
- Indirect Parameter Handling bit field

- If this AssemN/AssemExaN is specified as the ConnectionN T→O or O→T Format and indirect parameters exist in this AssemN/AssemExaN, set this value set to 0 (treat indirect parameters as indirect parameter).
- If this AssemN/AssemExaN is specified as the ConnectionN Target Config Format and indirect parameters exist in this AssemN/AssemExaN, set this value to 1 (treat indirect parameters as direct parameter).

See also

[Target Config Format field](#) on [page 59](#)

[T->O and O->T Format fields](#) on [page 58](#)

[ConnectionN keyword](#) on [page 53](#)

[Member Reference field](#) on [page 67](#)

Member Size field

No EDS AOP specific handling.

Member Reference field

If this AssemN/AssemExaN is specified as the ConnectionN Target Config Format, then this member is included in the configuration Module-Defined data type.

If this AssemN/AssemExaN is specified as the ConnectionN T→O or O→T Format, then the size of this member is included in the Input/ Output Module-Defined data type array size.

See also

[T->O and O->T Format fields](#) on [page 58](#)

[Target Config Format field](#) on [page 59](#)

[ConnectionN keyword](#) on [page 53](#)

VariantN/VariantExaN keyword

The Variant/VariantExaN keywords are supported as defined in the CIP Common specification, see *THE CIP NETWORKS LIBRARY Volume 1, Common Industrial Protocol (CIP™), Edition 3.11*.

The supported uses of the Variant/VariantExaN keywords are listed in the description of the referencing keyword. See [Connection Manager] section, ConnectionN keyword and Target Config Format field descriptions, and [Assembly section], AssemN/AssemExaN keyword, Name field description.

See also

[ConnectionN keyword](#) on [page 53](#)

[Target Config Format field](#) on [page 59](#)

[Name field](#) on [page 65](#)

[Params] section

The [Params] section includes support for these keywords:

- ParamN keyword
- EnumN keywords

ParamN keyword

The ParamN keyword contains additional information about attributes in the device. Use this information for Input/Output/Configuration details.

The EnumN keyword contains additional information about enumerations for ParamN entries.

Input/Output/Configuration details uses

Input/Output/Configuration details uses includes:

- ConnectionN O→T Format.
- ConnectionN T→O Format.
- ConnectionN Target Config Format.

See the *ConnectionN keyword* for more information.

The ParamN name should use Module-Defined data type member name naming conventions. See *Parameter Name field* for more information.

See also

[ConnectionN keyword](#) on [page 53](#)

[T->O and O->T Format fields](#) on [page 58](#)

[Target Config Format field](#) on [page 59](#)

[Parameter Name field](#) on [page 71](#)

ParamN fields

The ParamN keyword includes these fields:

- Link Path
- Descriptor
- Data Type
- Data Size
- Parameter Name
- Units String
- Help String
- Minimum Value
- Maximum Value
- Default Value
- Link (Multiplier, Divider, Base, and Offset)
- Scaling (Multiplier, Divider, Base, and Offset)
- Decimal Precision
- International Parameter Name
- International Engineering Units

Link Path field

- International Help String

Ignored.

Descriptor field

The Descriptor field:

- Supports settable path
Ignored.
- Get_Enumerated_String Supported

Ignored.



Tip: Enumerations are supported as described in other parts of this document without this bit being set.

- Supports Scaling
Ignored.
- Supports Scaling Links
Ignored.

- Read Only Parameter

If the parameter described by the ParamN entry is included in the module configuration, it is hidden in the Configuration module-defined data type and will not be included in an output Module-Defined tag. It will also not appear in the output data. The output reverts to an array of SINT.

- Monitor Parameter

Ignored.

- Supports Extended Precision

Ignored.

- Supports non-consecutive enumerated strings
- Ignored.
- Allows both enumeration and min/max range values

Ignored.

- Non-displayed parameter

If set, ParamN is not displayed.

- Indirect Parameter reference

If set and the ParamN is in a Configuration assembly, it shows in the **Custom Data Type** dialog box, **Module Definition**. If the ParamN is in an Input or Output assembly, the selected ParamN in the **Custom Data Type** dialog box, **Module Definition**, is used as the Module-Defined data type member.

- Not Addressable

Ignored.

- Save Supported

Ignored.

- Apply Supported
Ignored.
- Write Only Parameter
Ignored.

See also

[Custom Data Type dialog box](#) on [page 32](#)

Data Type field

STRING2, STRINGN, EPATH, and STRINGI types are not supported. The ParamN entry is discarded and any keyword that specifies the ParamN entry as a field value is discarded.

If ParamN is in a Configuration assembly, the data type determines the Module- Defined data type member type:

- CIP BOOL type results in controller BOOL type.
- CIP SINT type results in controller SINT type.
- CIP INT type results in controller INT type.
- CIP DINT type results in controller DINT type.
- CIP LINT type results in controller LINT type.
- CIP USINT type results in controller USINT type.

If the maximum value of the unsigned 8-bit integer is 128 or larger, the entire Module-Defined data type will be a SINT array.

- CIP UINT type results in controller UINT type.

If the maximum value of the unsigned 16-bit integer is 32768 or larger, the entire Module-Defined data type will be a SINT array.

- CIP UDINT type results in controller UDINT type.

If the maximum value of the unsigned 32-bit integer is 2147483648 or larger, the entire Module-Defined data type will be a SINT array.

- CIP ULINT type results in controller ULINT type.

If the maximum unsigned ULINT value would result in a negative LINT value, the entire Module-Defined data type becomes a SINT array.

- CIP REAL type results in controller REAL type.
- CIP LREAL type results in controller LREAL type.
- CIP DATE type results in controller UINT type.
- CIP TIME_OF_DAY type results in controller UDINT type.
- CIP STRING type results in array of controller USINT type.
- CIP BYTE type results in controller USINT type.
- CIP WORD type results in controller UINT type.
- CIP DWORD type results in controller UDINT type.
- CIP LWORD type results in controller ULINT types.
- CIP FTIME type results in controller TIME32 type.

- CIP ITIME type results in controller INT type.
- CIP SHORT_STRING type results in array of controller USINT type.
- CIP NTIME type results in controller LTIME type
- CIP LTIME type results in controller TIME type
- CIPT TIME type results in controller DINT type
- CIP UTIME type results in controller DT type
- CIP STIME type results in controller LDT type
- CIP DATE_AND_TIME type results in controller UINT (DATE) and UDINT (TIME)
- CIP ENGUNIT type results in controller UINT type

Data Size field

The Data Size field follows the CIP specification. See *THE CIP NETWORKS LIBRARY Volume 1, Common Industrial Protocol (CIP™)*.

Parameter Name field

If the ParamN is a member of a Configuration assembly, the Parameter Name is used as the Module-Defined data type member name. The Parameter Name is converted to conform to the IEC 61131 name standard of alpha-numeric characters or underscore (_) characters. All unsupported characters, including spaces, are replaced with an underscore (_) character during the conversion, consecutive underscore (_) characters and trailing underscore (_) characters are removed and leading numbers are prefixed with an underscore character. Do not use underscore and space characters in Parameter Names.

Follow these guidelines for Parameter Names that are included in Configuration assemblies:

- Use short Member names when practical. Users do not want to type ConfigDataBits when Data suffices. The user can infer much from the context. Data appearing in an input data type is obviously input data. Data from a discrete module is almost always represented as a bit. The goal for Module-Defined data type member names is 15 characters or less.
- Capitalize the first letter in each word making up the parameter name. This allows the words to stand out when the data type name is compressed after spaces are removed.
- Use consistent naming. Refer to existing data types as a guide.
- Do not use redundant naming.
- Do not use the data type in the parameter name.
- Naming of BOOLEAN members should be for the positive sense of the member. Note that this requires the proper design of the device. Also note that the on state is not necessarily the most interesting state.
- Prefix (not post fixed) channel data with Ch followed by the channel number. It is not spelled out since it has become a common convention.
- In cases when using individual homogeneous bits named (versus indexed through a Data array), prefixed (not post fixed) the individual bits with Pt followed by the 2-digit point number.

Units String field

Ignored.

Help String field

Ignored.

Minimum Value field

Ignored if the parameter is part of a Configuration assembly. If used in the **Module Definition** dialog box, this field represents the minimum value that the user can choose.

