SIEMENS

Getting process values via CIP with SIMATIC S7

S7-1500/ S7-1200/ EtherNet/IP

https://support.industry.siemens.com/cs/ww/en/view/109782317

Siemens
Industry
Online
Support

Legal information

Warning notice system

This manual contains notices you have to observe in order to ensure your personal safety, as well as to prevent damage to property. The notices referring to your personal safety are highlighted in the manual by a safety alert symbol, notices referring only to property damage have no safety alert symbol. These notices shown below are graded according to the degree of danger.

⚠ DANGER	**indicates that death or severe personal injury will result if proper precautions are not taken.**

⚠ WARNING	**indicates that death or severe personal injury may result if proper precautions are not taken.**

⚠ CAUTION	**indicates that minor personal injury can result if proper precautions are not taken.**

NOTICE	**indicates that property damage can result if proper precautions are not taken.**

If more than one degree of danger is present, the warning notice representing the highest degree of danger will be used. A notice warning of injury to persons with a safety alert symbol may also include a warning relating to property damage.

Qualified Personnel

The product/system described in this documentation may be operated only by personnel qualified for the specific task in accordance with the relevant documentation, in particular its warning notices and safety instructions. Qualified personnel are those who, based on their training and experience, can identify risks and avoiding potential hazards when working with these products/systems.

Proper use of Siemens products

Note the following:

⚠ WARNING	**Siemens products may only be used for the applications described in the catalog and in the relevant technical documentation. If products and components from other manufacturers are used, these must be recommended or approved by Siemens. Proper transport, storage, installation, assembly, commissioning, operation and maintenance are required to ensure that the products operate safely and without any problems. The permissible ambient conditions must be complied with. The information in the relevant documentation must be observed.**

Siemens AG
Division Digital Factory
Postfach 48 48
90026 NÜRNBERG
Germany

V1.0.2
Ⓟ 12/2023 Subject to change

Trademarks

All names identified by ® are registered trademarks of Siemens AG. The remaining trademarks in this publication may be trademarks whose use by third parties for their own purposes could violate the rights of the owner.

Disclaimer of Liability

We have reviewed the contents of this publication to ensure consistency with the hardware and software described. Since variance cannot be precluded entirely, we cannot guarantee full consistency. However, the information in this publication is reviewed regularly and any necessary corrections are included in subsequent editions.

Security information

Siemens provides products and solutions with industrial security functions that support the secure operation of plants, systems, machines, and networks.

In order to protect plants, systems, machines, and networks against cyber threats, it is necessary to implement – and continuously maintain – a holistic, state-of-the-art industrial security concept. Siemens' products and solutions only form one element of such a concept.

Customer is responsible to prevent unauthorized access to its plants, systems, machines, and networks. Systems, machines, and components should only be connected to the enterprise network or the internet if and to the extent necessary and with appropriate security measures (e.g., use of firewalls and network segmentation) in place.

Additionally, Siemens' guidance on appropriate security measures should be taken into account. For more information about industrial security, please visit https://www.siemens.com/industrialsecurity.

Siemens' products and solutions undergo continuous development to make them more secure. Siemens strongly recommends to apply product updates as soon as available and to always use the latest product versions. Use of product versions that are no longer supported, and failure to apply latest updates may increase customer's exposure to cyber threats.

To stay informed about product updates, subscribe to the Siemens Industrial Security RSS Feed under https://www.siemens.com/cert.

Siemens AG
Division Digital Factory
Postfach 48 48
90026 NÜRNBERG
Germany

V1.0.2
Ⓟ 12/2023 Subject to change

Table of contents

1 Preface

Purpose

This document contains information about the “LCCF_CipClient” function block for SIMATIC S7-1200 and S7-1500. The previously available blocks “PUT_R” and “GET_R” are being replaced with this single block, which is part of the LCCF (Library of Competitor Conversion Functions).

As with the beforementioned blocks the “LCCF_CipClient” allows easy access to process values inside a supported Rockwell Automation control system. It is using the control systems native protocol preventing any changes inside the already running program.

For this matter, the block is ideally used in scenarios, where the control system program cannot be changed for the extension with a SIMATIC system.

Core content

This document explains the use of the “LCCF_CipClient” function block for SIMATIC to read or write process values inside the Rockwell Automation controller. The only necessary knowledge is, that the user can program the SIMATIC controller.

No changes inside the Rockwell Automation controller are necessary.

The following core issues are covered in this document:

- Purpose of the function block
- Parameterization
- Data Exchange with a CIP server

Required basic knowledge

General knowledge in communications over Ethernet and programming and configuring the S7-1200 or S7-1500 with the TIA Portal is assumed and will not be part of this document. It is also assumed that the terms Server and Client and their meaning are familiar to the reader.

Delimitation

The document does not describe:

- How to setup Ethernet networks
- How to assign IP addresses and the split into subnets
- How to configure the controllers in this example

Basic knowledge about the above topics is assumed.

Validity

This document is valid for the following components

- TIA Portal
- SIMATIC S7 Controller

The block has been tested with the following Rockwell Automation systems

- ControlLogix/ CompactLogix
- SLC 551
- Micro850

The following hardware and software are used throughout this document

Table 1-1: used components

Name	Part number	Version
SIMATIC S7-1511-1PN	6ES7 511-1AK01-0AB0	V2.6 (or above)
SIMATIC S7-1215C	6ES7 215-1AG40-0XB0	V4.2 (or above)
SIMATIC S7-1505S	6ES7 672-5AC00-0YA0	V2.1 (or above)
TIA Portal STEP7 Prof.		V15.1 Update 4 (or above)
SCALANCE X208	6GK5 208-0BA10-2AA3	
CompactLogix	1769-L27ERM-QBFC1B	V33.13
Studio 5000	9324-RLD700NXENE	V33.13

2 Introduction

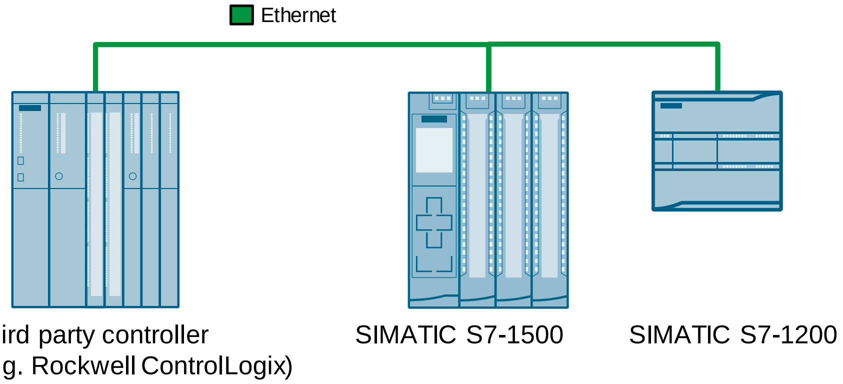
2.1 Description

Often SIMATIC controller exchange data with other controller, peripheral systems, and upper-level control systems or with MES or SCADA systems. To do this in a meaningful manner the communication partners need to use the same protocol

The following application example demonstrates the use of CIP/ PCCC as the protocol to perform this data exchange.

A simplified setup is shown below

Figure 2-1: simplified setup



The physical connection between the controller uses Ethernet cables, such as PROFINET cable. A SCALANCE X208 switch has been put in between, but is not shown, as direct connections are possible as well.

As the “LCCF_CipClient” block exists for both SIMATIC families, this application example is applicable to S7-1200 and S7-1500 systems. In the above schematic only one of the SIMATIC controller is necessary. As a CIP server may be able to serve multiple clients, multiple different SIMATIC controller may be connected to the CIP server. The number of connectable clients depends on the capabilities of the CIP server

2.2 Common Industrial Protocol

The Common Industrial Protocol (for short CIP) is defined by the Open DeviceNet Vendor Association (ODVA®). This protocol is an object-oriented mechanism to access process values as well as configuration and diagnostic data of a conformant device.

CIP defines several objects, which in turn provide services to access the attributes of that object. Each attribute has a certain meaning, which is either predefined or vendor specific.

The “LCCF_CipClient” function block uses two different objects depending on the configured CIP server.

One object is the PCCC object. The LCCF_CipClient implements a few of its services.

The other object is the Message Router (MR) object. The LCCF_CipClient block implements also only a few defined services.

Each such service execution is realized as a “Request – Response” pair.

2.3 Programmable Controller Communication Commands

The Programmable Controller Communication Commands (for short PCCC, PC³ or PC cube) are a standardized set of services defined in the PCCC object. These services allow the access to process values inside the automation system implementing this object.

The SIMATIC implementation provided in the “LCCF_CipClient” function block implements two services of the PCCC object.

- SLC typed protected logical read with 3 address fields
- SLC typed protected logical write with 3 address fields

These two services are sufficient to access a wide range of automation system internal memories. There are Rockwell controller families supporting the PCCC object directly. These families are:

- SLC 5/0x programmed with RS Logix 500
  therefore, known as SLC500
- MicroLogix 1x00 programmed with RS Logix 500

Both families of Rockwell Automation PLC have the same memory addressing schema. Variables are collected in so called files of common data type. This means, that variables of the same data type are stored in a file of that data type.

NOTE

Another legacy Rockwell Automation system is the PLC-5. It also supports the PCCC object and has the same addressing schema as the SLC and MicroLogix. The PLC-5 uses different commands to provide access to the object’s attributes. These services are:

- PLC-5 typed read
- PLC-5 typed write
- PLC-5 typed masked write

These services come with 3,4 or even 7 address fields.

This Rockwell Automation system is not in scope of the LCCF_CipClient function block.

2.4 Message Router

The Message Router object defines services, which allow the access to process variables inside a different family of Rockwell controller.

The SIMATIC implementation provided in the “LCCF_CipClient” function block implements two services of the Message Router object.

- CIP_DataTableRead
- CIP_DataTableWrite

These services are supported by the following Rockwell Automation systems:

- CompactLogix
- ControlLogix

and their respective Safety counterparts called GuardLogix.

2.5 Description

As with the “PUT_R” and “GET_R” blocks the “LCCF_CipClient” block allows an easy access to Rockwell Automation control systems (a.k.a. Allen Bradley). It can read and write process values inside the controllers without any changes necessary inside the controllers.

The” LCCF_CipClient” acts as a client requesting the services from the controller. It uses the controller native CIP (Common Industrial Protocol) encapsulated on Ethernet.

This document will show you how easy it is to achieve the process value access. What needs to be configured and what can be achieved.

2.6 Function principle

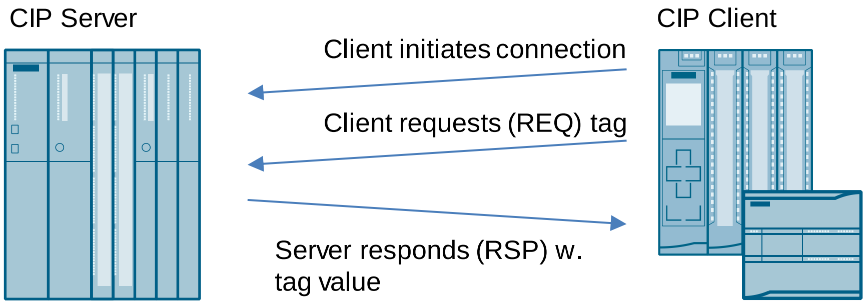
The application example demonstrates the necessary steps to set up the communication and to start the data exchange between a SIMATIC controller and a third-party system.

The communication with CIP/ PCCC uses the Client-Server principle. In this application the CIP server is being realized by the Rockwell ControlLogix controller. The CIP client is realized by either a SIMATIC S7-1500 or a SIMATIC S7-1200. Only one of the SIMATIC systems is necessary, but both or even multiple can be operated at the same time.

The CIP protocol is embedded into the payload of the EtherNet/IP packets sent to the Rockwell Automation system.

The CIP client establishes the connection and requests data from the CIP server to use in its own automation program. The LCCF_CipClient block also allows writing to process values inside the Rockwell controller.

Figure 2-2: schematic functional principle


CIP Server
CIP Client
Client initiates connection
Client requests (REQ) tag
Server responds (RSP) w. tag value

With the “LCCF_CipClient” function block a user can establish a connection to a directly accessible Rockwell Automation controller system. Once enabled the function block will automatically contact the Rockwell Automation controller. It establishes a TCP/IP connection and negotiates a communication session.

Within the session context all parameterized process tags will be exchanged with the Rockwell Automation controller. Hereby the “LCCF_CipClient” block executes both Reading and Writing requests within a single request.

