Adroit Technologies Protocol Driver

Name	Allen Bradley Ethernet IP
DLL	ABTCP

COPYRIGHT

Advanced Worx 112 (Pty) Ltd
20 Waterford Office Park
189 Witkoppen Road (c/o Waterford Drive)
Fourways
South Africa

https://adroittech.co.za/

Contents

1. Overview

ABTCP supports communication with the following PLCs:

- ControlLogix 5550 with 1756-ENET module
- CompactLogix (tested on L35E)
- SoftLogix5
- MicroLogix 1000/1200/1500 via channel 0 with 1761-NET-ENI Module
- MicroLogix 1100 using Ethernet IP/CIP.
- ControlLogix Gateway to PLC5 via DH+
- ControlLogix Gateway to SLC5/04 via DH+
- PLC5/20E
- PLC5/40E
- PLC5/80E
- PLC5 enhanced processors with 1785-ENET (Ethernet Side Car)
- SLC5/05
- SLC500 via channel 0 with 1761-NET-ENI Module
- PLC5 processors via channel 0 with 1761-NET-ENI Module

This driver also supports the X8 PLC range from RS OEMax.

2. Wiring

Standard Ethernet interconnection (10BASE-T/E), 100-BASE E.

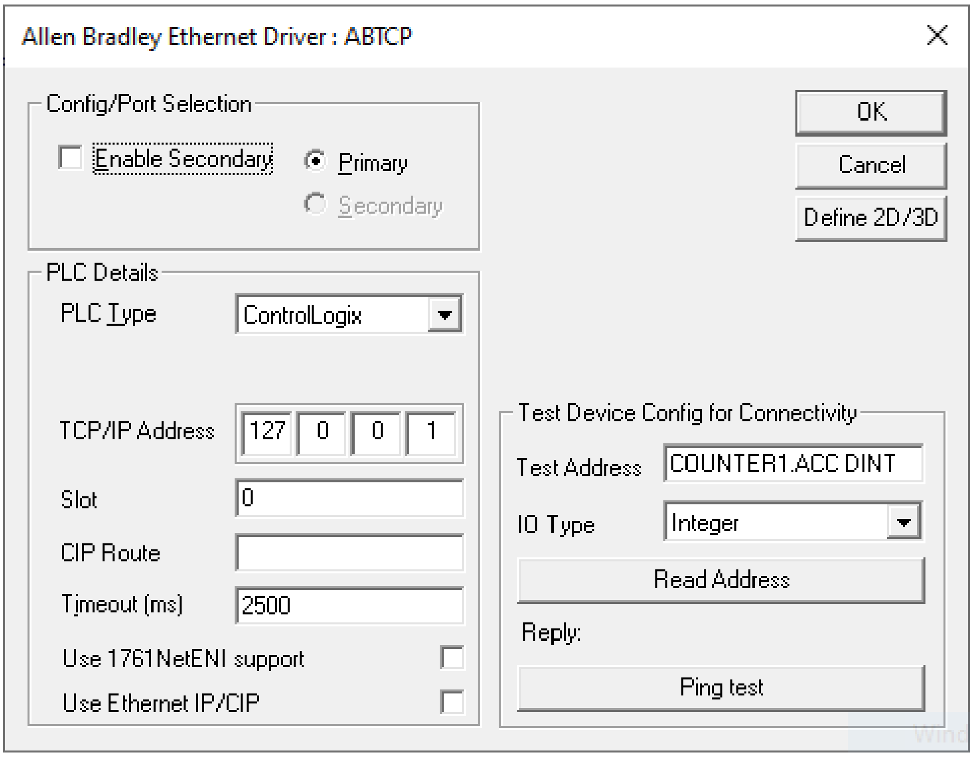
Please note this driver requires an additional OCX file and is installed automatically.

PLEASE NOTE

Asabtcp.ocx rev 2.10.1.0 required for revision 21 onwards.

Manual installation procedure:
Make sure the asabtcp.ocx is copied to the adroit root directory.
Run “regsvr32 c:\adroit\asabtcp.ocx” at the command prompt.
Run Setup and configure driver as normal.