See also

[Module Definition dialog box](#) on [page 30](#)

Maximum Value field

If in a Configuration assembly, used to determine if a parameter can be properly represented. If used in the **Module Definition** dialog box, this field represents the maximum value that can be chosen.

See also

[Module Definition dialog box](#) on [page 30](#)

Default Value field

Used as defined in the CIP specification. See *THE CIP NETWORKS LIBRARY Volume 1, Common Industrial Protocol (CIP™), Edition 3.11*.

Link (Multiplier, Divider, Base and Offset) fields

Ignored.

Scaling (Multiplier, Divider, Base and Offset) fields

Ignored.

Decimal Precision field

Ignored.

International Parameter Name field

Ignored.

International Engineering Units field

Ignored.

International Help String field

Ignored.

ParamN enumeration, EnumN

If the EnumN is for a bit string (BYTE, WORD, DWORD, DWORD, LWORD) ParamN that is a member of a configuration assembly, then the bit name is used as the Module-Defined data type member name. The bit name is converted to conform to the IEC 61131 name standard. All unsupported characters, including spaces, are replaced with an underscore (_) character during the conversion, consecutive underscore (_) characters and trailing underscore (_) characters are removed and leading numbers are prefixed with an underscore (_) character.

Follow these guidelines for bit names that are included in Configuration assemblies:

- Use short Bit names when practical.
- Capitalize the first letter in each word making up the bit name. This allows the words to stand out when the data type name is compressed after spaces are removed.
- Use consistent naming (for example, Data and Status). If two different modules return status, name both Status. Refer to existing data types as a guide as opposed to getting creative.
- Do not use redundant naming. For example, O.On instead of O.OutputOn.
- Do not use the data type in the name. For example, On, not OnBit.
- Naming of bits should be for the positive sense of the member. FuseBlown is 1 when the fuse is blown, 0 when it is not. It is generally preferable to name a bit member in a manner that reflects the interesting state rather than the steady state value. In most cases, this is the on, enabled, or energized state versus the off state. For example, FuseBlown is preferable to FuseOK or FuseNotBlown. Note that this requires the proper design of the device. Note also that the on state is not necessarily the most interesting state.

[Groups] section

Ignored.

[Internationalization] section

The language the Logix Designer application displays determines which string from the EDS file is used for the items below. If a string for the Logix Designer application language is not specified, then the string from the non-Internationalization section version of the keyword is used.



Tip: The keywords listed below should have English, French, German, Spanish, Portuguese, Italian, Korean, Japanese, and Chinese strings because Logix Designer application is translated to each of these languages.

See also

[ProdName keyword](#) on [page 74](#)

[AssemN/AssemExaN keyword](#) on [page 65](#)

[ConnectionN keyword](#) on [page 74](#)

ProdName keyword

If this ProdName is defined, it is used instead of the ProdName specified in the [Device] section.

See also

[ProdName keyword](#) on [page 50](#)

ParamN keyword

Ignored

International Parameter Name field

Ignored

International Engineering Units field

Ignored

International Help String field

Ignored

EnumN keyword

Ignored.

GroupN keyword

Ignored

AssemN/AssemExaN keyword

Used only for assemblies selected by I/O variants. If the Internationalized Assembly Name String is defined, it is used instead of the name defined in the [Assembly] section in all cases.

ConnectionN keyword

If the Internationalized Connection Name String is defined, it is used instead of the name defined in the [Connection Manager] section in all cases.

The Internationalized Connection Help String is not used.

[Modular] section

Ignored.

[Ethernet Link Class] section

According to the ODVA specifications, devices must implement one instance of the Ethernet Link object per external EtherNet/IP port. It is recommended that the EDS file should also contain the [Ethernet Link Class] section.

IMPORTANT If the module has more than one Ethernet port, this is reflected in the EDS as specified in the ODVA specifications.

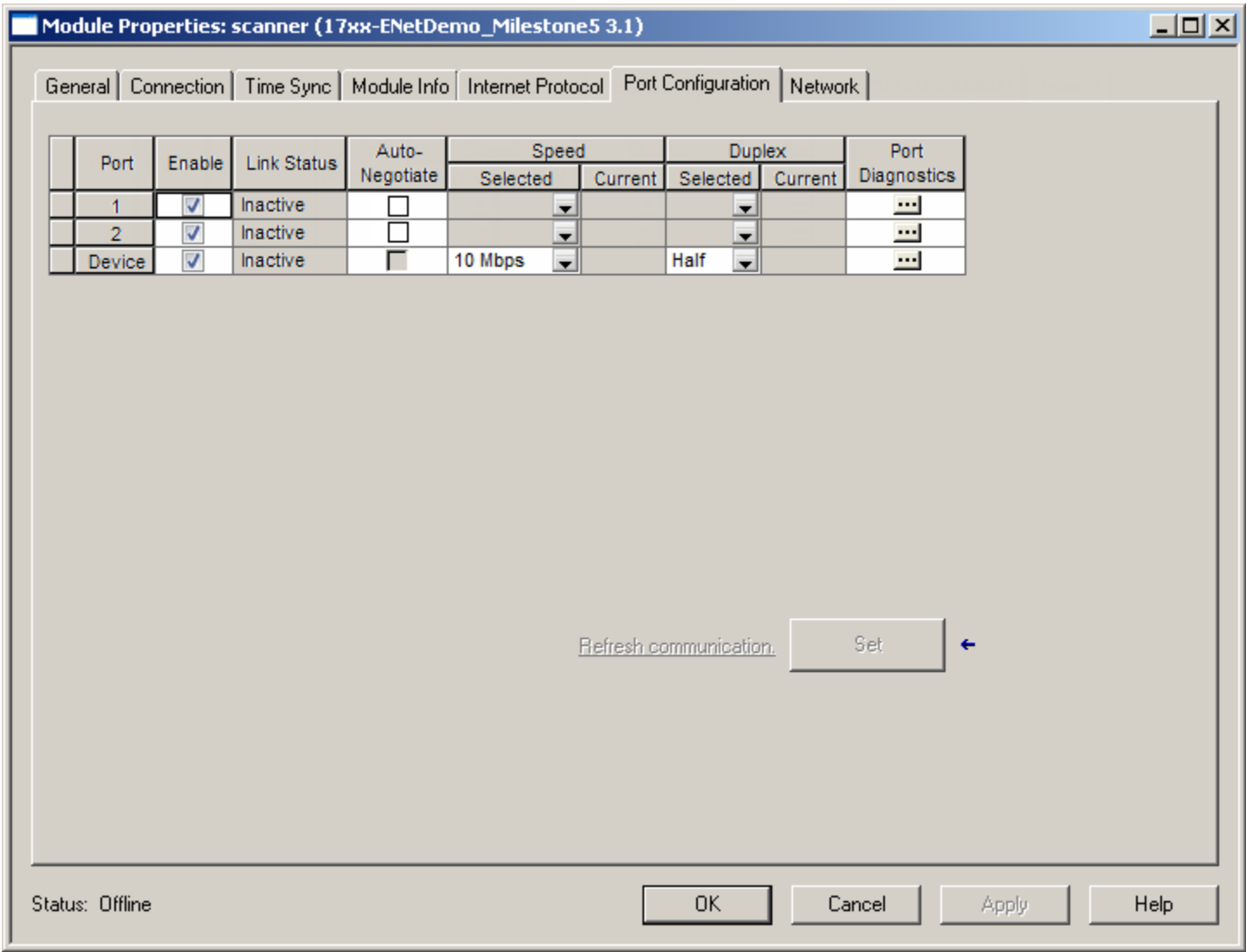
Define these keywords:

- Number_Of_Static_Instances
- MaxInst
- Either
 - InterfaceLabelN
 - In the [Params] section, create one or more ParamN entries with Path value of 20 F6 24 NN 30 0A where NN is the Ethernet Link object instance. The default value of these ParamN entries should be the interface label string.



Tip: If no interface label is specified, a default label is displayed, the instance value.

The **Port Configuration** tab display is affected by these items. It is also recommended that the device implement the Ethernet Link object Interface Counters and Media Counters attributes.



See also

[Port Configuration tab](#) on [page 40](#)

[TCP/IP Interface Class] section

The TCP/IP Interface object is a required object for EtherNet/IP devices. As such, the EDS file should include the [TCP/IP Interface Class] section.