For this the CIP client utilizes one TCP connection.

Advantages

This application offers the following advantages:

- Direct connection between controllers of different vendors
- No gateways or protocol converters required
- Modern and easy to install network

2.7 Scope of delivery

The application example consists of a TIA Portal project containing two SIMATIC controller.

The program in the S7-1200 is identical to the program in the S7-1500 controller.

2.8 What is new?

As mentioned previously, the provided block replaces the previous blocks “PUT_R” and “GET_R”. The “LCCF_CipClient” block has several advantages:

- Auto allocation of Connection ID
- Single block for both reading and writing of tags
- Access to structure and array elements
- Reduction in communication load
- Configuration Tool with direct import into TIA Portal project
- Recently, Added array handling capability for CLX Syntax (ControlLogiX / Compact Logix)

Unfortunately, the above advantages come with a disadvantage:

- access to memory areas, larger than 64 Bit (LWORD) is no longer possible

3 Commissioning

3.1 Preparation

As preparation for the application example to work the above-mentioned hardware components should be placed into a rack or on a solid table to prevent slip or fall.

⚠ WARNING

Risk of electric shock

To operate this application example the connection of the above hardware to electrical power is required. Disregarding local regulations and common sense may cause an electric shock and because of that injury or death.

Always follow the rules for working with electrical equipment.

Further download the TIA Portal project from the SIOS entry or download the latest version of the LCCF (Library of Competitor Conversion Functions). Store the downloaded file onto your local computer for later use.

Hardware components

In the following description we will be using the below listed hardware. In your scenario you may exchange one or the other component with an equivalent.

Table 3-1: recommended hardware components

Component	No.	Order number	Note
SIMATIC S7-1500	1	6ES7 511-1AK01-0AB0	as of firmware V2.6 onwards
SIMATIC S7-1215C	1	6ES7 215-1AG40-0AA0	as of firmware V4.2 onwards
SIMATIC S7-150xS	1	6ES7 672-xAC00-0YA0	as of firmware V2.1 onwards
ControlLogix L72S	1	1756-5572S	only one of these systems or equivalent systems required
CompactLogix	1	1769-L27ERM-QBFC1B	
SLC500	1	1747-L551	

NOTE Even though the used ControlLogix is a GuardLogix Safety rated controller, the “LCCF_CipClient” cannot access Safety Tags. Therefore, the GuardLogix is operated as regular ControlLogix.

Software components

In this document the below listed software components are being used.

Table 3-2: recommended software components

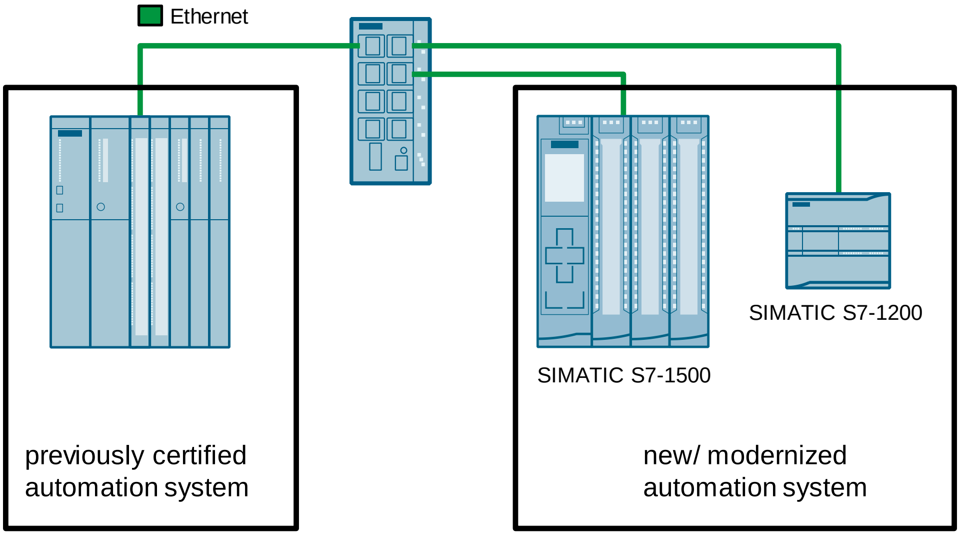
Component	No.	Order number	Note
TIA Portal V15.1 (incl. Openness)	1		used for programming the SIMATIC controller
Studio 5000 V33.13	1	9324-RLD700NXENE	used to define the tags inside the Rockwell ControlLogix or CompactLogix systems
RSLogix 500	1	9324-RL0300DEM	used to define data files inside the legacy SLC 500 or MicroLogix system

As an alternative to TIA Portal in V15.1 also V16, V17 and V18 can be used. The sample projects and the libraries exist for all the above TIA Portal versions.

3.2 Connecting the hardware components

A possible general setup is shown below.

Figure 3-1: simplified network setup


🟩 Ethernet
SIMATIC S7-1200
SIMATIC S7-1500
previously certified automation system
new/ modernized automation system

Setup

The communication is realized using Ethernet as the communication network. A direct connection using a cross over CAT5 cable or via a network switch with straight CAT5 cables can be used.

As already mentioned before, only one of the SIMATIC controller is necessary, but both can be operated simultaneously as well.

4 Configuration/Engineering

4.1 Creating and managing projects

Before we can parameterize the “LCCF_CipClient” block and configure the tag list, we need to get a list of process values (tags) which we want to access.

This can be achieved either by getting a printout of the tag list, which is in question or browsing the controller to access.

The relevant information is slightly different for the currently supported Rockwell Automation systems.

In general, the different data types are used for certain things

Table 4-1: data type usage

to store a …	… use this data type	
Bit	BOOL	
Bit array	BOOLEAN ARRAY (32 Bit)	
8 Bit Integer	SINT	
16 Bit Integer	INT	
32 Bit Integer	DINT	
32 Bit Float	REAL	
32 Bit Milliseconds	TIMER	Timer is a structure. Only individual elements can be addressed
32 Bit of range	COUNTER	Counter is a structure. Only individual elements can be addressed

Examples are shown in the following sub chapters on the following page.

ControlLogix/ CompactLogix

In the current systems of ControlLogix or CompactLogix the most important information is the name of the process value.

Addressable process values are shown as examples in the below table.

Table 4-2: Logix5000 data

To access …	… specify the tag name	Example
single integer tag named „parts”	tagname	“parts”
6th element of an array of REALs named “setpoints”	tagname[index]	“setpoints[6]”
single integer [2,5,257] of a three-dimensional array named “profile”	tagname[index, index, index]	“profile[2,5,257]”
accumulated value of a timer named “dwell3”	tagname.ACC	“dwell3.ACC”
wear member inside the singular STRUCT_A structure “struct1”	tagname.structure element name	“struct1.wear”
5th “hourlyCount” element of a single STRUCT_B structure “struct2”	tagname.structure element name[index]	“struct2.hourlyCount[5]”

NOTE

Casing of the tag name to access in a ControlLogix or CompactLogix system is important. The following tags are not the same.

- “parts” <> “Parts”
- “parts” <> “PARTS”
- “parts” <> “pArts”

Make sure the spelling is correct, otherwise the tag cannot be accessed.

Misspelling can be avoided, when using the CIP service configuration tool.

Micro800

Another current control system designed for micro automation is the Micro800 series controller. Tags are addressable in the same way as in the above mentioned CompactLogix and ControlLogix systems. They are referenced by their name.

As the Micro800 has a different addressing schema, the parameter “slot” needs to be set to 127 (fix). This value indicates the changed addressing schema.

NOTE

As with the ControlLogix and CompactLogix systems the spelling of the tag’s name is important.

SLC – Small Logic Controller

The below shown addressing examples are applicable to the SLC family of controllers. They are valid for the MicroLogix family of controllers as well.

The tag name is identical with the addressed memory location extended by the prefix ‘@’.

Table 4-3 SLC/MicroLogix data

To access …	… specify the tag name	Example
The 1st single integer tag in N file 7	@filename filenumber**:**element number	“@N7:0”
The 3rd bit of the binary file B3 in word 4	@filename filenumber**:**element number**/**bitnumber	“@B3:4/3”
the 9th element of the REAL file F80	@filename filenumber**:**element number	“@F80:9”
the Accumulated value of the 3rd counter in file C5	@filename filenumber**:**element number**.ACC**	“@C5:3.ACC”
the Preset time value of the 6th timer in file T4	@filename filenumber**:**element number**.PRE**	“@T4:6.PRE”

NOTE

The legacy Rockwell Automation system PLC-5 uses the same syntax as the SLC and MicroLogix with a different prefix. This allows the LCCF_CipClient function block to distinguish between SLC/MicroLogix services and PLC-5 services to invoke.

4.2 Creating the tag lists

As mentioned, there are two possible ways to create a list of tags, which the “LCCF_CipClient” block should access.

As prerequisite a TIA Portal project containing the PLC of your choice should already exist.

In this document this TIA Portal project is called “LCCF_CIPClient” and uses TIA Portal V15.1 and applicable for onwards versions.

4.2.1 Manual tag list creation

When you decide to create the list of tags manually, you need to take extra care for tags, which should be written. In this case it is important to know the data type code. The codes are listed below.

Table 4-4: Type codes for data types

Data type	type code	Note
BOOL	16#00C1	a Boolean value
SINT	16#00C2	8-bit signed integer
INT	16#00C3	16-bit signed integer
DINT	16#00C4	32-bit signed integer
REAL	16#00CA	32-bit single precision floating point number
DWORD	16#00D3	32-bit collection of bit

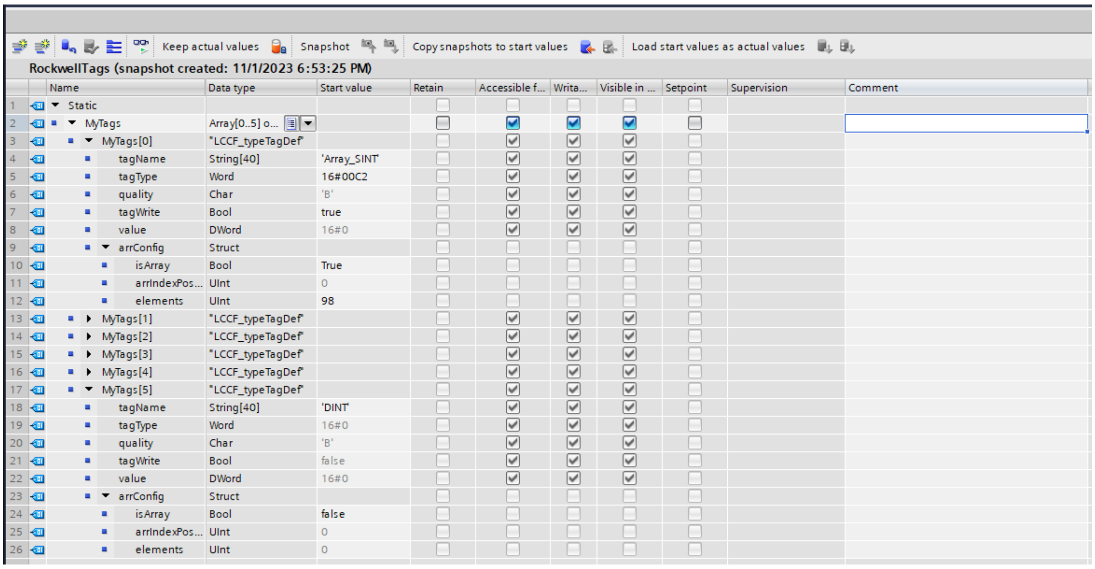
Each tag is stored in a variable of type “LCCF_typeTagDef”. The tags should be stored in an array of any length. The type is provided as part of the “LCCF”.

A data block containing some tag definitions is shown below. Here an array of 6 tags has been defined.

For illustration purposes two of the defined tags are shown in detail.