3. Device Configuration



Allen Bradley Ethernet Driver : ABTCP

Config/Port Selection

□ Enable Secondary ● Primary ○ Secondary

PLC Details

PLC Type: ControlLogix

TCP/IP Address: 127 . 0 . 0 . 1

Slot: 0

CIP Route:

Timeout (ms): 2500

Use 1761NetENI support □

Use Ethernet IP/CIP □

Test Device Config for Connectivity

Test Address: COUNTER1.ACC DINT

IO Type: Integer

Reply:

3.1.1. Config/Port Selection

Enable Secondary – Enables secondary configuration for dual redundant PLCs via the primary and secondary radio buttons. Do not enable if a secondary channel is not configured correctly.

3.1.2. PLC Details

PLC Type - Applicable Allen Bradley PLC types

PLC Type
PLC5
SLC500/MicroLogix
ControlLogix
X8-Series

Model – Only applicable to PLC series PLCs. Used to size I/O ranges etc.

TCP/IP Address – TCP/IP network address of the remote PLC.

Slot – (Optional) Refers to the ControlLogix slot where the target CPU resides.

CIP route – ControlLogix Gateway CIP Routing Path. See Protocol Definition and Considerations for details.

Timeout - The time, in milliseconds, that Adroit will wait for a valid response from a PLC before failing a transaction (Read or Write).

Use 1761NetENI support - If checked, it will enable the driver to use 1761NetENI for communication.

Use Ethernet IP/CIP - If checked, it will enable the driver to use Ethernet IP/CIP for communication (Especially Required for MicroLogix 1100).

3.1.3. Test Device Config for Connectivity

This section is available to help users determine whether the settings they have entered are valid and Adroit can communicate with your device.

Test Address – An address in your device which will be polled when the "Read Address" button is clicked. The address entered must be valid.

IO Type – The address type being read (i.e., Boolean, integer, real or string).

Read Address – Click this button to poll your device with the current settings in your dialog.

Reply – When "Read Address" is clicked, Adroit will poll the device. The value returned from your device will be displayed here.

Ping Test – Click to verify that the TCP/IP address entered into the boxes is accessible.

Buttons

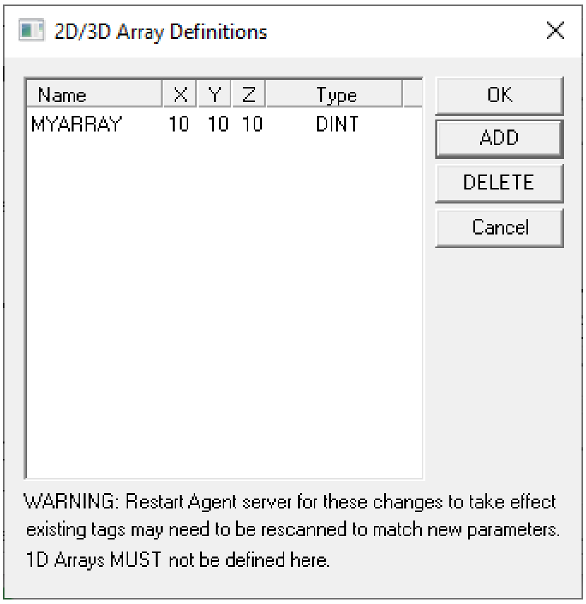
Define 2D/3D – Click to setup 2D and 3D array definitions. These array definitions are used to configure array maximum size(s) for each array dimension(s) (x, y, z). The driver uses these to verify correct array sizes and to assist the driver in calculating optimal scan task sizing.

INFORMATION

Since one dimensional (1D) arrays do not need to be defined in this way, they cannot be defined here.

PLEASE NOTE

The Agent Server must be restarted if any changes are made here.



2D/3D Array Definitions

Name	X	Y	Z	Type
MYARRAY	10	10	10	DINT

WARNING: Restart Agent server for these changes to take effect existing tags may need to be rescanned to match new parameters. 1D Arrays MUST not be defined here.

OK – To accept any changes (Restart of agent server is REQUIRED).

ADD – Create a new definition or modify an existing definition.

PLEASE NOTE

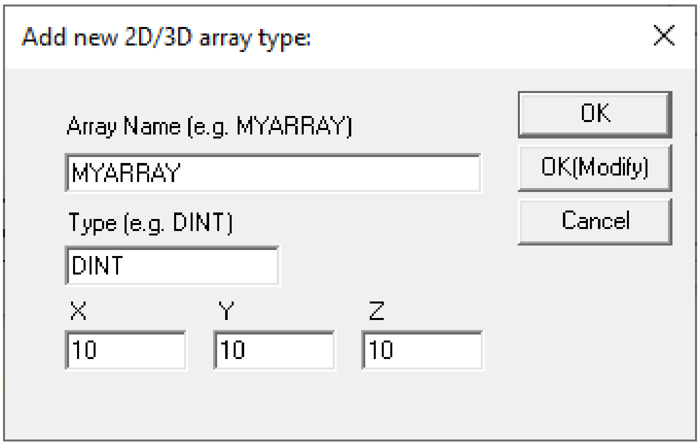
The Agent Server must be restarted if any changes are made here.