Define these keywords:

- In the [Params] section, create a ParamN entry with a Path value of 20 F5 24 01 30 02. The default value of this ParamN should represent the TCP/IP object capabilities for this module. This is necessary for the **Internet Protocol** tab to properly reflect the device's capabilities.



Tip: If no ParamN entry exists, the value of 0 is used for the TCP/ IP capabilities attribute.

The **Internet Protocol** tab is affected by this item.

Module Properties: scanner (Test Device #9L 1.1)

General | Connection | Module Info | **Internet Protocol** | Port Configuration

Internet Protocol (IP) Settings
 IP settings can be manually configured or can be automatically configured if the network supports this capability.

☒ Manually configure IP settings
☐ Obtain IP settings automatically using BOOTP
☐ Obtain IP settings automatically using DHCP

IP Settings Configuration

Physical Module IP Address: 0 . 0 . 0 . 0 Subnet Mask: 0 . 0 . 0 . 0

IP Address in physical module does not match address in general properties (10.88.46.58)
[Copy IP address from general properties.](#) Gateway Address: 0 . 0 . 0 . 0

Domain Name: Primary DNS Server Address: 0 . 0 . 0 . 0

Host Name: Secondary DNS Server Address: 0 . 0 . 0 . 0

[Refresh communication.](#) ←

Status: Running

See also

[Internet Protocol tab](#) on [page 39](#)

[ParamN keyword](#) on [page 68](#)

[DLR Class] section

If the device is capable of being a ring supervisor, this keyword assignment enables ring supervisor functionality on the **Network** tab.

Ring_Supervisor_Capable = Yes

If the device cannot be a ring supervisor, use the value **No** for this keyword.

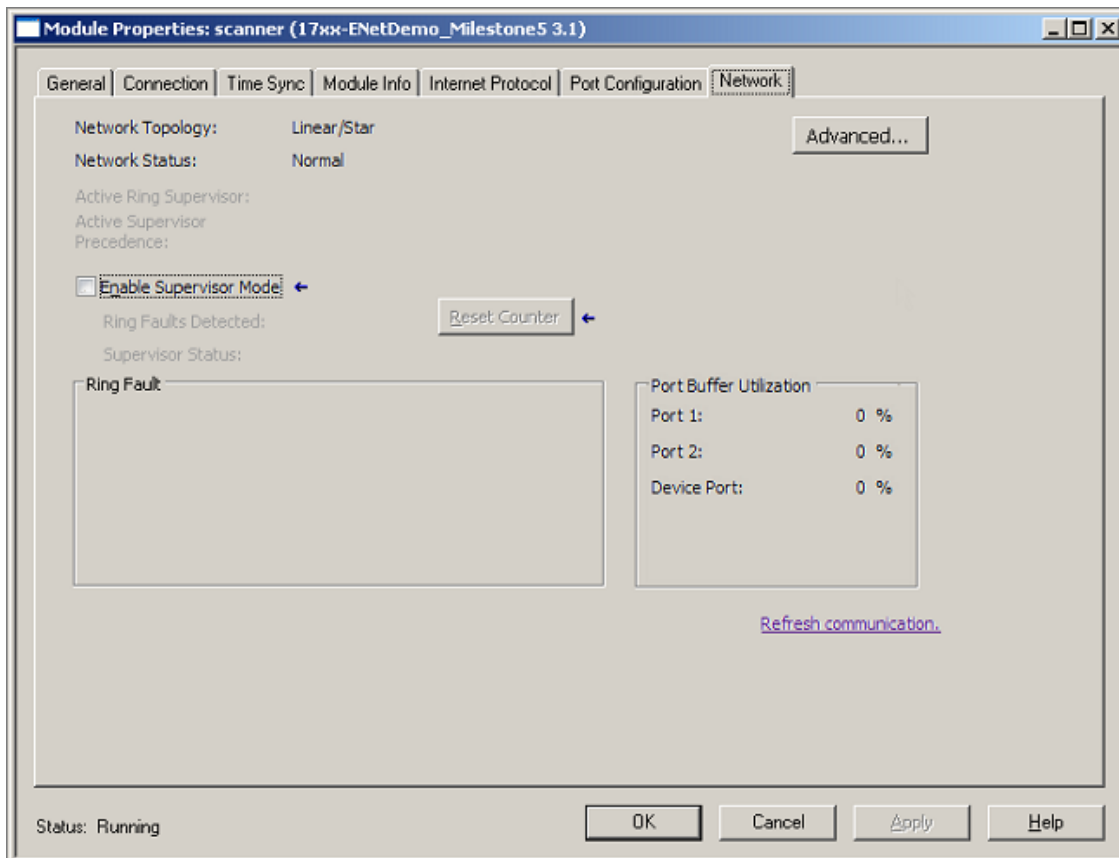


Tip: If the keyword is omitted, the device is assumed to not have supervisor capability.

If the device is capable of being a ring supervisor, also define these:

- In the [Assembly] section, create an AssemN entry with Path value of 20 47 24 01 30 04.
- The AssemN entry should include the Member References for each member of the Ring Supervisor Config attribute. The ParamN that represents the Beacon Interval member should specify the minimum and maximum values supported for this module's Beacon Interval.
- If no Ring Supervisor Config AssemN entry exists, 400 microseconds is used as the minimum Beacon Interval and 100,000 microseconds is used as the maximum Beacon Interval.

The **Network** tab displays if this section exists.



See also

[AssemN/AssemExaN keyword](#) on [page 65](#)

[Member Reference field](#) on [page 67](#)

[Network tab](#) on [page 42](#)

[Time Sync Class] section

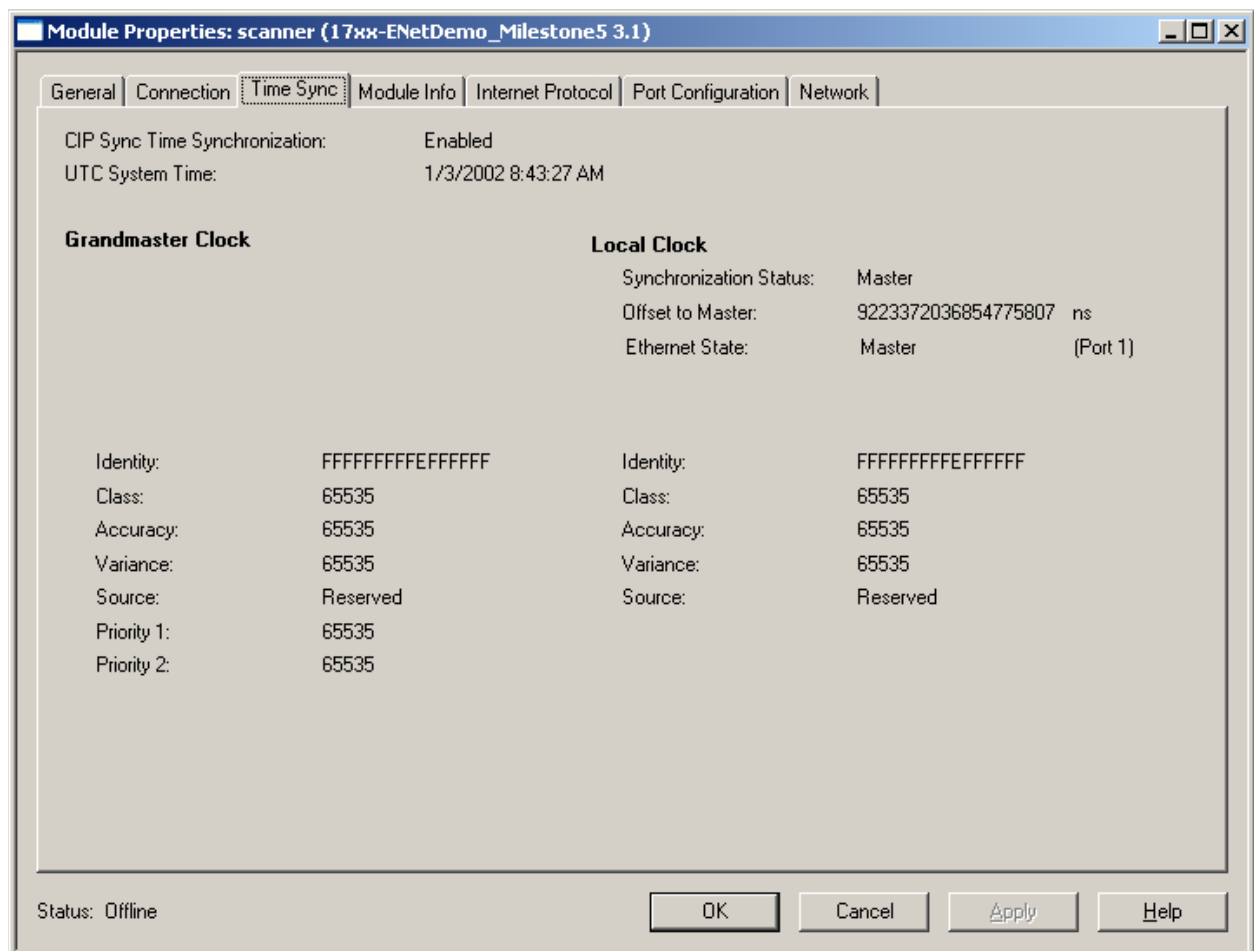
If the module supports the Time Sync object, this section should be included in the EDS file. The information found in this dialog box comes from the required attributes of the object class.