- MyTags[0] is called “Array_SINT”, is of type SINT with type code 16#00C2(refer to the above table), isArray set to true as this tag is an array, arrIndexPosition is set to 0 (values of Array_SINT will be placed in Index 0 of the ArrayByteList Datablock, Figure 4-3) and Elements is set to 98 (Array_SINT is an array of SINT having 98 elements)
- MyTags[5] is called “DINT” and isArray is set to False as it’s a non-array tag

Figure 4-1: TIA Portal Datablock with tag definitions

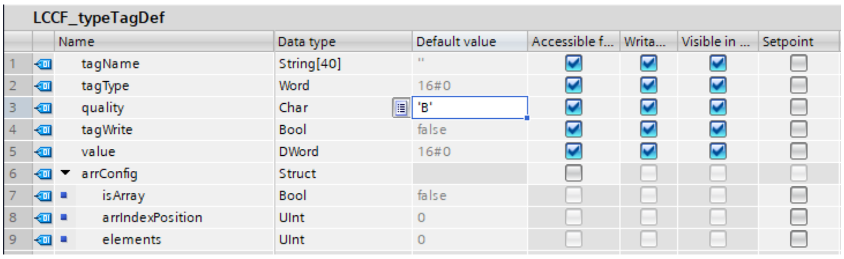

Keep actual values
Snapshot
Copy snapshots to start values
Load start values as actual values

RockwellTags (snapshot created: 11/1/2023 6:53:25 PM)

	Name	Data type	Start value	Retain	Accessible f...	Writa...	Visible in ...	Setpoint	Supervision	Comment
1	Static			□	□	□	□	□		
2	MyTags	Array[0..5] o...		□	☑	☑	☑	□		
3	MyTags[0]	"LCCF_typeTagDef"		□	☑	☑	☑	□		
4	tagName	String[40]	'Array_SINT'	□	☑	☑	☑	□		
5	tagType	Word	16#00C2	□	☑	☑	☑	□		
6	quality	Char	'B'	□	☑	☑	☑	□		
7	tagWrite	Bool	true	□	☑	☑	☑	□		
8	value	DWord	16#0	□	☑	☑	☑	□		
9	arrConfig	Struct		□	□	□	□	□		
10	isArray	Bool	True	□	□	□	□	□		
11	arrIndexPos...	UInt	0	□	□	□	□	□		
12	elements	UInt	98	□	□	□	□	□		
13	MyTags[1]	"LCCF_typeTagDef"		□	☑	☑	☑	□		
14	MyTags[2]	"LCCF_typeTagDef"		□	☑	☑	☑	□		
15	MyTags[3]	"LCCF_typeTagDef"		□	☑	☑	☑	□		
16	MyTags[4]	"LCCF_typeTagDef"		□	☑	☑	☑	□		
17	MyTags[5]	"LCCF_typeTagDef"		□	☑	☑	☑	□		
18	tagName	String[40]	'DINT'	□	☑	☑	☑	□		
19	tagType	Word	16#0	□	☑	☑	☑	□		
20	quality	Char	'B'	□	☑	☑	☑	□		
21	tagWrite	Bool	false	□	☑	☑	☑	□		
22	value	DWord	16#0	□	☑	☑	☑	□		
23	arrConfig	Struct		□	□	□	□	□		
24	isArray	Bool	false	□	□	□	□	□		
25	arrIndexPos...	UInt	0	□	□	□	□	□		
26	elements	UInt	0	□	□	□	□	□		

The used type is shown in detail below.

Figure 4-2: TIA Portal data type “LCCF_typeTagDef”



LCCF_typeTagDef

	Name	Data type	Default value	Accessible f...	Writa...	Visible in ...	Setpoint
1	tagName	String[40]	''	☑	☑	☑	□
2	tagType	Word	16#0	☑	☑	☑	□
3	quality	Char	'B'	☑	☑	☑	□
4	tagWrite	Bool	false	☑	☑	☑	□
5	value	DWord	16#0	☑	☑	☑	□
6	▼ arrConfig	Struct		□	□	□	□
7	isArray	Bool	false	□	□	□	□
8	arrIndexPosition	UInt	0	□	□	□	□
9	elements	UInt	0	□	□	□	□

As you can see, the type has several members which will be explained in the following table.

NOTE

The access direction in the below table is seen from the LCCF_CipClient block’s viewpoint.

This means that the direction “read” indicates a reading access from the block. The block reads this value and acts accordingly.

The value “write” means that the block disregards any value in here and overwrites it with the value it deems fit.

An indication of “read/write” means that, depending on other field values, this fields value is either read or written.

Table 4-5: type “LCCF_typeTagDef”

Member	data type	access direction	comment
tagName	STRING[40]	read	Contains the name of the process value to access. • Use the prefix ‘@’ for access to SLC/ MicroLogix systems. • Use the tag name to access ControlLogix or CompactLogix systems
tagType	WORD	read/ write	contains the type-code for the process value accessed. This is relevant for writing access. Refer to the table “Type codes for data types”
quality	CHAR	write	contains a quality code for the value provided in the member ‘value’. ‘G’ means ‘GOOD’, the value is valid and can be used for further processing. ‘B’ means ‘BAD’, the value is not valid and shall not be used for further processing.
tagWrite	BOOL	read	TRUE indicates a writing access, which will transfer the value provided in the member ‘value’ to the Rockwell Automation system. FALSE indicates a reading access, which will store the value from the Rockwell Automation system into the member ‘value’.

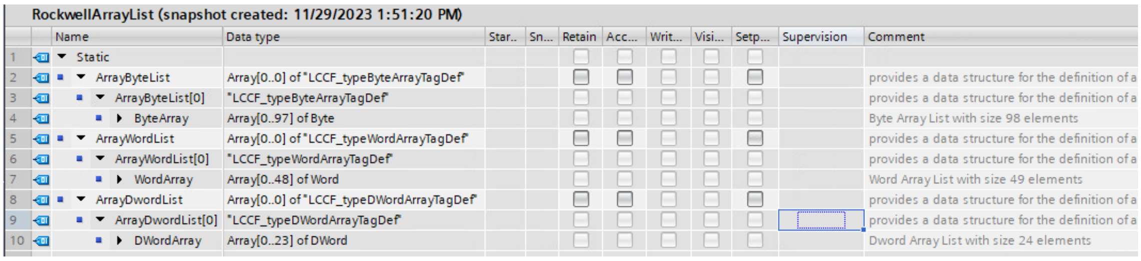
Member	data type	access direction	comment
value	DWORD	read/ write	In this member the value is stored, which is either retrieved from (tagWrite = FALSE) the Rockwell Automation system or which is being written to (tagWrite = TRUE) the Rockwell Automation system. Note: - Array values will be stored in ArrayTagDef according to size (Byte, Word, Dword), see below table.
arrConfig	Struct		Configuration of tag which is an array
isArray	BOOL	read	TRUE indicates that the tag is an array FALSE indicates that the tag is not an array
arrIndexPosition	UINT	read	Position in the relevant Array List where the array's values will be placed
elements	UINT	read	Number of Elements in the array tag

NOTE

When you keep the 'tagWrite' member at its default 'FALSE' the first access is a 'Reading' access, which also populates the type-code into 'tagType'.

You may change the 'tagWrite' member's value during operation at any time via your program. This will freeze the value on the Rockwell Automation side as the same value is being written if the 'tagWrite' member remains 'TRUE'.

Figure 4-3: TIA Portal Datablock with Array list



RockwellArrayList (snapshot created: 11/29/2023 1:51:20 PM)

	Name	Data type	Star..	Sn...	Retain	Acc...	Writ...	Visi...	Setp...	Supervision	Comment
1	Static				□	□	□	□	□		
2	ArrayByteList	Array[0..0] of "LCCF_typeByteArrayTagDef"			□	□	□	□	□		provides a data structure for the definition of a
3	ArrayByteList[0]	"LCCF_typeByteArrayTagDef"			□	□	□	□	□		provides a data structure for the definition of a
4	ByteArray	Array[0..97] of Byte			□	□	□	□	□		Byte Array List with size 98 elements
5	ArrayWordList	Array[0..0] of "LCCF_typeWordArrayTagDef"			□	□	□	□	□		provides a data structure for the definition of a
6	ArrayWordList[0]	"LCCF_typeWordArrayTagDef"			□	□	□	□	□		provides a data structure for the definition of a
7	WordArray	Array[0..48] of Word			□	□	□	□	□		Word Array List with size 49 elements
8	ArrayDwordList	Array[0..0] of "LCCF_typeDWordArrayTagDef"			□	□	□	□	□		provides a data structure for the definition of a
9	ArrayDwordList[0]	"LCCF_typeDWordArrayTagDef"			□	□	□	□	□		provides a data structure for the definition of a
10	DWordArray	Array[0..23] of DWord			□	□	□	□	□		Dword Array List with size 24 elements

The used types are shown in detail below.

Table 4-6: type “LCCF_typeByteArrayTagDef”

Member	data type	access direction	comment
ByteArray	Array[0..97] of Bytes	read/ write	Same as Value member, however, values of array elements are stored sequentially from 1st element to last element similarly in Rockwell e.g: Rockwell : Array_SINT[11] or [3,2,2] – elements: 0-11 Siemens : ByteArray[0..11] elements: 0-11 Tagtype: (SINT) would be stored in this array. Max number of elements is 98

Table 4-7: type “LCCF_typeWordArrayTagDef”

Member	data type	access direction	comment
WordArray	Array[0..48] of Word	read/ write	Same as Value member, however, values of array elements are stored sequentially from 1st element to last element similarly in Rockwell e.g: Rockwell : Array_INT[11] or [3,2,2] elements: 0-11 Siemens : WordArray[0..11] elements: 0-11 Tagtype: (INT) would be stored in this array. Max number of elements is 49

Table 4-8: type “LCCF_typeDWordArrayTagDef”

Member	data type	access direction	comment
DwordArray	Array[0..23] of Dword	read/ write	Same as Value member, however, values of array elements are stored sequentially from 1st element to last element similarly in Rockwell e.g: Rockwell : Array_DINT[11] or [3,2,2] elements: 0-11 Siemens : DWordArray[0..11] elements: 0-11 Tagtype: (DINT, DWORD, REAL) would be stored in this array. Max number of elements is 24

NOTE

Number of Elements limit for Array = 98 / Number of Bytes per elements

For example, for Array of SINT, limit would be 98/1 = 98 elements

Best Practice of forming the Tags List with the first tag having the lowest size to the last tag(e.g., BOOL) having the largest size of all (e.g., Array of DINT of size 24)

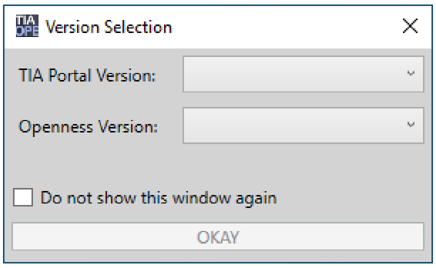
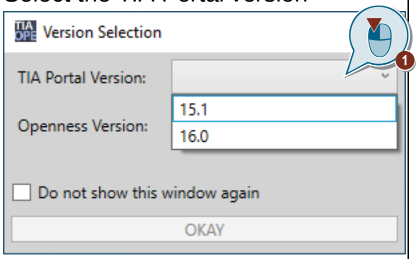
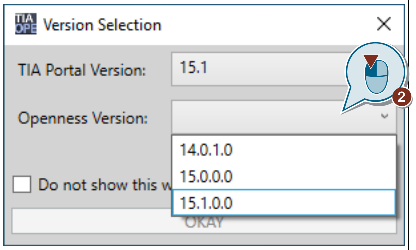
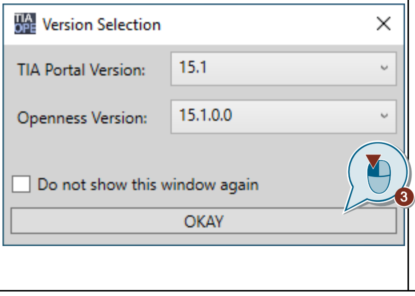
Array Capabilities are valid for CLX Syntax (ControlLogiX / Compact Logix)

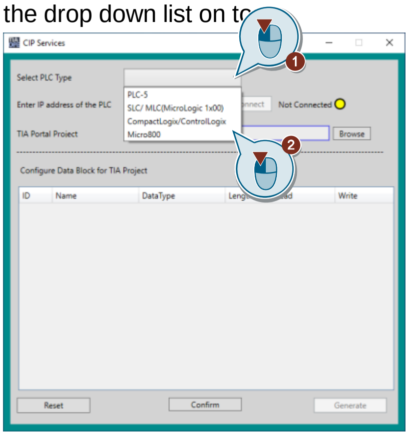
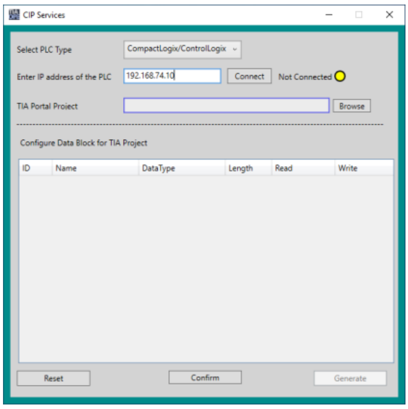
4.2.2 Tool based tag list creation

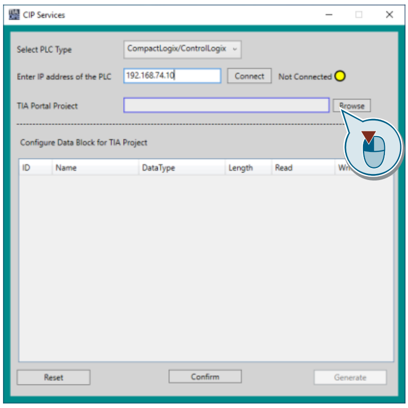
For your convenience we created a tool, which can create a SIMATIC data block with selected tag definitions in it. It also has the capability to import the data block directly into a TIA Portal project. Its use is explained on the following pages.