DELETE – Remove a selected definition.

PLEASE NOTE

The Agent Server must be restarted if any changes are made here.

Cancel – Cancel all changes.



Add new 2D/3D array type:

Array Name (e.g. MYARRAY): MYARRAY

Type (e.g. DINT): DINT

X: 10
Y: 10
Z: 10

Array Name – Must be unique i.e., the name used when scanning the array in the driver address is NOT the name of the tag in Logix. In other words, this name is ONLY used in the driver as a representation of the array properties.
E.g. If array of type INT is called PUMP01[0,0,0] to [1,2,2], name could be "1BY2BY2INT" this is used in the address as a postfix in the following way: "PUMP[1|1|0] 1BY2BY2INT"

Type – The register type in the PLC (e.g. DINT, BOOL, REAL, etc.)
(DO NOT define STRING in 2D or 3D)

X, Y, Z – Each dimension size is inserted in each entry field. Use 0 to disable it.
Please Note: only Z can be zero.
E.g., If dimensions elements are [0,0] to [1,3] then X must be 2 and Y must be 4 and Z must be 0.

OK – To accept change. Duplicate error if modify was used, if this is required use "OK(Modify)"

PLEASE NOTE

The Agent Server must be restarted if any changes are made here.

Ok(Modify) – To accept modifications to a definition (Restart of agent server is REQUIRED).

Cancel – Cancel all changes.

4. Supported Addresses

4.1. Control Logix

WARNING

It is HIGHLY recommended that new projects be commissioned using arrays wherever possible because arrays are retrieved MUCH faster due to many more tags per transaction.

If individual tags are used the driver's turnaround times will be extremely excessive by Ethernet standards, because each tag is retrieved in a separate transaction.
Mixed tag objects known as UDT (User Defined Types) are supported but must be fully qualified and are thus considered tags. See example below.

PLEASE NOTE

It is not recommended to use UDT's because of the speed implication as described above. This does not include TIMERS and COUNTERS (See below).

Accessing non array tags:
<tagname> <rawType>
E.g. "MYINT INT"

Supported raw types:
SINT – Short Integer (8bit)
INT – Integer type (16Bit).
DINT – Double Integer type (32Bit).
REAL – Floating real type;
BOOL – Boolean/digital type

Special:
CT - Counter (See comment below)
TM - Timer
Counters and Timers are special UDT types, where the ACCUMULATOR (ACC) is used by default if not fully qualified. To access say the PRESET value, qualify the tag fully such as, for example, "MYCOUNTER.PRE DINT". Make sure the rawtype is correct else unpacking may fail or result in incorrect values.

PLEASE NOTE

UDTs, other than CT and TM, must be fully qualified down to their constituent subtypes. See example below.

String support:
STRING – standard Logix5550 82-byte string.
STRINGxx – where xx is anything between 1 and 79.
e.g., STRING16 represents a 16 character array.

Please refer section 5.3egarding additional information and warnings especially as far as the use of user defined strings.
Strings can also be defined as arrays where they will be retrieved in groups of 5.
e.g. MY_STRING_ARRAY[0] STRING.
e.g. MY_STRING_ARRAY12[3] STRING12.

WARNING

Strings only supported in 1D arrays. Do not define STRING or STRINGxx in 2D or 3D.

ControlLogix supports two different scopes for tags: Controller and Program.

Let's say in the program scope one has define MYPUMP as a DINT (with value 123) and in the controller scope one has define MYPUMP also as a DINT (with value 456).

Program tags are defined within the scope of a program. To refer to a program tag, you must include the prefix "program:" and the program name when scanning the tag.