Define the keyword:

- In the [Params] section, create a ParamN entry with a Path value of 20 43 24 01 30 0B. This ParamN is used to initialize the **Time Sync** dialog box with the number of time sync ports. If this does not exist, then the value of 1 is used. This is necessary for the **Time Sync** dialog box to properly reflect the device's current time sync information.

If the product has a Rockwell Automation vendor ID, a check is made to see if vendor specific service 0x35 of the Time Sync object is supported, or if the 1_Time_Sync_Request_Clock_Description_Extension keyword is defined in the Time Sync section. This is used to initialize the **Time Sync** dialog box, indicating the ReqClockDescription service is supported.

The **Time Sync** tab displays if the [Time Sync Class] section exists.



See also

[Time Sync tab](#) on [page 44](#)

Description of data types and tags

This section describes the rules the EDS AOP follows for creating Module-Defined Configuration, Input, and Output data type names, tag names, and Module-Defined data type member names.

See also

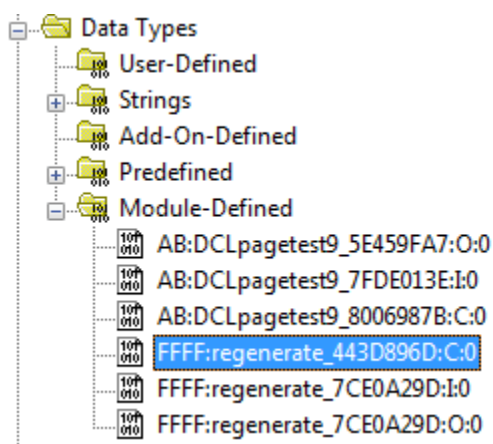
[Module-Defined Data Type Naming](#) on [page 79](#)

[Tag Naming](#) on [page 80](#)

[Module-Defined Data Type Member Naming](#) on [page 81](#)

Module-Defined data type naming

Module-defined data types define the structure of the data used by the module for its Input, Output, and Configuration Data. These tags provide access to this data via the controller's program. This section describes the rules Logix Designer application uses for naming module-defined data types. The names display in the **Controller Organizer**, under **Data Types>Module-Defined**.



A C/I/O Module Defined data type name is constructed from:
 <Vendor>:<Catalog_Number>_<CRC>:<Tag_Type>:<Version>

- Vendor code: the 4-digit hex value assigned by ODVA.
- Catalog_Number string value:

The Catalog_Number String comes from the Catalog string in the EDS file. Dash (-) characters are converted to underscore () characters and all non-IEC-61131 characters are removed. If no Catalog string exists in the EDS file, the 4 digit hex ProdType and 4 digit hex ProdCode are used, separated by an underscore () character (for example:

0002_0006). The Catalog_Number string is truncated to limit the Module-Defined data type name length to 40 characters.

- CRC string value: Generated from the Module-Defined data type contents, the details of this algorithm are not provided.
- Tag_Type string value:
 - C
 - I
 - O
 - SI
 - SO
- Version string value: Always 0.

See also

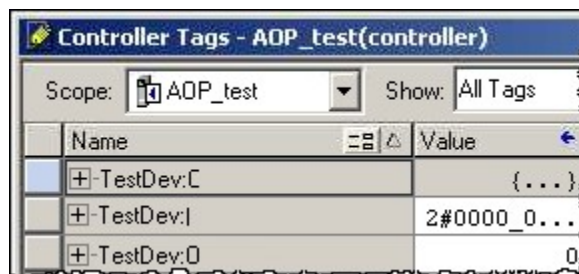
[Tag Naming](#) on [page 80](#)

[Module-Defined Data Type Member Naming](#) on [page 81](#)

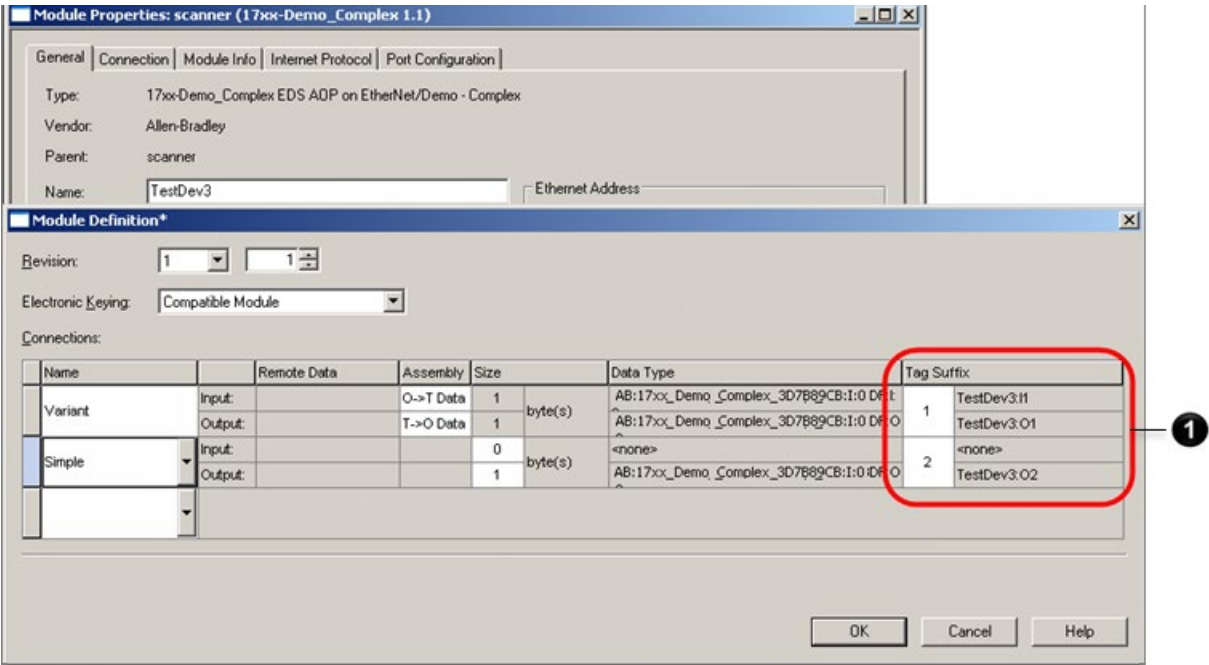
Tag naming

Tag naming follows these principles:

- Use the Name on the **Module Properties** dialog box, **General** tab:
 - <Name>:<I, O, SI, or SO>[<Ordinal>]
 - <Name>:C



- The ordinal name comes from the **Module Definition** dialog box, **Tag Suffix** column. Ordinal only exists when the user can configure more than one connection.



Number	Description
1	The ordinal name comes from the Module Definition dialog box, Tag Suffix column.

See also

[General tab](#) on [page 27](#)
[Module Definition dialog box](#) on [page 30](#)

Module-Defined data type member naming

Member naming for module-defined data types follow these principles:

- Member names are truncated to 40 characters.
- Duplicate member names are made unique by appending an ordinal suffix.

Example EDS files

These sections describe examples of EDS files and how these EDS files translate into details in Logix Designer application.



Tip: The example EDS files are attached to this PDF file. In Adobe Acrobat, or Adobe Reader, you display the Attachments panel by:

- Selecting the small paperclip icon in the bottom left corner of the Acrobat or Reader window.
- Selecting the **View > Navigation Panels > Attachments** command to display the Attachments panel.