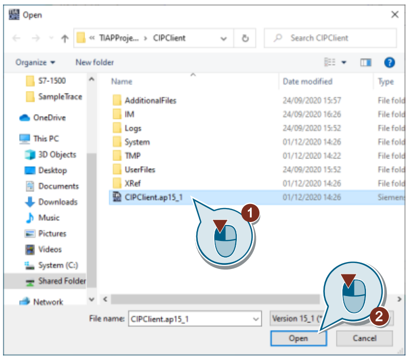
The tool uses TIA Portal Openness for the import feature and can browse into Rockwell Automation controllers, which are directly connected to the computer running the “CIP Service Tool”

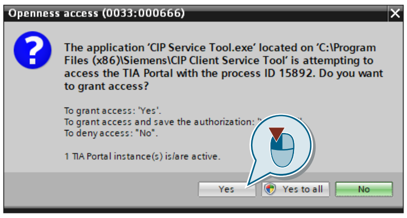
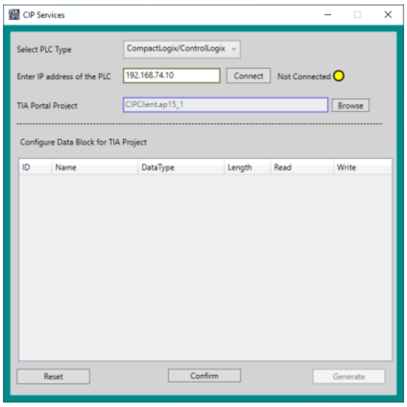
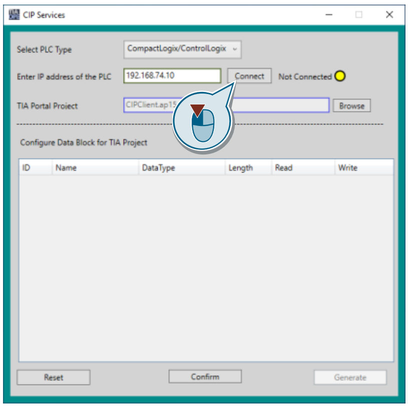
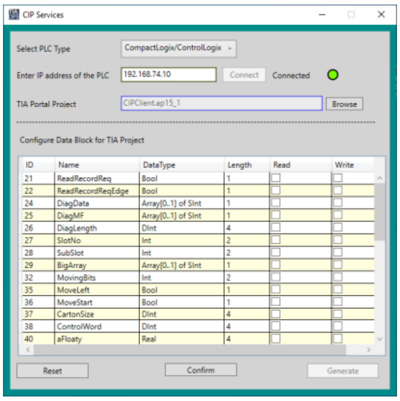
Table 4-9 TagList

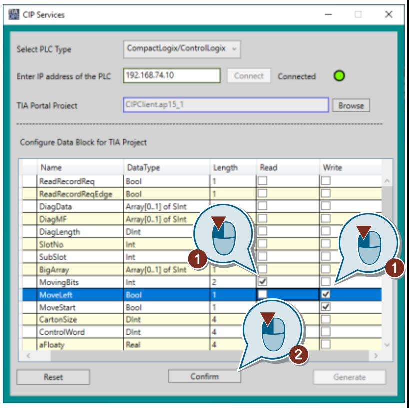
	What to do	Result
1.	start the CIP Service Tool 	A dialog box pops up allowing the selection of the TIA Portal version to be used.  Version Selection TIA Portal Version: Openness Version: □ Do not show this window again OKAY
2.	Select the TIA Portal version  Version Selection TIA Portal Version: Openness Version: 15.1 16.0 □ Do not show this window again OKAY and the Openness version you want to use.  Version Selection TIA Portal Version: 15.1 Openness Version: 14.0.1.0 15.0.0.0 15.1.0.0 □ Do not show this w OKAY finally confirm your selection with OKAY  Version Selection TIA Portal Version: 15.1 Openness Version: 15.1.0.0 □ Do not show this window again OKAY 	here we select TIA Portal V15.1, but TIA Portal V16 works the same way.

	What to do	Result
3.	As next we select the Rockwell Automation controller family from the drop down list on to  CIP Services Select PLC Type Enter IP address of the PLC TIA Portal Project PLC-5 SLC/ MLC(MicroLogic 1x00) CompactLogix/ControlLogix Micro800 Connect Not Connected Browse Configure Data Block for TIA Project ID Name DataType Length Read Write Reset Confirm Generate 1 2	here we select the ControlLogix/ CompactLogix controller family.
4.	Enter the IPv4 address the Rockwell Automation controller is accessible on.  Select PLC Type CompactLogix/ControlLogix Enter IP address of the PLC 192.168.74.10 CIP Services Connect Not Connected TIA Portal Project Browse Configure Data Block for TIA Project ID Name DataType Length Read Write Reset Confirm Generate	you need to adjust the here selected IPv4 address to yours. Make sure your engineering system computer can access the controller.

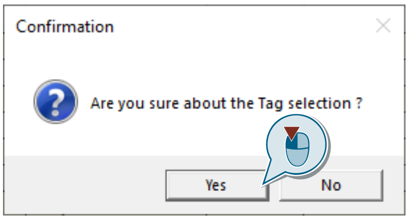
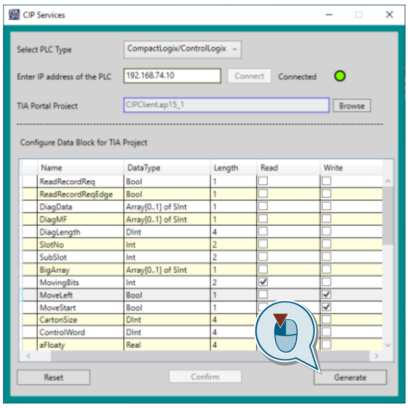
<table>
<tr><th></th><th>What to do</th><th>Result</th></tr>
<tr><td>5.</td><td>Browse for the TIA Portal project you want to perform the import.

CIP Services
Select PLC Type: CompactLogix/ControlLogix
Enter IP address of the PLC: 192.168.74.10 Connect Not Connected
TIA Portal Project: Browse
Configure Data Block for TIA Project
ID Name DataType Length Read Write
Reset Confirm Generate

Select the project from the File Open dialog

Open
« TIAPProje... › CIPClient
Search CIPClient
Organize ▾ New folder
S7-1500
SampleTrace
OneDrive
This PC
3D Objects
Desktop
Documents
Downloads
Music
Pictures
Videos
System (C:)
Shared Folder
Network
<table><tr><th>Name</th><th>Date modified</th><th>Type</th></tr><tr><td>AdditionalFiles</td><td>24/09/2020 15:57</td><td>File fold</td></tr><tr><td>IM</td><td>24/09/2020 16:26</td><td>File fold</td></tr><tr><td>Logs</td><td>24/09/2020 15:52</td><td>File fold</td></tr><tr><td>System</td><td>01/12/2020 14:26</td><td>File fold</td></tr><tr><td>TMP</td><td>01/12/2020 14:22</td><td>File fold</td></tr><tr><td>UserFiles</td><td>24/09/2020 15:52</td><td>File fold</td></tr><tr><td>XRef</td><td>01/12/2020 14:26</td><td>File fold</td></tr><tr><td>CIPClient.ap15_1</td><td>01/12/2020 14:26</td><td>Siemens</td></tr></table>File name: CIPClient.ap15_1 Version 15_1 (*
Open Cancel

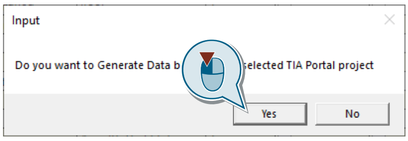
You may have to allow access to TIA Portal through the firewall

Openness access (0033:000666)
The application 'CIP Service Tool.exe' located on 'C:\Program Files (x86)\Siemens\CIP Client Service Tool' is attempting to access the TIA Portal with the process ID 15892. Do you want to grant access?
To grant access: 'Yes'.
To grant access and save the authorization:
To deny access: "No".
1 TIA Portal instance(s) is/are active.
Yes Yes to all No</td><td>you are now ready to generate a data block containing the tag definitions.

CIP Services
Select PLC Type: CompactLogix/ControlLogix
Enter IP address of the PLC: 192.168.74.10 Connect Not Connected
TIA Portal Project: CIPClient.ap15_1 Browse
Configure Data Block for TIA Project
ID Name DataType Length Read Write
Reset Confirm Generate</td></tr>
<tr><td>6.</td><td>Clock on “Connect” to establish a connection to the Rockwell Automation system and browse the available tags.

CIP Services
Select PLC Type: CompactLogix/ControlLogix
Enter IP address of the PLC: 192.168.74.10 Connect Not Connected
TIA Portal Project: CIPClient.ap15_1 Browse
Configure Data Block for TIA Project
ID Name DataType Length Read Write
Reset Confirm Generate</td><td>The list of tags in the center is now populated and allows the selection of tags to read and/ or to write to.

CIP Services
Select PLC Type: CompactLogix/ControlLogix
Enter IP address of the PLC: 192.168.74.10 Connect Connected
TIA Portal Project: CIPClient.ap15_1 Browse
Configure Data Block for TIA Project<table><tr><th>ID</th><th>Name</th><th>DataType</th><th>Length</th><th>Read</th><th>Write</th></tr><tr><td>21</td><td>ReadRecordReq</td><td>Bool</td><td>1</td><td>□</td><td>□</td></tr><tr><td>22</td><td>ReadRecordReqEdge</td><td>Bool</td><td>1</td><td>□</td><td>□</td></tr><tr><td>24</td><td>DiagData</td><td>Array[0..1] of SInt</td><td>1</td><td>□</td><td>□</td></tr><tr><td>25</td><td>DiagMF</td><td>Array[0..1] of SInt</td><td>1</td><td>□</td><td>□</td></tr><tr><td>26</td><td>DiagLength</td><td>DInt</td><td>4</td><td>□</td><td>□</td></tr><tr><td>27</td><td>SlotNo</td><td>Int</td><td>2</td><td>□</td><td>□</td></tr><tr><td>28</td><td>SubSlot</td><td>Int</td><td>2</td><td>□</td><td>□</td></tr><tr><td>29</td><td>BigArray</td><td>Array[0..1] of SInt</td><td>1</td><td>□</td><td>□</td></tr><tr><td>32</td><td>MovingBits</td><td>Int</td><td>2</td><td>□</td><td>□</td></tr><tr><td>35</td><td>MoveLeft</td><td>Bool</td><td>1</td><td>□</td><td>□</td></tr><tr><td>36</td><td>MoveStart</td><td>Bool</td><td>1</td><td>□</td><td>□</td></tr><tr><td>37</td><td>CartonSize</td><td>DInt</td><td>4</td><td>□</td><td>□</td></tr><tr><td>38</td><td>ControlWord</td><td>DInt</td><td>4</td><td>□</td><td>□</td></tr><tr><td>40</td><td>aFloaty</td><td>Real</td><td>4</td><td>□</td><td>□</td></tr></table>Reset Confirm Generate</td></tr>
</table>

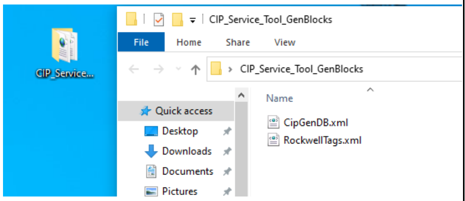
<table>
<tr><th></th><th>What to do</th><th>Result</th></tr>
<tr><td>7.</td><td>Make the selection and “Confirm” your selection.