For program tags: Addresses use the following syntax:
program:<program name>.<tag name>
e.g. "PROGRAM:PROGRAM1.MYPUMP DINT" returns the value 123

Controller tags are defined in the scope of the controller and do not require you to prefix the tag name with any additional information. To refer to a controller tag, simply specify the tag name for the Address.

e.g. "MYPUMP DINT" return the value 456
To get a tag inside a UDT the address must be fully qualified:

```
e.g.  MYUDT consists of {    VOLTS REAL,
                             EARTHED BOOL,
                             OPEN BOOL
                             }
```

To get the open status the following address will have to be scanned to a digital agent:
"MYUDT.OPEN BOOL"

Notice the subtype above is specified and not the UDT type itself.
Please note the warning about UDT scans being slow.
If it were a program tag:
"PROGRAM1.MYUDT.OPEN BOOL"

Arrays in general :
Because scan task sizes must be known and calculable, only 2D and 3D tags must be predefined. 1D arrays are not predefined sizing on these is determined during scanning.
If defining a 2D or 3D array you should merely copy the structure that was defined in the Logix PLC itself via the programming software.

Accessing 1D array tags:
<arraytagname>[x] <rawtype>

To scan the second element of a 1D array of items say BOOLS use the following:
"BOOLARRAY[2] BOOL"
It is not necessary to scan all items. They can be scanned into Adroit in any order or not scanned at all.

Accessing 2D/3D array tags:
<arraytagname>[x|y] <typename>
<arraytagname>[x|y|z] <typename>

To scan the item in the third row, fifth column of a 2D array of items, say of type DINTS, use the following:
"DINTARRAY[2|4] MY2DARRAY"
Remember 0 is a valid item (first element).

PLEASE NOTE

This scan will only succeed if MY2DARRAY was configured in setup definitions as MY2DARRAY at least 3 for x and at least 5 for y. The dimension for Z may be 0, and the TYPE should be DINT, however INT might also pass but the read will not work if the array is a DINT.

It is not necessary to scan all items. They can be scanned into Adroit in any order or not at all.

4.2. PLC5

Device		Min Range	Max Range	PLC5 Data Type	Boolean	Int	Real
Integer File	**N**	0:0	998:998	Bool Int16 Int32 Dword32	N3:0/0 - - -	- N3:0 N3:0 L N3:0 D	- N3:0 N3:0 L N3:0 D
Floating Point	**F**	0:0	998:998	Float32	-	-	F3:0 F
Binary	**B**	0:0/0	998:998/15	Bool Int16	B9:0/15 -	- B9:0	- B9:0
Inputs	**I**	000/0	037/17 * 177/17 277/17	Bool Int16	I:000/17 -	- I:000	- O:000
Outputs	**O**	000/0	037/17 * 177/17 277/17	Bool	O:000/17 O:000/17 O:000/17	- O:000 O:000	- O:000 O:000
Timer	**T**	0:0	998:998	Int16	-	T3:0	-
Counter	**C**	0:0	998:998	Int16	-	C3:0	-
Status	**S**	0/0	127/15	Status Int16	S:0/15 -	- S:0	- S:0

***** Size Model dependant and Inputs and Outputs are OCTAL

4.3. SLC 500

Device		Min Range	Max Range	PLC5 Data Type	Boolean	Int	Real
Integer File	**N**	0:0	998:998	Int16 Int32 Dword32	N3:0/0 - -	N3:0 N3:0 L N3:0 D	N3:0 N3:0 L N3:0 D
Floating Point	**F**	0:0 F	998:998	Float32	-	-	F3:0 F
Binary	**B**	0:0/0	998:998/15	Bool Int16	B9:0/15 -	- B9:0	- B9:0
Inputs	**I**	0.0/0	30.255/15 *	Bool Int16	I:0.0/15 -	- I:0	- O:0
Outputs	**O**	0.0/0	30.255/15 *	Bool Int16	O:0.0/15 * -	- O:0.0	- O:0.0
Timer	**T**	0:0	999:999	Int16	-	T3:0	-
Counter	**C**	0:0	999:999	Int16	-	C3:0	-
Status	**S**	0/0	127/15	Status Int16	S:0/15 * -	- S:0	- S:0