See *Access the attachments* for more details.

Example EDS files are discussed in these section:

- [Minimum EDS](#) on [page 83](#)
- [Enhanced EDS, Single Connection](#) on [page 84](#)
- [Enhanced EDS, Single Connection, Configuration Data](#) on [page 85](#)
- [Enhanced EDS, multiple choice of connections and multiple simultaneous connections](#) on [page 87](#)
- [Enhanced EDS, safety and standard connections](#) on [page 89](#)

See also

[Access the attachments](#) on [page 10](#)

Minimum EDS

This section describes the minimum that is required to make an EDS usable in a Logix Designer application environment, plus a number of extensions that improve its usability.

The Minimal EDS file described in this section makes use of the Allow Value Edit bit while the more complex EDS files described later use parameters to structure the assemblies. Rockwell Automation strongly recommends adding more details to an EDS as outlined in the more detailed EDS files shown later. The Minimal EDS file is shown here, mainly for reference and to break the evolution of an EDS into several steps.

With the Minimal EDS, the Logix Designer application EDS AOP offers only these choices and structures.

- Choice of IP address.
- Selection of Unicast and Multicast data production where appropriate and offered by the EDS.
- Selection of Cyclic or Change-of-State trigger if offered by the EDS.
- Creation of an Input tag (an Exclusive-Owner Connection, an Input-Only Connection, or a Listen-Only Connection) and possibly an

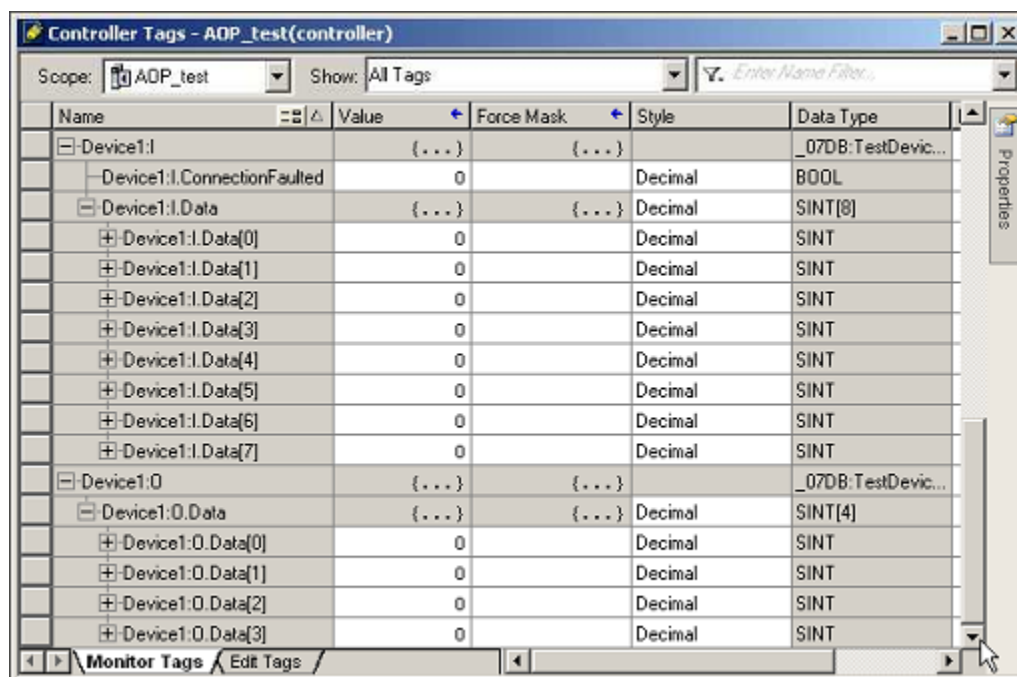
Output tag (only for an Exclusive-Owner Connection), depending on what is offered in the EDS.

- Choice of SINT, INT, DINT, or REAL data structure for the I/O tags.

This sample EDS actually contains only an entry for an Exclusive-Owner Connection. See the attached *Minimal EDS.eds* file for details.

The Minimal EDS file creates only an Input data tag and an Output data tag of the defined sizes. No structure or meaning of the I/O data is visible. Logix Designer application adds a status bit to the Input tag that reports when the connection fails.

The resulting tags appear with data type SINT:



See also

[Access the attachments](#) on [page 10](#)

[Example EDS Files](#) on [page 83](#)

[Enhanced EDS, single connection](#) on [page 84](#)

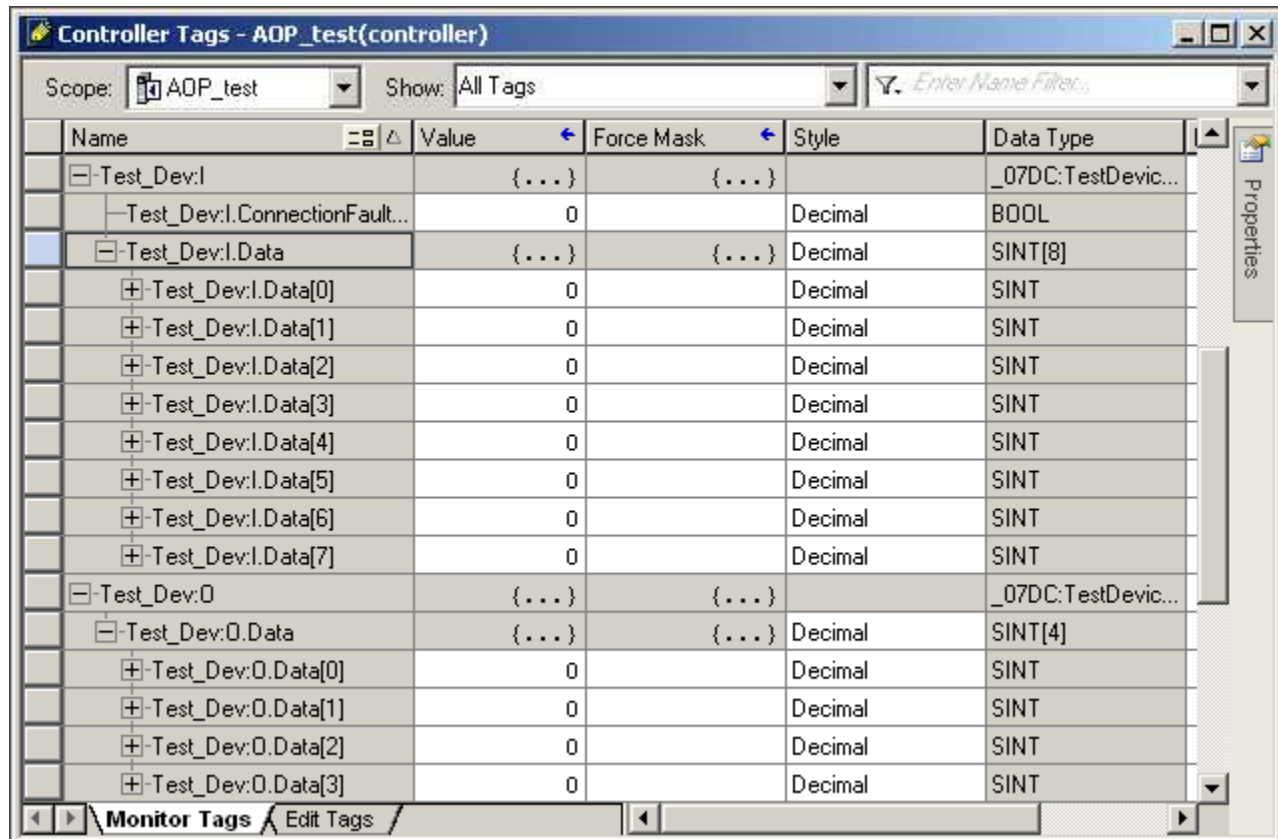
Enhanced EDS, single connection

For a better user experience, add a range limitation for Requested Packet Interval (RPI).

See the attached *One Connection, IO Assemblies with Structures EDS.eds* file for details. The EDS file includes six parameters and two assemblies.

- Param1 limits the RPI range to values between 4.0 and 100.0ms with a default value of 10.0ms.