Select PLC Type: CompactLogix/ControlLogix
Enter IP address of the PLC: 192.168.74.10 Connect Connected ●
TIA Portal Project: CIPClient.ap15_1 Browse
Configure Data Block for TIA Project
<table>
<tr><th>Name</th><th>DataType</th><th>Length</th><th>Read</th><th>Write</th></tr>
<tr><td>ReadRecordReq</td><td>Bool</td><td>1</td><td>☐</td><td>☐</td></tr>
<tr><td>ReadRecordReqEdge</td><td>Bool</td><td>1</td><td>☐</td><td>☐</td></tr>
<tr><td>DiagData</td><td>Array[0..1] of SInt</td><td></td><td>☐</td><td>☐</td></tr>
<tr><td>DiagMF</td><td>Array[0..1] of SInt</td><td></td><td>☐</td><td>☐</td></tr>
<tr><td>DiagLength</td><td>DInt</td><td></td><td>☐</td><td>☐</td></tr>
<tr><td>SlotNo</td><td>Int</td><td></td><td>☐</td><td>☐</td></tr>
<tr><td>SubSlot</td><td>Int</td><td></td><td>☐</td><td>☐</td></tr>
<tr><td>BigArray</td><td>Array[0..1] of SInt</td><td>1</td><td>☐</td><td>☐</td></tr>
<tr><td>MovingBits</td><td>Int</td><td>2</td><td>☑</td><td>☐</td></tr>
<tr><td>MoveLeft</td><td>Bool</td><td>1</td><td>☐</td><td>☑</td></tr>
<tr><td>MoveStart</td><td>Bool</td><td>1</td><td></td><td>☑</td></tr>
<tr><td>CartonSize</td><td>DInt</td><td>4</td><td></td><td>☐</td></tr>
<tr><td>ControlWord</td><td>DInt</td><td>4</td><td></td><td>☐</td></tr>
<tr><td>aFloaty</td><td>Real</td><td>4</td><td></td><td>☐</td></tr>
</table>

Acknowledge your selection

Confirmation
Are you sure about the Tag selection ?
 </td><td>The generation of the Datablock will now become available.</td></tr>
<tr><td>8.</td><td>Next click on “Generate”


Select PLC Type: CompactLogix/ControlLogix
Enter IP address of the PLC: 192.168.74.10 Connect Connected ●
TIA Portal Project: CIPClient.ap15_1 Browse
Configure Data Block for TIA Project
<table>
<tr><th>Name</th><th>DataType</th><th>Length</th><th>Read</th><th>Write</th></tr>
<tr><td>ReadRecordReq</td><td>Bool</td><td>1</td><td>☐</td><td>☐</td></tr>
<tr><td>ReadRecordReqEdge</td><td>Bool</td><td>1</td><td>☐</td><td>☐</td></tr>
<tr><td>DiagData</td><td>Array[0..1] of SInt</td><td>1</td><td>☐</td><td>☐</td></tr>
<tr><td>DiagMF</td><td>Array[0..1] of SInt</td><td>1</td><td>☐</td><td>☐</td></tr>
<tr><td>DiagLength</td><td>DInt</td><td>4</td><td>☐</td><td>☐</td></tr>
<tr><td>SlotNo</td><td>Int</td><td>2</td><td>☐</td><td>☐</td></tr>
<tr><td>SubSlot</td><td>Int</td><td>2</td><td>☐</td><td>☐</td></tr>
<tr><td>BigArray</td><td>Array[0..1] of SInt</td><td>1</td><td>☐</td><td>☐</td></tr>
<tr><td>MovingBits</td><td>Int</td><td>2</td><td>☑</td><td>☐</td></tr>
<tr><td>MoveLeft</td><td>Bool</td><td>1</td><td>☐</td><td>☑</td></tr>
<tr><td>MoveStart</td><td>Bool</td><td>1</td><td>☐</td><td>☑</td></tr>
<tr><td>CartonSize</td><td>DInt</td><td>4</td><td></td><td>☐</td></tr>
<tr><td>ControlWord</td><td>DInt</td><td>4</td><td></td><td>☐</td></tr>
<tr><td>aFloaty</td><td>Real</td><td>4</td><td></td><td>☐</td></tr>
</table>

and confirm the operation

Input
Do you want to Generate Data b selected TIA Portal project
 </td><td>You will now have a new or modified data block named “RockwellTags” containing the tag definitions.

An XML file is generated containing the selections made. It can be imported using TIA Portal Openness later.


CIP_Service...
CIP_Service_Tool_GenBlocks
File
Home
Share
View
CIP_Service_Tool_GenBlocks
Quick access
Desktop
Downloads
Documents
Pictures
Name

CipGenDB.xml
RockwellTags.xml</td></tr>
</table>

NOTICE	Make sure the “CIP Service Tool” is closed and has disconnected from the project you used for the import, before opening it with TIA Portal. Otherwise, you are unable to open it.

With the “CIP Service Tool” you can easily create the necessary data block, which contains the process values accessible to you.

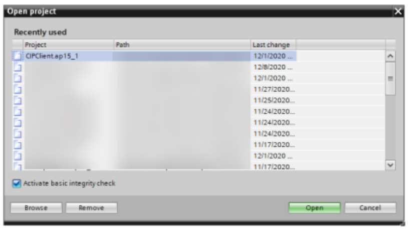
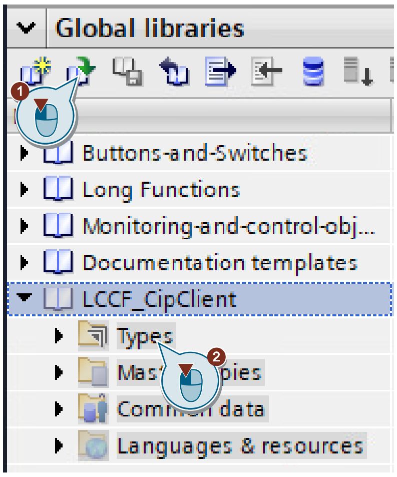
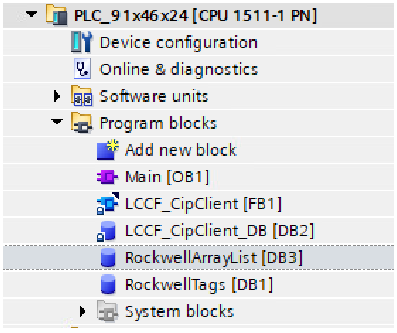
5 Operating

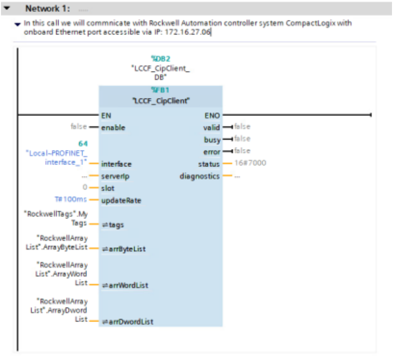
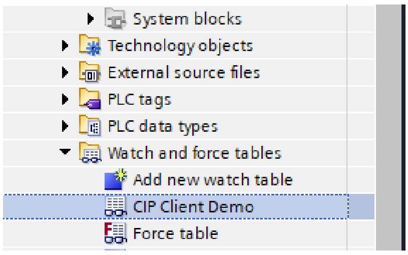
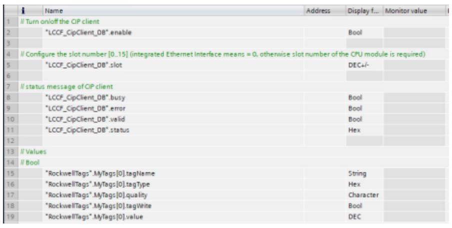
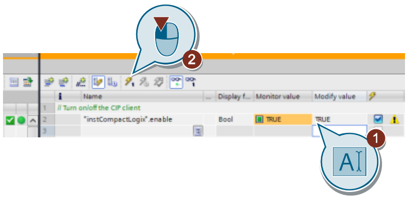
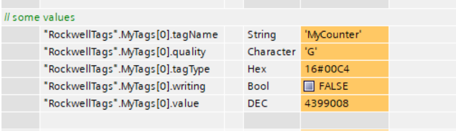
5.1 Start the application

Once the data block with the Rockwell Automation process values is generated this document will show you how the “LCCF_CipClient” block needs to be parameterized to function properly.

Follow the steps listed below to parameterize the “LCCF_CIPClient” block.

Table 5-1

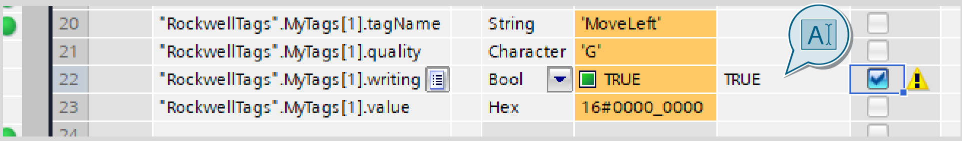
	What to do	Result
1.	open the TIA Portal project of your choice. Preferably this is the same as you imported the data block into.  Open project Recently used Project Path Last change CIPClient.ap15_1 12/1/2020 ... 12/8/2020 ... 12/1/2020 ... 11/27/2020... 11/25/2020... 11/24/2020... 11/24/2020... 11/24/2020... 11/17/2020... 12/1/2020 ... 11/17/2020... ☑ Activate basic integrity check Browse Remove Open Cancel	The project contains already a PLC with the imported data block “RockwellTags”, “RockwellArrayList” in.
2.	open the “LCCF” and  Global libraries 1 Buttons-and-Switches Long Functions Monitoring-and-control-obj... Documentation templates LCCF_CipClient Types 2 Mas[illegible]pies Com[illegible] data Languages & resources import the “LCCF_CipClient” block from the Types  PLC_91x46x24 [CPU 1511-1 PN] Device configuration Online & diagnostics Software units Program blocks Add new block Main [OB1] LCCF_CipClient [FB1] LCCF_CipClient_DB [DB2] RockwellArrayList [DB3] RockwellTags [DB1] System blocks	You now have the “LCCF_CipClient” block in your project.

	What to do	Result
3.	Open the block, where you want to place the call to the "LCCF_CipClient" block.  Network 1: In this call we will commnicate with Rockwell Automation controller system CompactLogix with onboard Ethernet port accessible via IP: 172.16.27.06 %DB2 "LCCF_CipClient_DB" %FB1 "LCCF_CipClient" EN — ENO false — enable — valid — false busy — false error — false 64 "Local~PROFINET_interface_1" — interface — status — 16#7000 ... — serverIp — diagnostics — ... 0 — slot T#100ms — updateRate "RockwellTags".MyTags — ⇄tags "RockwellArrayList".ArrayByteList — ⇄arrByteList "RockwellArrayList".ArrayWordList — ⇄arrWordList "RockwellArrayList".ArrayDwordList — ⇄arrDwordList It is recommended to place it either in the free running program cycle (OB1) or into a cyclic interrupt (OB30) but not in both.	You will be asked for an instance DB to be created. This document uses "LCCF_CipClient_DB" as the instance DB.
4.	Compile and Download to your SIMATIC controller.	
5.	open a new or existing watch table  System blocks Technology objects External source files PLC tags PLC data types Watch and force tables Add new watch table CIP Client Demo Force table	
6.	Populate at least the tags "LCCF_CipClient_DB".enable "LCCF_CipClient_DB".slot	 Name / Address / Display f... / Monitor value 1 // Turn on/off the CIP client 2 "LCCF_CipClient_DB".enable — Bool 3 4 // Configure the slot number [0..15] (integrated Ethernet Interface means = 0, otherwise slot number of the CPU module is required) 5 "LCCF_CipClient_DB".slot — DEC+/- 6 7 // status message of CIP client 8 "LCCF_CipClient_DB".busy — Bool 9 "LCCF_CipClient_DB".error — Bool 10 "LCCF_CipClient_DB".valid — Bool 11 "LCCF_CipClient_DB".status — Hex 12 13 // Values 14 // Bool 15 "RockwellTags".MyTags[0].tagName — String 16 "RockwellTags".MyTags[0].tagType — Hex 17 "RockwellTags".MyTags[0].quality — Character 18 "RockwellTags".MyTags[0].tagWrite — Bool 19 "RockwellTags".MyTags[0].value — DEC
7.	Enable the CIP client block by modifying the value of "LCCF_CipClient_DB".enable = TRUE  Name / Display f... / Monitor value / Modify value 1 // Turn on/off the CIP client 2 "instCompactLogix".enable — Bool — TRUE — TRUE 	You will see the read values from the Rockwell Controller  // some values "RockwellTags".MyTags[0].tagName — String — 'MyCounter' "RockwellTags".MyTags[0].quality — Character — 'G' "RockwellTags".MyTags[0].tagType — Hex — 16#00C4 "RockwellTags".MyTags[0].writing — Bool — ■ FALSE "RockwellTags".MyTags[0].value — DEC — 4399008