* Size Model dependant and Inputs and Outputs are DECIMAL

4.4. X8-Series

Device		Min Range	Max Range	PLC5 Data Type	Boolean	Int	Real
Integer File	**N**	0:0	998:998	Int16 Int32 Dword32	N3:0/0 - -	N3:0 N3:0 L N3:0 D	N3:0 N3:0 L N3:0 D
Long File	**L**	0:0 L	998:998	Long32	-	L3:0 L	L3:0 L
Floating Point	**F**	0:0 F	998:998	Float32	-	-	F3:0 F
Binary	**B**	0:0/0	998:998.15	Bool Int16	B9:0/3 -	- B9:0	- B9:0
Inputs (X1)	**X**	0.0/0	30.255.15 *	Bool Int16	X1:0.0/0 -	- X1:0.0	- X1:0.0
Outputs (Y0)	**Y**	0.0/0	30.255.15 *	Bool Int16	Y0:0.0/15 * -	- Y0:0.0	- Y0:0.0
Timer	**T**	0:0	999:999	Int16	-	T3:0	-
Counter	**C**	0:0	999:999	Int16	-	C3:0	-
Status	**S**	0:0	127/15	Status Int16	S:0/15 * -	- S:0	- S:0

* Size Model dependant and Inputs and Outputs are DECIMAL

5. Protocol Definition and Considerations

5.1. CIP Gateway Considerations

CIP route – ControlLogix Gateway CIP Routing Path.

The driver supports ControlLogix Gateway to PLC5 and SLC5/04 via Data Highway Plus using the 1756-DHRIO Module.

The gateway operation requires a CIP Routing Path, which indicates to the gateway where to route incoming and outgoing messages.

5.1.1. ControlLogix Gateway CIP Routing Path Description:

To specify a complete CIP routing path the driver needs to know:

Message path from 1756-ENET module.
Backplane slot of the 1756-DHRIO module.
1756-DHRIO Channel to use (A or B).
Data Highway Plus node address of the target PLC.

5.1.2. Format

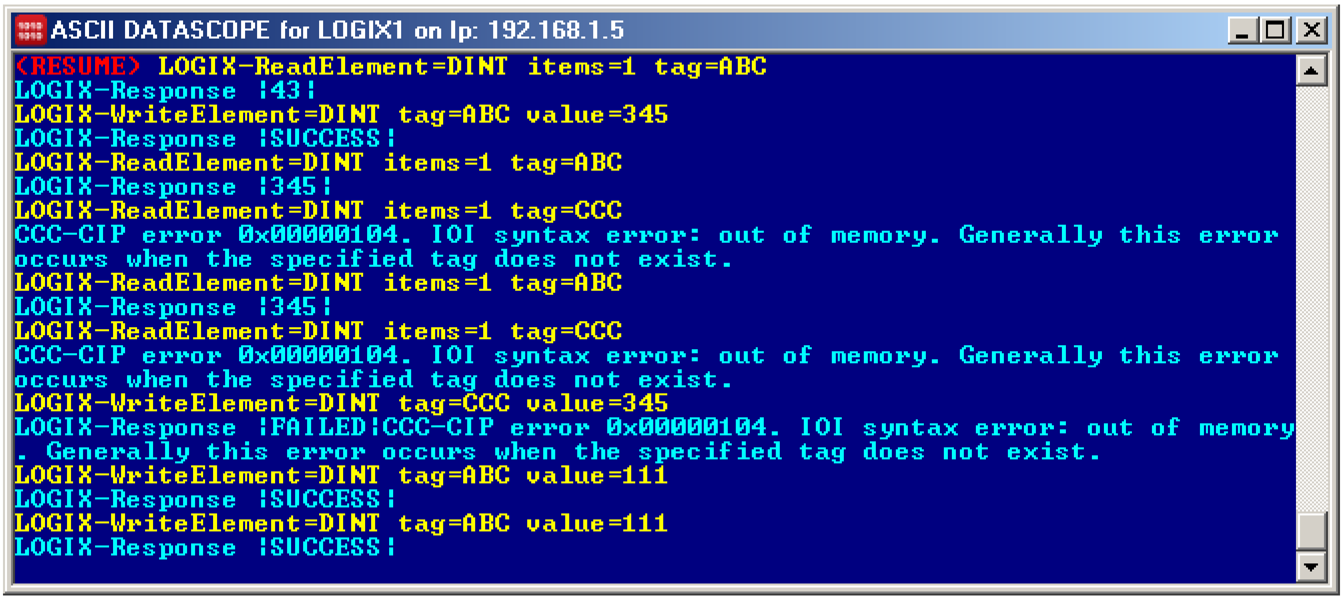
The format for the CIP Routing Path is as follows:
DH:A.B.C.D
where
DH = Data Highway Plus bridging
A = Backplane Number (always 1)
B = Slot number of local 1756-DHRIO (decimal)
C = Channel of local 1756-DHRIO (A or B)
D = Node address of remote device (octal for DH)

5.1.3. Examples

This example shows the routing path to access two DH+ nodes on two Data Highway Plus networks:
Route path “DH:1.2.A.7” will access the PLC at DH+ node 7 (octal) using channel A of the 1756-DHRIO module located in slot 2 of the ControlLogix backplane.