This module appears in Logix Designer application:



See also

[Access the attachments](#) on [page 10](#)

[Minimum EDS](#) on [page 83](#)

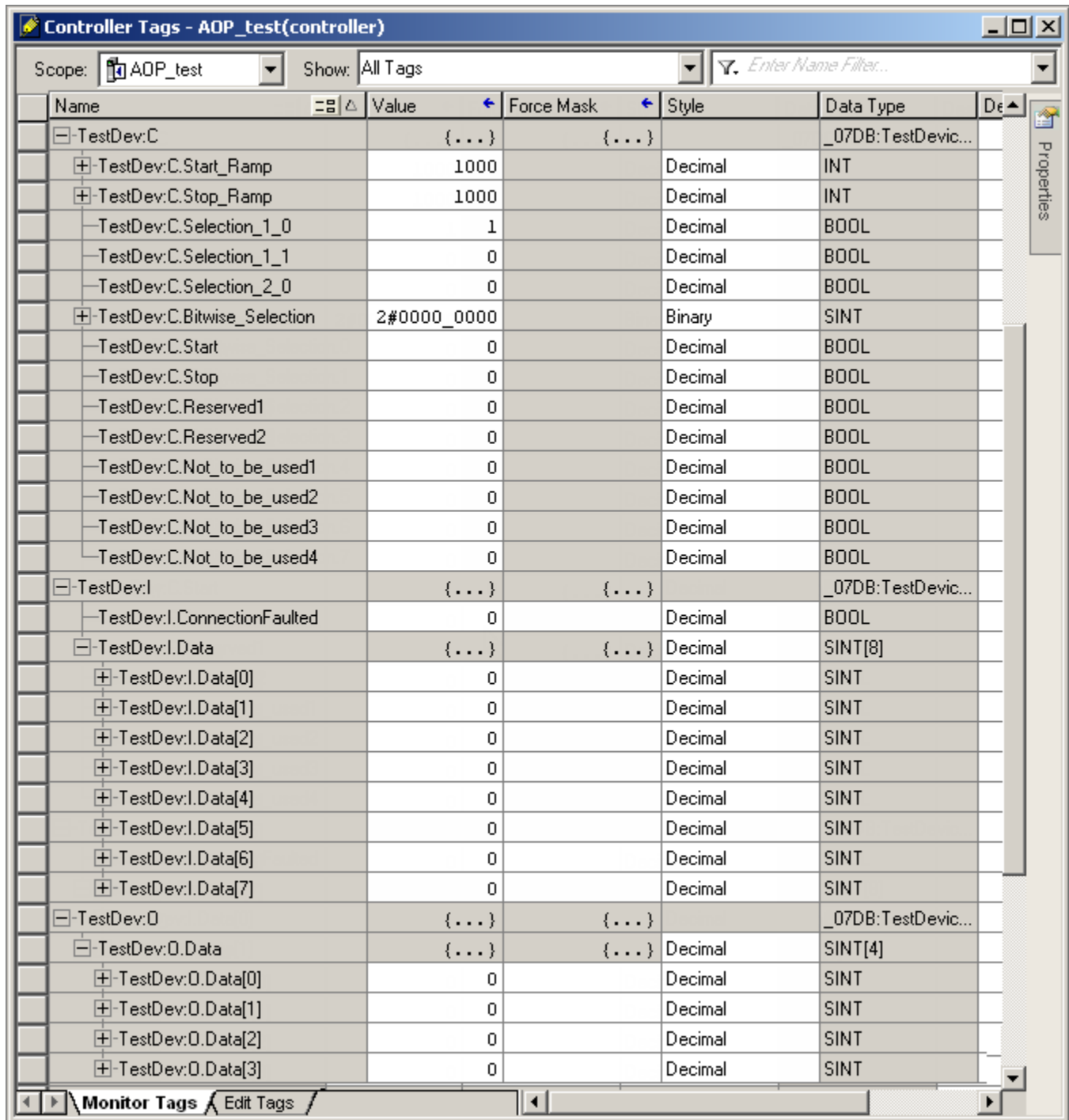
[Enhanced EDS, single connection, configuration data](#) on [page 85](#)

Enhanced EDS, single connection, configuration data

The *One Connection, 3 Assemblies with Structures EDS.edc* file includes a configuration assembly structured into individual parameters, in addition to the features of the *One Connection, IO Assemblies with Structures EDS.edc* file.

There are now a total of 11 parameters in the EDS file. The additional parameters, 7 through 11, describe the structure of the Configuration Assembly.

This module appears in the Logix Designer application tag structure:



See also

[Access the attachments](#) on [page 10](#)

[Enhanced EDS, single connection](#) on [page 84](#)

[Enhanced EDS, multiple choice of connections and multiple simultaneous connections](#) on [page 87](#)

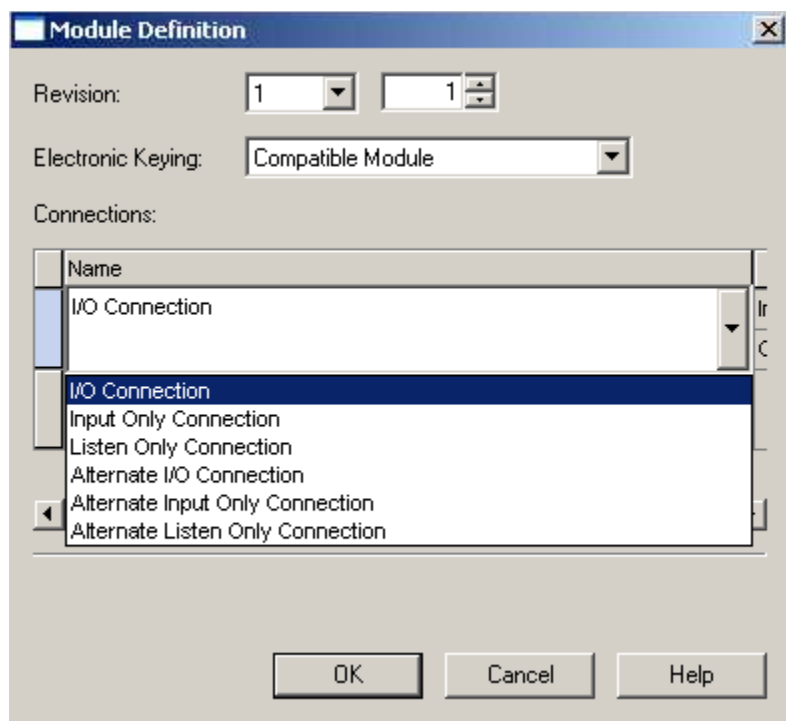
Enhanced EDS, multiple choice of connections and multiple simultaneous connections

The attached *Multiple Connections 3 Assemblies with Structures EDS.eds* file defines multiple ConnectionN entries for a module. There are typically a set of up to three ConnectionN entries describing one or more connections:

- An Exclusive-Owner Connection (if the module has Output data).
- An Input-Only Connection.
- A Listen-Only Connection.

Such a set of ConnectionN entries specify the same Input Connection Point and (if included) the same configuration data. Multiple sets like this may exist that describe connections using different I/O Assemblies.

With the *Multiple Connections 3 Assemblies with Structures EDS.eds* file, select one of the connections in the **Module Definition** dialog box. The first Exclusive-Owner Connection in the EDS displays as the default connection.



If Logix Designer application determines that there are multiple sets of ConnectionN entries, a group of connections is created.

The **Module Definition** dialog box shows the following configuration:

- Revision:** 1 (dropdown) and 1 (spin box)
- Electronic Keying:** Compatible Module (dropdown)
- Connections:**

Name	Input:	Output:	Tag Suffix
I/O Connection	1	TestDev2:I1	TestDev2:O1
Select a connection			

Buttons at the bottom: OK, Cancel, Help.

Once a member of a group of connections has been chosen, further connections are only selected from ConnectionN entries outside that group.

The **Module Definition*** dialog box shows the following configuration:

- Revision:** 1 (dropdown) and 1 (spin box)
- Electronic Keying:** Compatible Module (dropdown)
- Connections:**

Name	Input:	Output:	Tag Suffix
I/O Connection	1	TestDev2:I1	TestDev2:O1
Alternate Input Only Connection	2	TestDev2:I2	<none>

Buttons at the bottom: OK, Cancel, Help.

The tags created by multiple connections have numbers appended to make them unambiguous.