NOTE

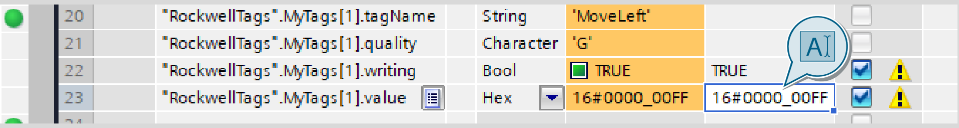
When you want to write a value to the CIP server (Rockwell controller), then toggle the writing(***changed to "tagWrite" in current version***) to 'TRUE'

For this you can either type the word TRUE or a 1 into the marked column



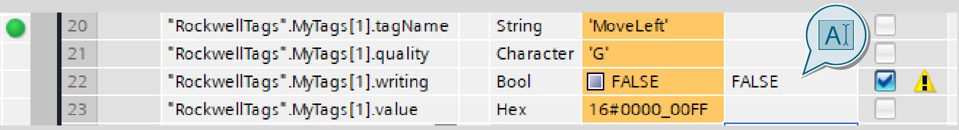
20	"RockwellTags".MyTags[1].tagName	String	'MoveLeft'		☐	
21	"RockwellTags".MyTags[1].quality	Character	'G'		☐	
22	"RockwellTags".MyTags[1].writing	Bool	🟩 TRUE	TRUE	☑	⚠
23	"RockwellTags".MyTags[1].value	Hex	16#0000_0000		☐	

followed by modifying the value



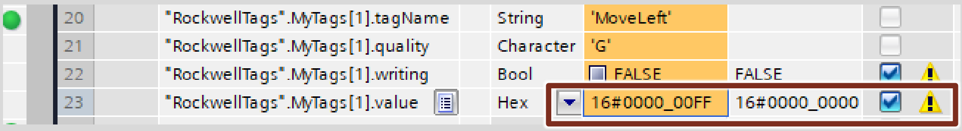
20	"RockwellTags".MyTags[1].tagName	String	'MoveLeft'		☐	
21	"RockwellTags".MyTags[1].quality	Character	'G'		☐	
22	"RockwellTags".MyTags[1].writing	Bool	🟩 TRUE	TRUE	☑	⚠
23	"RockwellTags".MyTags[1].value	Hex	16#0000_00FF	16#0000_00FF	☑	⚠

to finally remove the 'write' command ('writing' = FALSE)



20	"RockwellTags".MyTags[1].tagName	String	'MoveLeft'		☐	
21	"RockwellTags".MyTags[1].quality	Character	'G'		☐	
22	"RockwellTags".MyTags[1].writing	Bool	☐ FALSE	FALSE	☑	⚠
23	"RockwellTags".MyTags[1].value	Hex	16#0000_00FF		☐	

As a test you may overwrite the value to see it updating.



20	"RockwellTags".MyTags[1].tagName	String	'MoveLeft'		☐	
21	"RockwellTags".MyTags[1].quality	Character	'G'		☐	
22	"RockwellTags".MyTags[1].writing	Bool	☐ FALSE	FALSE	☑	⚠
23	"RockwellTags".MyTags[1].value	Hex	16#0000_00FF	16#0000_0000	☑	⚠

5.2 Troubleshooting

In case the result is not as expected the cause could be found on both sides of the communication path.

Before you try to change any of the program or configuration, check the physical installation first.

5.2.1 Physical check

1. Is the SIMATIC powered up?
2. Is the SCALANCE switch powered up if you have used a network switch?
3. Is the SIMATIC Comfort Panel powered up?
4. Are the network cable properly inserted into the LAN sockets?
   This can be determined by evaluating the port LEDs of the devices. At least the Link LED should be illuminated.

Table 5-2: Physical checks

observation	possible cause	remedy
SIMATIC is not reachable from TIA Portal	SIMATIC is not powered up.	• Check power supply and wiring with the installation manual. • Correct wiring • Power the Power Supply up
	SIMATIC does not have network connection	• Check network cable to be inserted properly into the network socket (P1.X1 or P1.X2) • Check and correct network settings of your PC
SIMATIC Comfort Panel is not reachable from TIA Portal	Comfort Panel is not powered up	• Check and correct the power supply to Comfort Panel
	Comfort Panel does not have network connection	• Check and correct the power supply to the network switch • Check network cable to be inserted properly into the network socket. • Check and correct network settings of your PC.
SIMATIC cannot communicate with SIMATIC Comfort Panel	Network switch is not powered up	• Check and correct power supply to the network switch.

If you checked everything and there is no communication at all, then perform the checks recommended in the next chapter.

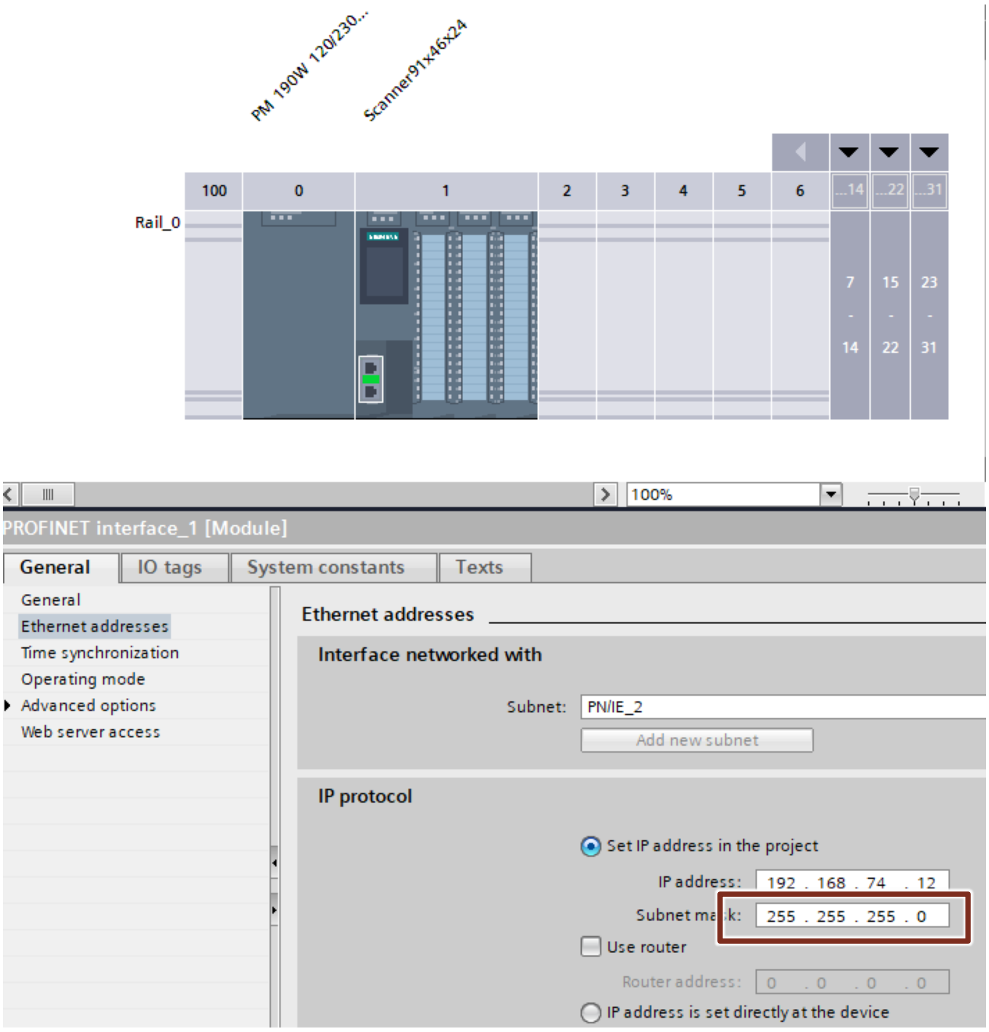
5.2.2 Network settings

Communication issues can be caused also by a misconfiguration of one of the communication partners along the whole path.

Check the Ethernet settings for the communication partners to have

1. The same subnet mask (here: 255.224.0.0)

Figure 5-1: Ethernet settings



PROFINET interface_1 [Module]

Ethernet addresses

Interface networked with

Subnet: PN/IE_2

IP protocol

● Set IP address in the project

IP address: 192 . 168 . 74 . 12

Subnet mask: 255 . 255 . 255 . 0

□ Use router

Router address: 0 . 0 . 0 . 0

○ IP address is set directly at the device

2. Different IP addresses of the same subnet
   For example this application example uses
   192.168.74.12 for the S7-1500
   192.168.74.10 for the Rockwell CompactLogix

5.2.3 SIMATIC Program

Answering the following questions may give you a hint on what needs to be corrected.

Table 5-3: CIP client checks

Observation	Cause	Remedy
status information does not change their values, when enable is set to true	The block is not executed	place an unconditional call to the block in either • cyclic program • cyclic interrupt program
error is true, the moment enable is set to true	Parameterization error	check the status code and correct the parameterization
valid becomes false after a certain time	Connection problems	check the status code and follow the specific recommendations further down in the document.

The CIP client block reports certain error codes to inform the user about issues in the execution. This document describes the status codes the CIP client block reports in the chapter “Block Parameter”.

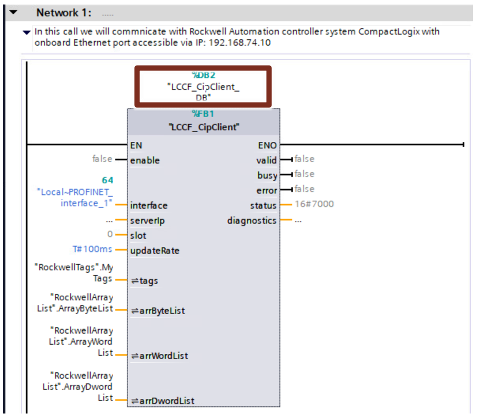
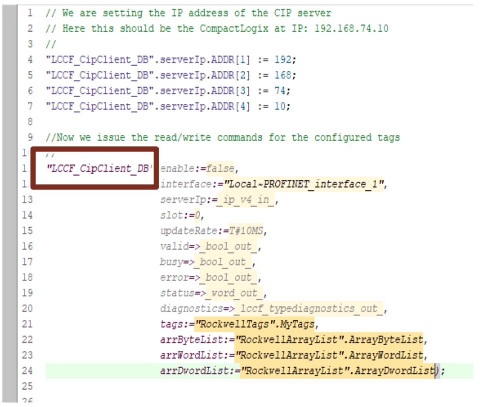
6 LCCF_CipClient block

6.1 Parameters

The CIP client block has been designed to require a minimum of parameters to make its use as easy as possible. Still a minimum external configuration is necessary, which is explained in the following chapter.

A call to the LCCF_CipClient block requires an instance DB to store operation relevant data internally as shown in the below figure.

Table 6-1: Block call to CIP Client

LAD	SCL
	

Network 1: ......