Route path “DH:1.2.B.13” will access the PLC at DH+ node 13 (octal) using channel B of the 1756-DHRIO module located in slot 2 of the ControlLogix backplane.

5.2. Datascope Example



ASCII DATASCOPE for LOGIX1 on Ip: 192.168.1.5

```
<RESUME> LOGIX-ReadElement=DINT items=1 tag=ABC
LOGIX-Response ¦43¦
LOGIX-WriteElement=DINT tag=ABC value=345
LOGIX-Response ¦SUCCESS¦
LOGIX-ReadElement=DINT items=1 tag=ABC
LOGIX-Response ¦345¦
LOGIX-ReadElement=DINT items=1 tag=CCC
CCC-CIP error 0x00000104. IOI syntax error: out of memory. Generally this error
occurs when the specified tag does not exist.
LOGIX-ReadElement=DINT items=1 tag=ABC
LOGIX-Response ¦345¦
LOGIX-ReadElement=DINT items=1 tag=CCC
CCC-CIP error 0x00000104. IOI syntax error: out of memory. Generally this error
occurs when the specified tag does not exist.
LOGIX-WriteElement=DINT tag=CCC value=345
LOGIX-Response ¦FAILED¦CCC-CIP error 0x00000104. IOI syntax error: out of memory
. Generally this error occurs when the specified tag does not exist.
LOGIX-WriteElement=DINT tag=ABC value=111
LOGIX-Response ¦SUCCESS¦
LOGIX-WriteElement=DINT tag=ABC value=111
LOGIX-Response ¦SUCCESS¦
```

5.3. Logix String Support

WARNING

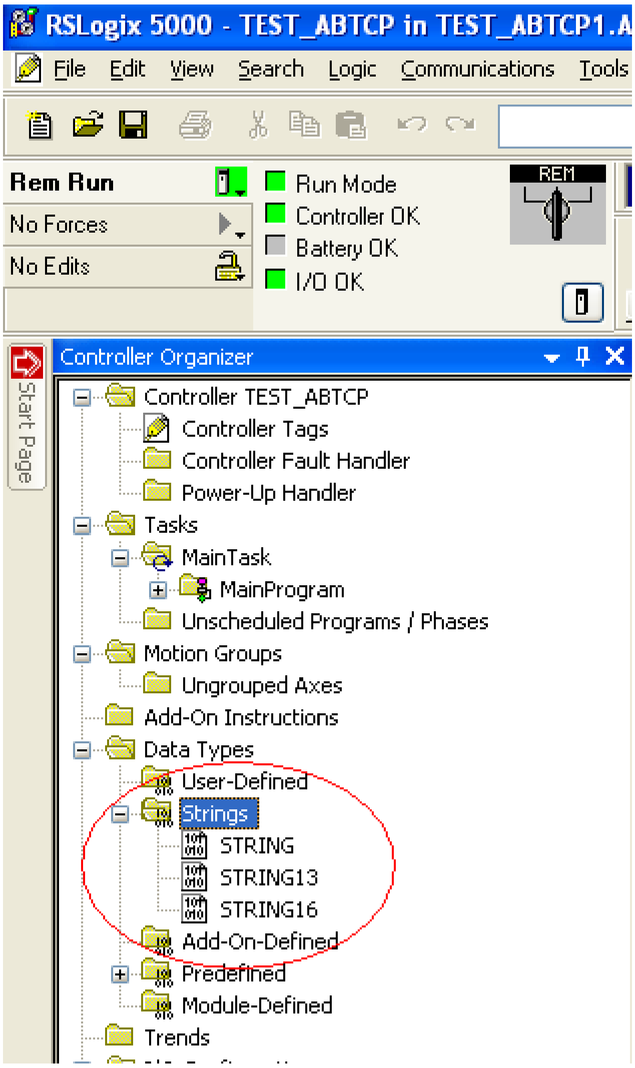
When using User defined strings, it is very important to ensure that the definition in Adroit and in the PLC match. If a tag scanned with the incorrect length is shorter in Adroit, only the shorter length will be available in reads and writes due to truncation, and if longer in Adroit, then writes will be rejected by the PLC (no truncation will be attempted) but reads will be truncated.