See also

[Access the attachments](#) on [page 10](#)

[Module Definition dialog box](#) on [page 30](#)

Enhanced EDS, safety and standard connections

An EDS file may define both safety and standard connections. See the attached *Std+SafetyInOutExample.eds* file for details. This example also shows the use of ParamN entries that are used by the **Connection Properties** and **Internet Protocol** tabs. See Params500 through Param510 in the EDS file.

With this EDS, the **Module Definition** dialog box lets users enable the safety input connection, safety output connection, standard connection, or a combination of these connections.

Revision: 1 001

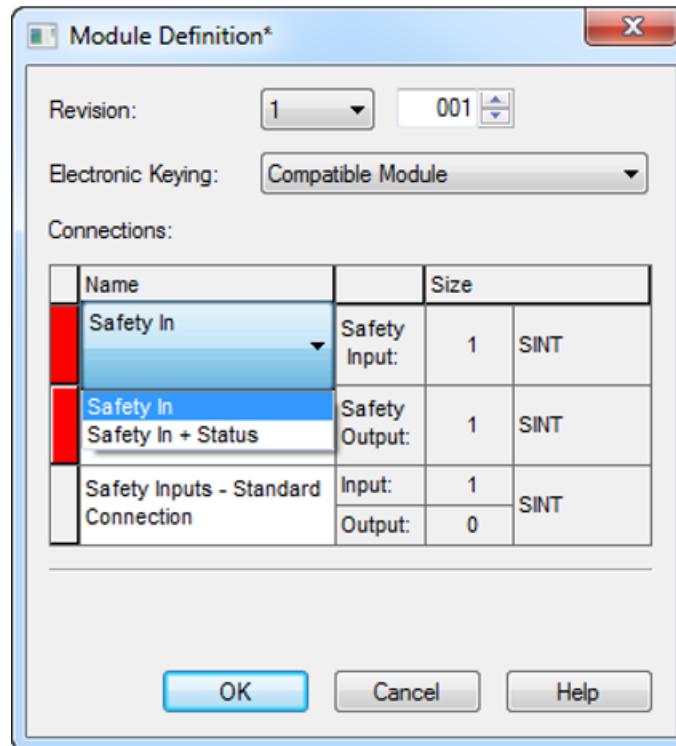
Electronic Keying: Compatible Module

Connections:

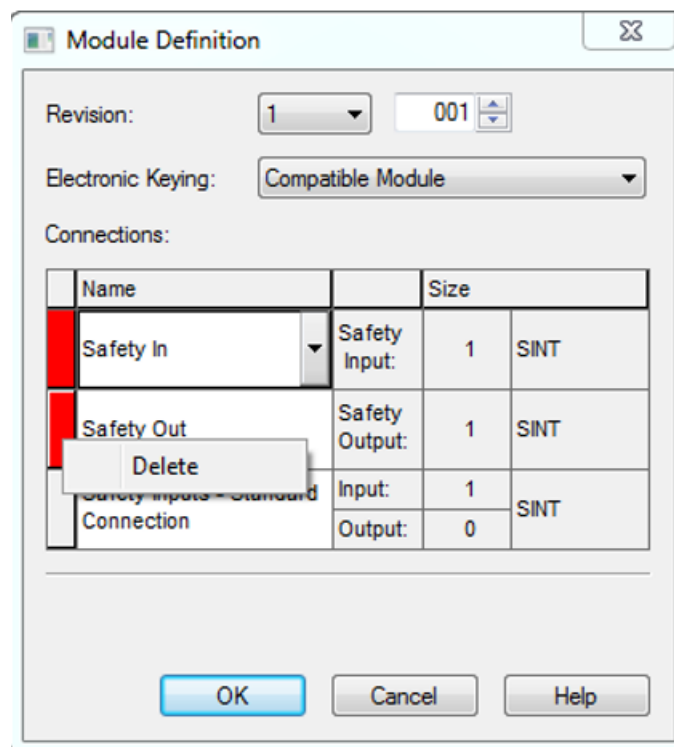
Name		Size
Safety In	Safety Input:	1 SINT
Safety Out	Safety Output:	1 SINT
Safety Inputs - Standard Connection	Input:	1 SINT
	Output:	0

OK Cancel Help

To select a connection, select the drop-down arrow next to the connection name.



To remove a connection, select the row's left bar and choose **Delete**.



In the Logix Designer application, the standard connection is designated with the I tag type. The safety connection tags are designated with the SI and SO tag types.

Controller Tags - V32_EDS_AOP_Safety(controller) X						
Scope: V32_EDS_AOP_		Show: All Tags				
Name	Value	Force Mask	Style	Data Type	Class	
Test_Development	{...}	{...}		FFDC:TestSafetyDevice13_D51EFC07:10	Standard	
Test_Development.ConnectionFaulted	0		Decimal	BOOL	Standard	
Test_Development.Data	{...}	{...}	Decimal	SINT[1]	Standard	
Test_Development.Data[0]	0		Decimal	SINT	Standard	
Test_Development.Data[0].0	0		Decimal	BOOL	Standard	
Test_Development.Data[0].1	0		Decimal	BOOL	Standard	
Test_Development.Data[0].2	0		Decimal	BOOL	Standard	
Test_Development.Data[0].3	0		Decimal	BOOL	Standard	
Test_Development.Data[0].4	0		Decimal	BOOL	Standard	
Test_Development.Data[0].5	0		Decimal	BOOL	Standard	
Test_Development.Data[0].6	0		Decimal	BOOL	Standard	
Test_Development.Data[0].7	0		Decimal	BOOL	Standard	
Test_Development.SI	{...}	{...}		FFDC:TestSafetyDevice13_D51EFC07:SI0	Safety	
Test_Development.SI.ConnectionFaulted	0		Decimal	BOOL	Safety	
Test_Development.SI.Data	{...}	{...}	Decimal	SINT[1]	Safety	
Test_Development.SI.Data[0]	0		Decimal	SINT	Safety	
Test_Development.SI.Data[0].0	0		Decimal	BOOL	Safety	
Test_Development.SI.Data[0].1	0		Decimal	BOOL	Safety	
Test_Development.SI.Data[0].2	0		Decimal	BOOL	Safety	
Test_Development.SI.Data[0].3	0		Decimal	BOOL	Safety	
Test_Development.SI.Data[0].4	0		Decimal	BOOL	Safety	
Test_Development.SI.Data[0].5	0		Decimal	BOOL	Safety	
Test_Development.SI.Data[0].6	0		Decimal	BOOL	Safety	
Test_Development.SI.Data[0].7	0		Decimal	BOOL	Safety	
Test_Development.SO	{...}	{...}		FFDC:TestSafetyDevice13_FA74D033:SO0	Safety	
Test_Development.SO.Data	{...}	{...}	Decimal	SINT[1]	Safety	
Test_Development.SO.Data[0]	0		Decimal	SINT	Safety	
Test_Development.SO.Data[0].0	0		Decimal	BOOL	Safety	
Test_Development.SO.Data[0].1	0		Decimal	BOOL	Safety	
Test_Development.SO.Data[0].2	0		Decimal	BOOL	Safety	
Test_Development.SO.Data[0].3	0		Decimal	BOOL	Safety	
Test_Development.SO.Data[0].4	0		Decimal	BOOL	Safety	
Test_Development.SO.Data[0].5	0		Decimal	BOOL	Safety	
Test_Development.SO.Data[0].6	0		Decimal	BOOL	Safety	
Test_Development.SO.Data[0].7	0		Decimal	BOOL	Safety	

See also

[Access the attachments on page 10](#)

Index

[

- [Assembly] section** 62
 - configuration parameters 22
 - input and output parameters 19
- [Connection Manager] section** 51
- [Device Classification] section** 51
- [Device] section** 46
- [Ethernet Link Class] section** 71
- [File] section** 45
- [Groups] section** 70
- [Internationalization] section** 70
- [Modular] section** 71
- [ParamClass] section** 62
- [Params] section** 65
 - configuration parameters 22
 - input and output parameters 19
- [TCP/IP Interface Class] section** 72
- [Time Sync Class] section** 74

A

- AOP**
 - definition 14
 - Generic Device 13
- AssemN/AssemExaN keyword**
 - Descriptor field 63
 - Name field 63
 - Path field 63
 - Size field 63