In this call we will commnicate with Rockwell Automation controller system CompactLogix with onboard Ethernet port accessible via IP: 192.168.74.10

```
// We are setting the IP address of the CIP server
// Here this should be the CompactLogix at IP: 192.168.74.10
//
"LCCF_CipClient_DB".serverIp.ADDR[1] := 192;
"LCCF_CipClient_DB".serverIp.ADDR[2] := 168;
"LCCF_CipClient_DB".serverIp.ADDR[3] := 74;
"LCCF_CipClient_DB".serverIp.ADDR[4] := 10;

//Now we issue the read/write commands for the configured tags
//
"LCCF_CipClient_DB' enable:=false,
                    interface:="Local~PROFINET_interface_1",
                    serverIp:=_ip_v4_in_,
                    slot:=0,
                    updateRate:=T#10MS,
                    valid=>_bool_out_,
                    busy=>_bool_out_,
                    error=>_bool_out_,
                    status=>_word_out_,
                    diagnostics=>_lccf_typediagnostics_out_,
                    tags:="RockwellTags".MyTags,
                    arrByteList:="RockwellArrayList".ArrayByteList,
                    arrWordList:="RockwellArrayList".ArrayWordList,
                    arrDwordList:="RockwellArrayList".ArrayDwordList);
```

The instance DB is generated automatically by the TIA Portal, when the call to the block is placed. In here it is named “LCCF_CipClient_DB”

Table 6-2 Parameter of the LCCF_CipClient block

Name	Direction	Data Type	Description
enable	Input	BOOL	Rising edge enables the functionality of the block. Any previously reported fault will be cleared, and conditions re-evaluated. Falling edge shuts the block down and stops any communications.
interface	Input	HW_ANY	Hardware Identifier of the interface to use for the communication. This typically uses a system defined constant. It is possible to use any “Open User Communication” supporting interface. This includes Industrial Ethernet CMs and CPs

Name	Direction	Data Type	Description
serverIP	Input	IP_V4	Identifies the CIP Server to communicate with. The server is identified by its IP v4 address.
slot	Input	USINT	Identifies the slot, where the CIP server is located in. Valid numbers are in the range from 0 to 15. When addressing a Micro800 system, the correct value is 127 Default value is 0
updateTime[1]	Input	Time	Defines the desired update time. The default setting is 10ms.
tags	InOut	Array[*] of “LCCF_typeTagDef”	List of tags to read from or to write to as array with variable length. Each array element is of type “LCCF_typeTagDef”. Refer to chapter 4.2 Creating the tag lists for details about the tag definitions
arrByteList	InOut	Array[*] of "LCCF_typeByteArrayTagDef"	List of array tags values of type Byte to read from or to write to as array with variable length.
arrWordList	InOut	Array[*] of "LCCF_typeWordArrayTagDef"	List of array tags values of type Word to read from or to write to as array with variable length.
arrByteList	InOut	Array[*] of "LCCF_typeByteArrayTagDef"	List of array tags values of type Byte to read from or to write to as array with variable length.
arrWordList	InOut	Array[*] of "LCCF_typeWordArrayTagDef"	List of array tags values of type Word to read from or to write to as array with variable length.
arrByteList	InOut	Array[*] of "LCCF_typeByteArrayTagDef"	List of array tags values of type Byte to read from or to write to as array with variable length.

[1] The parameter “updateTime” may be hidden in the block call

Name	Direction	Data Type	Description
arrWordList	InOut	Array[*] of "LCCF_typeWordArrayTagDef"	List of array tags values of type Word to read from or to write to as array with variable length.
arrByteList	InOut	Array[*] of "LCCF_typeByteArrayTagDef"	List of array tags values of type Byte to read from or to write to as array with variable length.
arrWordList	InOut	Array[*] of "LCCF_typeWordArrayTagDef"	List of array tags values of type Word to read from or to write to as array with variable length.

6.1.1 Block status messages

The LCCF_CipClient block reports a status information to the user, which follows a standardized pattern.

The status code is split into the error flag and a status information value.

Table 6-3: Error and status message format

15	14		12	11			8	7							0
Error	Info/ Warning			Class Code				Specific Status Codes							
16#7 = Information 16#8 = Error				0 = Information											
				2 = Parameter related											
				4 = Internal Cause											
				6 = External Cause											

The CIP client reports specific status codes. They are listed and explained in the following table.

Table 6-4: LCCF_CipClient block status messages

Valid	Busy	Error	Status Code (in hex)	Cause	Remedy
TRUE	TRUE	FALSE	16#0000	Success/ OK	--
FALSE	FALSE	FALSE	16#7000	No Call/ Idle	Block is called with enable = FALSE. Create rising edge on enable to start execution
FALSE	TRUE	FALSE	16#7001	Initial call	Block starts initialization and performs parameter check
TRUE	TRUE	FALSE	16#7002	Follow Up call	Block continues initialization
FALSE	FALSE	TRUE	16#7003	SubfunctionStatus: 16#80C6	Remote partner cannot be reached, Error is cleared once connection

Valid	Busy	Error	Status Code (in hex)	Cause	Remedy
					is restored or enable set to FALSE
TRUE	TRUE	FALSE	16#7601	Warning: Server TimeOut	Nothing to do. Server will reset connection and restart automatically.
FALSE	FALSE	TRUE	16#8201	Invalid Interface specified	Review the Hardware Identifier specified at the Interface parameter. This should be a system managed constant pointing to an interface.
FALSE	FALSE	TRUE	16#8202	Invalid tag list boundaries	The assigned tag list array is invalid, e.g., lower limit is greater than upper limit. Review the tag list assigned variable.
FALSE	FALSE	TRUE	16#8203	Invalid tag definition found	At least one of the tags marked for writing contains an unsupported data type code (tagType).
FALSE	FALSE	TRUE	16#8204	Invalid slot number	The defined slot number has an invalid value. It must be either 0 to 15 or 127 (when addressing a Micro800)
FALSE	FALSE	TRUE	16#8401	Error: Cannot set up server connection	The CIP client block cannot set up a TCP connection. This indicates either an improperly selected interface or the lack of communication resources
FALSE	FALSE	TRUE	16#8402	Error: Cannot disconnect the server connection	The CIP client block fails to reset the server connection. It is recommended to reset the SIMATIC controller.
FALSE	FALSE	TRUE	16#8601	Error: failure during receive of data	The CIP client failed to complete a receive system call. This may indicate a connection break. Reset the server by cycling enable is recommended.
FALSE	FALSE	TRUE	16#8602	Error: failure during send of data	The CIP client failed to complete a send system call. This may indicate a broken connection. Reset the server by cycling the enable in recommended.

Valid	Busy	Error	Status Code (in hex)	Cause	Remedy
FALSE	FALSE	TRUE	16#8603	Error: Unknown ServiceCode received	An unknown service code has been received. This could have been the case, when the communication has been corrupted during transmission. Please inform the developer about this and take notes, when this happened.
FALSE	FALSE	TRUE	16#8600	Error: Undefined state	The CIP server block requested an undefined internal state. This needs to be reported to the developer alongside with the information stored in the diagnostics parameter.

6.1.2 Technical data

For better planning of the automation program the user must be aware, that operating the LCCF_CipClient block has certain impacts on the PLC load.

As all the protocol handling is done in a user program, the cycle time will be extended by the time the selected CPU model needs to place the requests and process the responses. As one could imagine the more often data are requested, the more often these requests will be answered.
Setting the update rates to the lowest acceptable value will reduce the load on the automation program.

Measurements for a SIMATIC S7-1512C show an average load below 1ms.

Table 6-5: Execution times for CIP Client

System	min. load	average load	max. load
S7-1512C	0.1ms	1.0ms	1.5ms
S7-1215C	0.6ms	1.2ms	1.8ms

The measured load was in both cases reading 3 values and writing 1 value.

Besides program execution time memory considerations should also be made, when selecting a CPU model.

For the S7-1200 controller family the technical data are listed in the following table

Table 6-6: Memory Consumption S7-1200

Block	Load Memory	Work Memory
LCCF_CipClient	271.599 Bytes	14.889 Bytes
Instance DB	11.346 Bytes	1.932 Bytes

The execution time of the block is CPU model dependent and was measured on a SIMATIC S7-1215C to be less than 1ms in average.

In the S7-1500 controller family the two required blocks require the memory listed in the following table.

Table 6-7: Memory Consumption S7-1500

Block	Load Memory	Work Memory
LCCF_CipClient	277.508 Bytes	15.200 Bytes
Instance DB	11.398 Bytes	2.020 Bytes

NOTICE

Each defined tag requires data memory additional to what has been listed already. Defining many tags may cause the memory requirements exceed the available memory in your system.

Load Memory	Work Memory
42 Bytes	48 Bytes

The provided information is on a per tag definition base. This means, that this value must be multiplied by the number of tags to get the size of the DB storing the tag definitions.

The CIP client block puts requests to a single server. It is possible to use multiple instances of the block to request values from different servers.

The number of parallel interrogated servers depends on the number of available OUC connection resources in the selected CPU model. For details refer to the technical data of the specific CPU model.

6.2 What is next?

Future extensions will enhance communication capabilities to support further Rockwell Automation control systems such as:

- MicroLogix
- PLC-5

Also, some improvements are planned for later versions in regard of supported data types:

- Structured tags

Further enhancements are planned for the configuration tool as well, which are:

- Delta Creation of tag definition
- Support for multiple target controller

Support for multiple SIMATIC controller per TIA Portal project

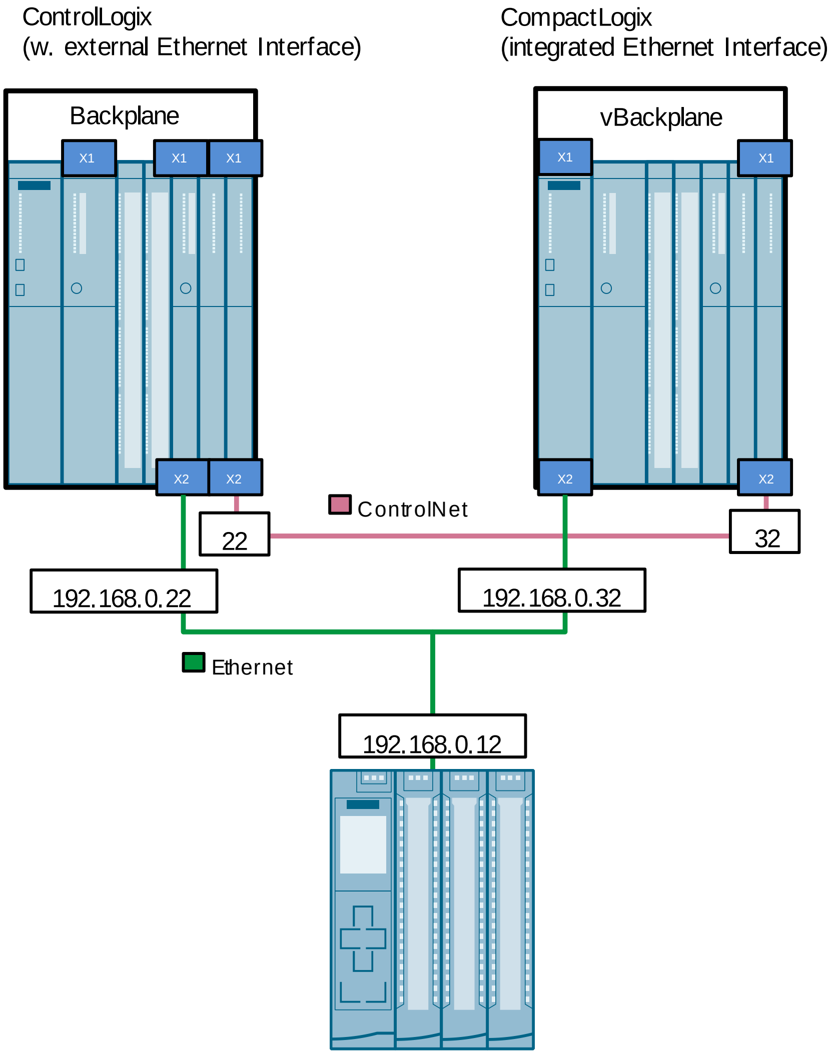
7 Appendix

7.1 CIP/ PCCC Communication path settings

The CIP uses a method of encapsulating the messages adding communication path segments to the command segment. This leads to a longer and longer growing message as the number of routing stations grows.

As an example, the following simple layout is used to explain that fact. Here the S7-1500 is acting as a CIP Client instead the WinCC Adv. RT used throughout the application example. The CIP server is represented by a ControlLogix or CompactLogix system.

Figure 7-1: Schematic communication path setup



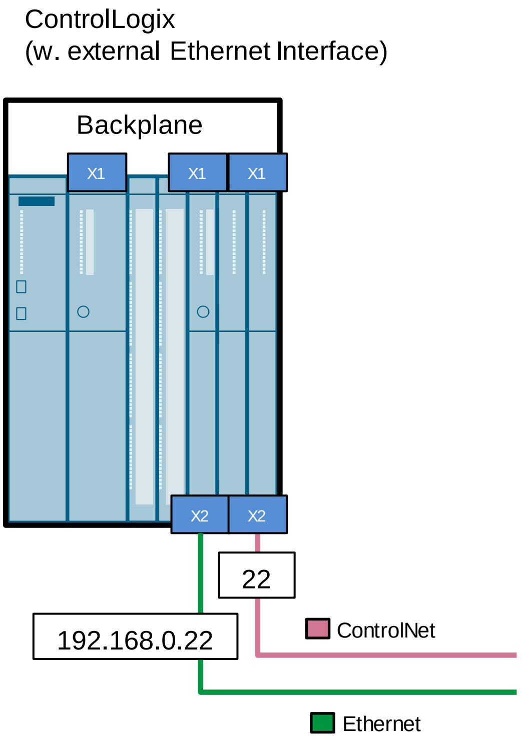
Each CIP message contains the routing information to the target. Every time a bridge accepts such a CIP message it strips out its own address information and forwards the modified CIP message to the next addressee along the way. When there is no further routing information available the destination target has been reached and the CIP message will be interpreted. Once interpreted the response will be send the same way back the request came. This means each bridge keeps a reference to the request package.