The Adroit STRING agent maximum size is 80 characters. Any attempt to validate a STRING81 or greater will be truncated back to STRING80 and the various mismatches described above will most likely result.

WARNING

Strings only supported in 1D arrays DO NOT define STRING or STRINGxx in 2D or 3D.

User defined strings are defined in RSLogix5000 by right clicking on the strings data type as seen in the diagram below.


RSLogix 5000 - TEST_ABTCP in TEST_ABTCP1.A
File Edit View Search Logic Communications Tools
Rem Run
No Forces
No Edits
Run Mode
Controller OK
Battery OK
I/O OK
REM
Start Page
Controller Organizer
Controller TEST_ABTCP
Controller Tags
Controller Fault Handler
Power-Up Handler
Tasks
MainTask
MainProgram
Unscheduled Programs / Phases
Motion Groups
Ungrouped Axes
Add-On Instructions
Data Types
User-Defined
Strings
STRING
STRING13
STRING16
Add-On-Defined
Predefined
Module-Defined
Trends

6. General Notes and Troubleshooting

6.1. Contact Details

Email: support@adroit.co.za and drivers@adroit.co.za

Phone: +27 11 658-8100

Web: www.adroit.co.za

6.2. Adroit Technologies Knowledge Base

For additional information, please refer to our Knowledgebase website:

Latest Drivers topics - Features, discussions, tips, tricks, questions, problems and feedback (adroittechnologiesautomation.com)

6.3. Troubleshooting Guide

“Unknown Error” is returned in the Datascope whenever the driver tries to poll a device. (Seen
1. using MicroLogix 1500 but applies to all PLC types)

One possible cause of this state is there is a problem OCX file and or its registration process.

Make sure the asabtcp.ocx is revision 2.6.1.1

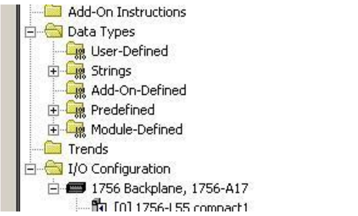
Make sure the OCX is registered.

Try the following manually:

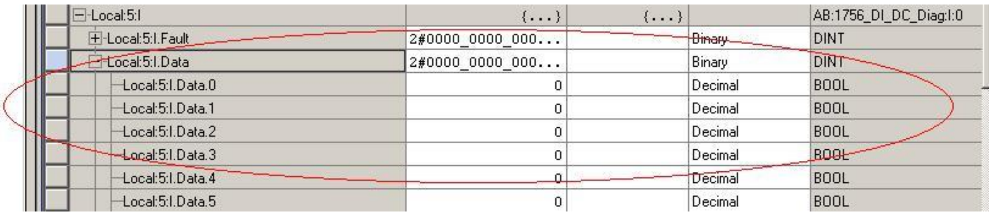
A. Open a command prompt, change directory to c:\adroit
B. Uninstall the OCX by typing “regsvr32 /u asabtcp.ocx” at the prompt
2. C. Reregister it again via “regsvr32 asabtcp.ocx” at the prompt.
D. Restart the agent server and test if this resolves the “unknown error”.

Destination unknown, class unsupported error and the importance of scanning the correct type in Logix PLCs.