B

- bit-enumerated parameters** 20, 23
- buses, supported** 13

C

- Catalog keyword** 46
- CfgAssembly keyword** 62
- CfgAssemblyExa keyword** 62
- CIP connections** 19
- ClassN keyword** 51
- Configuration**
 - configuration parameters 22, 23
 - data structure 22
 - tags 22
- conformance testing** 14
- Connection Name String field** 60
- Connection Parameters field** 53

- Connection tab** 33
- ConnectionN keyword** 51, 71
 - Connection Name String field 60
 - Connection Parameters field 53
 - Help String field 61
 - Path field 61
 - Proxy Config Size and Format fields 57
- CreateDate keyword** 46
- CreateTime keyword** 46
- creating EDS files** 19
- Custom Data Type dialog box** 32

D

- data structures**
 - Configuration 22
 - Input 19
 - Output 19
- definitions** 14
- deploying EDS files** 14
- Descriptor field** 63, 66
- Descriptor keyword** 62
- devices**
 - not supported 13
 - supported 13

E

- editing EDS files** 14
- EDS AOP**
 - definition 14
 - feature limitations 13
- EDS file sections**
 - [Assembly] 62
 - Connection Manager 51
 - Device 46
 - Device Classification 51
 - DLR Class 73
 - Ethernet Link Class 71
 - File 45
 - Groups 70
 - Internationalization 70
 - Modular 71
 - ParamClass 62
 - Params 65
 - TCP/IP Interface Class 72
 - Time Sync Class 74
- EDS files**
 - and <L 13
 - creating 19
 - deploying 14
 - editing 14
 - Enhanced EDS, multiple connections 85
 - Enhanced EDS, safety and standard connections 87
 - Enhanced EDS, single connection 82

Enhanced EDS, single connection,
configuration data 83
examples 81
general information 14
icon files 15
internationalization 70
Minimum EDS 81
not available 14
registering 14
requirements for module developers 13

EDS Hardware Installation Tool

registering EDS files 14
warnings 14

EDS, definition 14

Enhanced EDS

multiple connections file example 85
safety and standard connections 87
single connection file example 82
single connection, config data file
example 83

Enhanced EDS, safety and standard connections 87

enumeration

configuration parameter 23
EnumN 69
I/O parameter 20
ParamN 69

EnumN

enumeration 69
keyword 71

example EDS files

Enhanced EDS, multiple connections
85
Enhanced EDS, safety and standard
connections 87
Enhanced EDS, single connection 82
Enhanced EDS, single connection, config
data 83
Minimum EDS 81

F

fields

Data Size 68
Data Type 67
Decimal Precision 69
Default Value 69
Descriptor 63, 66
Help String 61
International Engineering Units 69
International Help String 69
International Parameter Name 69
Link (Multiplier, Divider, Base, and Offset)
69
Link Path 66
Maximum Value 69
Minimum Value 69

Parameter Name 68
Scaling (Multiplier, Divider, Base, and
Offset) 69

G

general information, EDS files 14

General tab 27

Generic Device AOP 13

GroupN keyword 71

H

Help String field 61

I

icon files

guidelines 15
Logix Designer 15

input

data structures 19
parameters 19
tags 19

installing EDS files 14

internationalization

EDS files 25
keywords 25

Internet Protocol tab 38

K

keywords

AssemN/AssemExaN 62
Catalog 46
CfgAssembly 62
CfgAssemblyExa 62
ClassN 51
ConnectionN 51, 71
CreateDate 46
CreateTime 46
Descriptor 62
EnumN 71
GroupN 71
MajRev 50
MinRev 50
ModDate 46
ModTime 46
ParamN 65
ProdName 48, 70
ProdTypeStr 50
Revision 46
SafetyCfgAssembly keyword 62
VariantN/VariantExaN 65
VendName 49

L

limitations of EDS AOP feature 13

literature, related 14**Logix Designer**

About Module Profile dialog box 29
 Advanced Network Configuration dialog
 box 42
 and EDS files 13
 Connection tab 33
 Custom Data Type dialog box 32
 EDS Hardware Installation Tool 14
 EDS icon files 15
 General tab 27
 Internet Protocol tab 38
 Module Definition dialog box 30
 Module Info tab 36
 Network tab 41
 Port Configuration tab 39
 Port Diagnostics dialog box 40
 Safety Configuration tab 37
 Safety tab 35
 Time Sync tab 43

M**MajRev keyword 50****Minimum EDS file example 81****MinRev keyword 50****ModDate and ModTime keywords 46****Module Definition dialog box 30****module developers, requirements 13****Module Info tab 36****module-defined data types**

member naming 79

naming 77

multiple languages 25**N****Name field 63****naming**

module-defined data type members 79

module-defined data types 77

Network tab 41**O****ODVA**

conformance testing 14

EZ-EDS authoring tool 14

Output

data structures 19

parameters 19

tags 20

P**parameters**

bit-enumerated 20, 23

Configuration 22

Input 19

Output 19

ParamN enumeration 69**ParamN keyword 65**

Data Size field 68

Data Type field 67

Decimal Precision field 69

Default Value field 69

Descriptor field 66

Help String field 61

International Engineering Units field 69

International Help String field 69

International Parameter Name field 69

 Link (Multiplier, Divider, Base, and Offset)
 fields 69

Link Path field 66

Maximum Value field 69

Minimum Value field 69

Parameter Name field 68

 Scaling (Multiplier, Divider, Base, and
 Offset) fields 69

Units String field 69

Path field 61, 63**Port Configuration tab 39****Port Diagnostics dialog box 40****ProdName keyword 48, 70****ProdTypeStr keyword 50****profile, definition 14****Proxy Config Size and Format fields 57****R****references 14****registering EDS files 14****related literature 14****requirements for module developers 13****S****Safety Configuration tab 37****Safety tab 35****SafetyCfgAssembly keyword 62****sample EDS files 81****Size field 63****supported devices and buses 13****T****T->O and O->T Format fields 56****T->O and O->T RPI fields 55****T->O and O->T Size fields 55****tags**

Configuration 22

Input 20

Output 20

Target Config Format field 57**Target Config Size field 57****Time Sync tab 43**

translation

- internationalization 70
- multiple language support 25

Transport field 52

Trigger field 52

U

unsupported devices 13

V

VariantN/VariantExaN keywords 65

VendName keyword 49

Rockwell Automation support

Use these resources to access support information.

Technical Support Center	Find help with how-to videos, FAQs, chat, user forums, and product notification updates.	rok.auto/support
Knowledgebase	Access Knowledgebase articles.	rok.auto/knowledgebase
Local Technical Support Phone Numbers	Locate the telephone number for your country.	rok.auto/phonesupport
Literature Library	Find installation instructions, manuals, brochures, and technical data publications.	rok.auto/literature
Product Compatibility and Download Center (PCDC)	Get help determining how products interact, check features and capabilities, and find associated firmware.	rok.auto/pcdc

Documentation feedback

Your comments help us serve your documentation needs better. If you have any suggestions on how to improve our content, complete the form at rok.auto/docfeedback.

Waste Electrical and Electronic Equipment (WEEE)



Note: At the end of life, this equipment should be collected separately from any unsorted municipal waste.





Rockwell Automation maintains current environmental information on its website at rok.auto/pec.

Allen-Bradley, expanding human possibility, Logix, Rockwell Automation, and Rockwell Software are trademarks of Rockwell Automation, Inc.

EtherNet/IP is a trademark of ODVA, Inc.

Trademarks not belonging to Rockwell Automation are property of their respective companies.

Rockwell Otomasyon Ticaret A.Ş. Kar Plaza İş Merkezi E Blok Kat:6 34752, İçerenköy, İstanbul, Tel: +90 (216) 5698400 EEE Yönetmeliğine Uygundur

Connect with us.    

rockwellautomation.com — expanding **human possibility**™

AMERICAS: Rockwell Automation, 1201 South Second Street, Milwaukee, WI 53204-2496 USA, Tel: (1) 414.382.2000, Fax: (1) 414.382.4444

EUROPE/MIDDLE EAST/AFRICA: Rockwell Automation NV, Pegasus Park, De Kleetlaan 12a, 1831 Diegem, Belgium, Tel: (32) 2 663 0600, Fax: (32) 2 663 0640

ASIA PACIFIC: Rockwell Automation, Level 14, Core F, Cyberport 3, 100 Cyberport Road, Hong Kong, Tel: (852) 2887 4788, Fax: (852) 2508 1846