7.1.1 Case 1 – Accessing data inside an Allen- Bradley PLC

In this case a ControlLogix uses the ENET bridge in slot 4 (0 based counting) for communicating with the CIP client. The processor module itself is in slot 1. An additional ControlNet Bridge is in slot 6 and is not used in this scenario.

The S7-1500 sends a request to the ControlLogix on 192.168.0.22, slot 1.

Figure 7-2: Access Path ControlLogix with external Ethernet Interface



Therefore, the parameters in the WinCC Panel's driver configuration needs to be:

Table 7-1: WinCC EtherNet/IP driver settings

Parameter Name	Value
IPv4	192.168.0.22
Communication Path	1,1

This communication path is formed as follows:

"192.168.0.22, 1, 1"

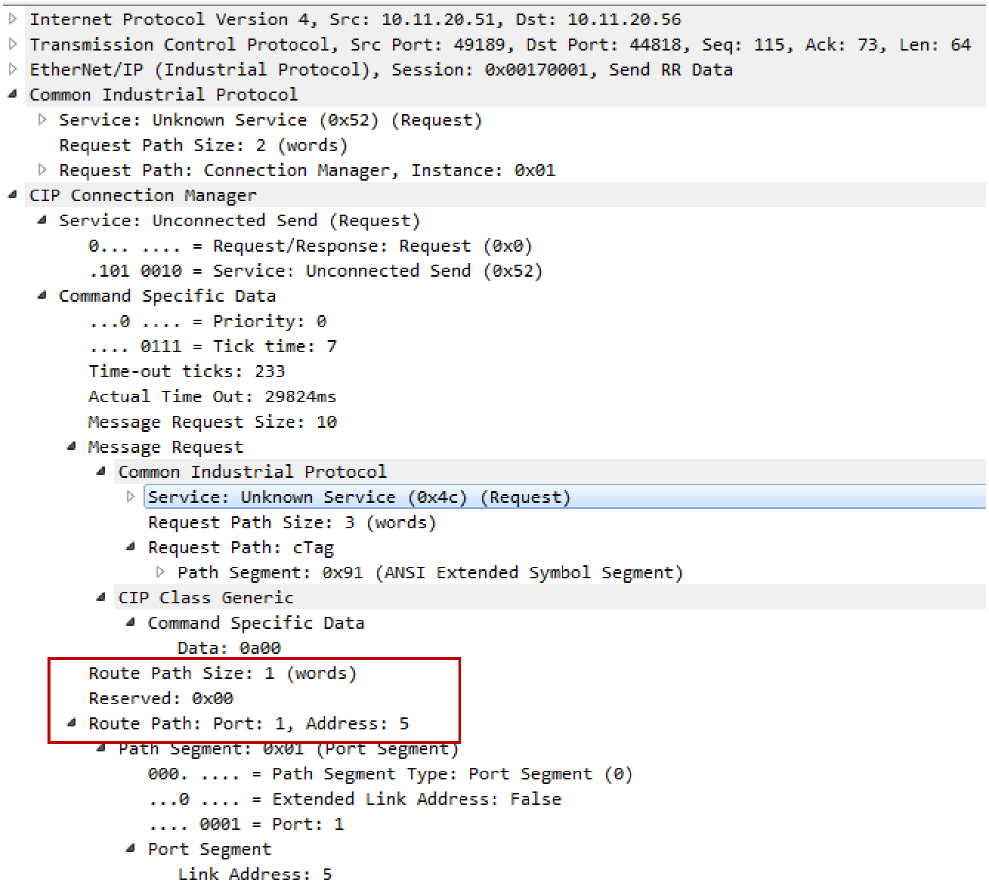
The IPv4 address (**192.168.0.22**) is addressing the X2 interface of the ENET card. With the following part (**1**) in the communication path the interface X1 (more accurate the backplane) is being addressed. This means the CIP message is being forwarded to the backplane after it has removed the IPv4 address from the communication path. The backplane therefore sees a reduced communication path like this

"1, 1"

It recognized that the CIP message was not meant for it and therefore, it removes its address part (**1**) from the CIP message. The remaining CIP message is being forwarded to the processor module as indicated by the remaining address information (1). The message has now a reduced communication path. Only the slot number is contained. The processor removes that information and starts interpreting the remaining message as there is no further routing information in the packet.

The following Wireshark packet analysis shows the additional segment.

Figure 7-3: Access Path encoding



```
Internet Protocol Version 4, Src: 10.11.20.51, Dst: 10.11.20.56
Transmission Control Protocol, Src Port: 49189, Dst Port: 44818, Seq: 115, Ack: 73, Len: 64
EtherNet/IP (Industrial Protocol), Session: 0x00170001, Send RR Data
Common Industrial Protocol
  Service: Unknown Service (0x52) (Request)
  Request Path Size: 2 (words)
  Request Path: Connection Manager, Instance: 0x01
CIP Connection Manager
  Service: Unconnected Send (Request)
    0... .... = Request/Response: Request (0x0)
    .101 0010 = Service: Unconnected Send (0x52)
  Command Specific Data
    ...0 .... = Priority: 0
    .... 0111 = Tick time: 7
    Time-out ticks: 233
    Actual Time Out: 29824ms
    Message Request Size: 10
    Message Request
      Common Industrial Protocol
        Service: Unknown Service (0x4c) (Request)
        Request Path Size: 3 (words)
        Request Path: cTag
          Path Segment: 0x91 (ANSI Extended Symbol Segment)
      CIP Class Generic
        Command Specific Data
          Data: 0a00
    Route Path Size: 1 (words)
    Reserved: 0x00
    Route Path: Port: 1, Address: 5
      Path Segment: 0x01 (Port Segment)
        000. .... = Path Segment Type: Port Segment (0)
        ...0 .... = Extended Link Address: False
        .... 0001 = Port: 1
        Port Segment
          Link Address: 5
```

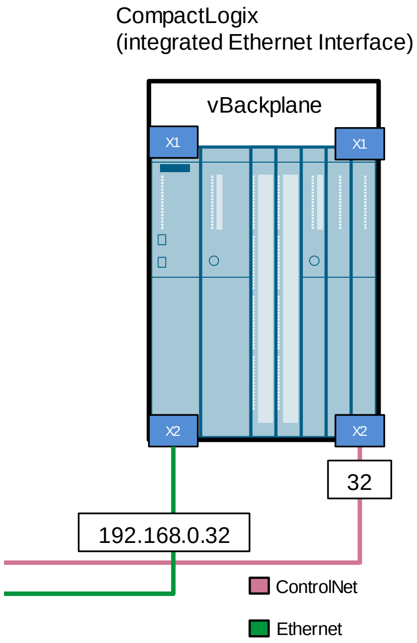
Contrary to the above-described situation, the communication path here shows that the processor module is in slot 5. Here the CIP message follows the same path as above, only difference is the slot number.

7.1.2 Case 2 – Accessing the Allen- Bradley PLC via integrated Ethernet port

In this scenario the CIP client wants to put some data into the CPU module using the integrated Ethernet port. In a CPU module with integrated Ethernet port the Ethernet port is in the same slot as the CPU module.

The CIP client (here the S7-1500) sends a request package to the CPU module at 192.168.0.32 addressing the processor module in slot 0.

Figure 7-4: Access Path CompactLogix with internal Ethernet Interface



Following the previous schema, the communication path would look like this

`“192.168.0.32, 1, 0”`

Following the previous made explanations the CIP message would first travel through the processor module to the backplane to be bounced back to the processor module for interpretation. For such situations it is possible to use an abbreviated communication path only containing the IPv4 address.

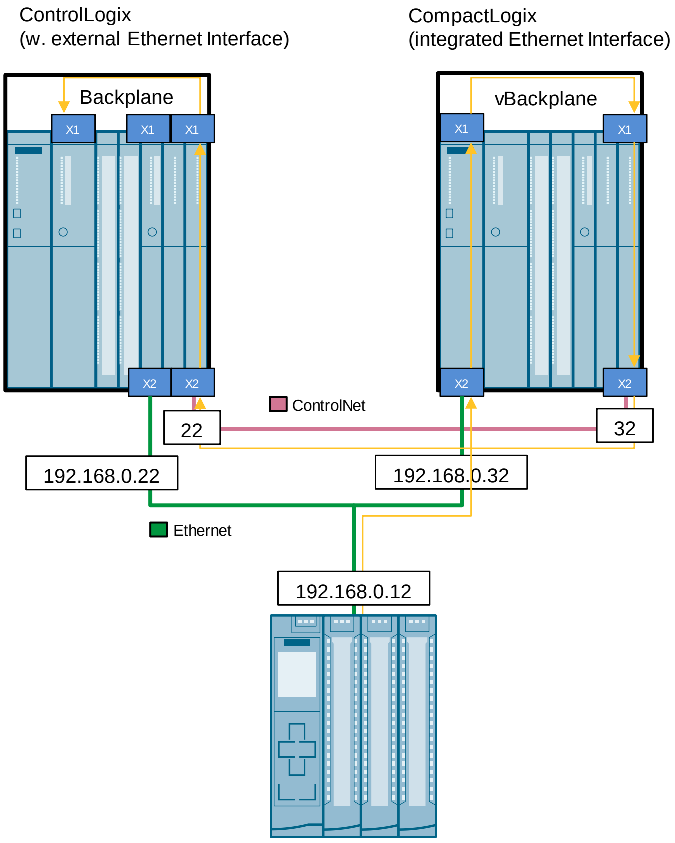
Table 7-2: WinCC EtherNet/IP driver settings

Parameter Name	Value
IPv4	192.168.0.32
Communication Path	1,0

7.1.3 Case 3 – Accessing the Allen- Bradley PLC via the ControlNet interface

In this scenario the CIP client (here S7-1500) wants to read data from the Allen-Bradley PLC. Now the communication path will be through the CompactLogix acting as bridge between EtherNet/IP and ControlNet.

Figure 7-5: Access Path with CompactLogix as bridge



The yellow line in the above picture is represented by the following communication path.

“192.168.0.32, 1, 6, 2, 22, 1, 1”

In the communication path the **red** marked sections always contain a network specific address. The **blue** marked part identifies the backplane or the interface to be used. Here it does not matter, whether this backplane is physically existing or virtual. The **green** part identifies the slot number of the next bridge module (next hop) or the destination module.

Table 7-3: WinCC EtherNet/IP driver settings

Parameter Name	Value
IPv4	192.168.0.32
Communication Path	1,6,2,22,1,1

NOTICE

This scenario is not supported by the LCCF_CipClient block. There is no possibility to define the communication path to such an extend at the LCCF_CipClient block.

7.2 Service and support

Industry Online Support

Do you have any questions or need assistance?

Siemens Industry Online Support offers round the clock access to our entire service and support know-how and portfolio.

The Industry Online Support is the central address for information about our products, solutions, and services.

Product information, manuals, downloads, FAQs, application examples and videos – all information is accessible with just a few mouse clicks:
support.industry.siemens.com

Technical Support

The Technical Support of Siemens Industry provides you fast and competent support regarding all technical queries with numerous tailor-made offers – ranging from basic support to individual support contracts. Please send queries to Technical Support via Web form:
www.siemens.com/industry/supportrequest

SITRAIN – Digital Industry Academy

We support you with our globally available training courses for industry with practical experience, innovative learning methods and a concept that is tailored to the customer’s specific needs.

For more information on our offered trainings and courses, as well as their locations and dates, refer to our web page:
www.siemens.com/sitrain

Service offer

Our range of services includes the following:

- Plant data services
- Spare parts services
- Repair services
- On-site and maintenance services
- Retrofitting and modernization services
- Service programs and contracts

You can find detailed information on our range of services in the service catalogue web page:
support.industry.siemens.com/cs/sc

Industry Online Support app

You will receive optimum support wherever you are with the "Siemens Industry Online Support" app. The app is available for iOS and Android:

support.industry.siemens.com/cs/ww/en/sc/2067

7.3 Related literature

Table 7-4

	Topic
\1\	Siemens Industry Online Support https://support.industry.siemens.com
\2\	Download page of this entry https://support.industry.siemens.com/cs/ww/en/view/109782317
\3\	

7.4 Change documentation

Table 7-5

Version	Date	Modifications
V1.0.0	01/2021	First version
V1.0.1	05/2022	Added Information about Micro800 communications
V1.0.2	12/2023	Added Information about New Interface for Array Tags