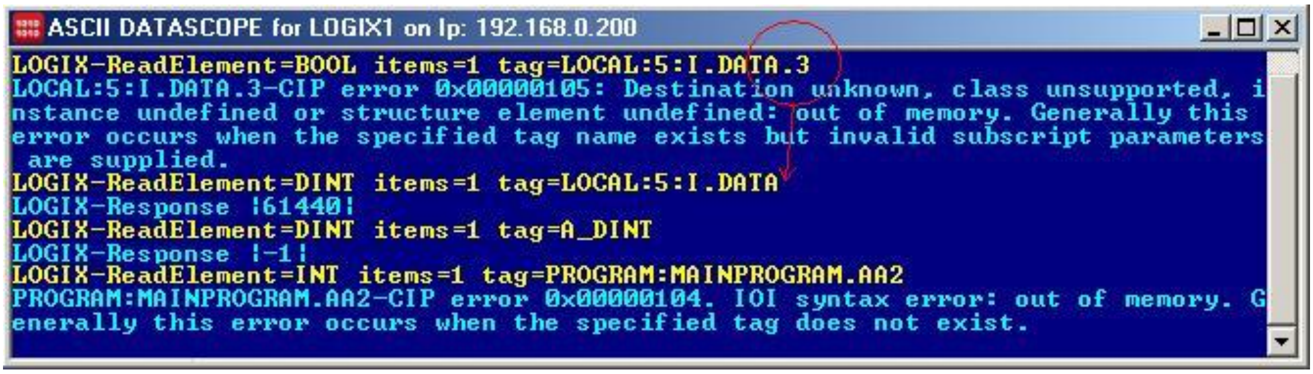
RSLogix5000 is able to drill down to bit level on almost any datatype in this case a DINT (Local:5:I.Data) which is a digital card. If an attempt is made to scan these as bits the device reports a Destination unknown, class unsupported error , this is due to the fact that the data is in fact NOT Boolean, so this must be scanned as an INTEGER type , use an AREC agent to get access to 32 bits, or a MARSHAL/MULTISTATE to get access to 16-bit types.



- Add-On Instructions
- Data Types
  - User-Defined
  - Strings
  - Add-On-Defined
  - Predefined
  - Module-Defined
- Trends
- I/O Configuration
  - 1756 Backplane, 1756-A17
    - [0] 1756-L55 compact1



Local:5:I	{...}	{...}		AB:1756_DI_DC_Diag:I:0
Local:5:I.Fault	2#0000_0000_000...		Binary	DINT
Local:5:I.Data	2#0000_0000_000...		Binary	DINT
Local:5:I.Data.0	0		Decimal	BOOL
Local:5:I.Data.1	0		Decimal	BOOL
Local:5:I.Data.2	0		Decimal	BOOL
Local:5:I.Data.3	0		Decimal	BOOL
Local:5:I.Data.4	0		Decimal	BOOL
Local:5:I.Data.5	0		Decimal	BOOL



```
LOGIX-ReadElement=BOOL items=1 tag=LOCAL:5:I.DATA.3
LOCAL:5:I.DATA.3-CIP error 0x00000105: Destination unknown, class unsupported, i
nstance undefined or structure element undefined: out of memory. Generally this
error occurs when the specified tag name exists but invalid subscript parameters
 are supplied.
LOGIX-ReadElement=DINT items=1 tag=LOCAL:5:I.DATA
LOGIX-Response ¦61440¦
LOGIX-ReadElement=DINT items=1 tag=A_DINT
LOGIX-Response ¦-1¦
LOGIX-ReadElement=INT items=1 tag=PROGRAM:MAINPROGRAM.AA2
PROGRAM:MAINPROGRAM.AA2-CIP error 0x00000104. IOI syntax error: out of memory. G
enerally this error occurs when the specified tag does not exist.
```

7. Revisions

7.1. Document Revision

Rev.	Date	Author	Comment	Approval
1.0	14-Dec-2022	J. Duncan	New Format	D. Jordaan
1.1	09-Jan-2022	K. Robinson	Grammar and wording changes	D. Jordaan
2.0				
3.0				
4.0				
5.0				
6.0				

7.2. DLL Revision

Rev.	Date	Changes
2	3/08/2005	First release , LOGIX mode only , PLC5 SLC500 disabled for now!
3	3/02/2006	Modifications documented; driver is in revision 7.
4	29/06/2006	Modifications documented; driver is in revision 8.
10	15/08/2006	Added better addressing support for PLC5 and SLC500 series PLCs. (Further Additions are possible)
21	22/12/2008	Added better compound read/write support, but bad read messages are still displayed on the Datascope. Fixed Boolean reads. Also rev 2.1.10.0 of the asabtcp.ocx is required for this change.
22	23/06/2009	Added L Register Readings for SLC500/Micrologic PLC Types
23	03/08/2009	Fixed L and F Items count calculation in Read when calling ReadPLC. (not serious but read too much data)
24	05/10/2009	Mixed use of registry access objects (driverlib stuff) would cause the Logix array types definition list to blank out when +-10 arrays were configured.
24a	15/10/2009	Driver excepted if multipleTagReads was 0.

25	22/10/2009	IMPORTANT FIX : Fixed a crash caused by unusually long error messages coming in from the API.
41	14/06/2013	Test connection added to dialog
59	01/05/2019	Added X8-Series PLC model, based on SLC500 range modified to match X and Y io type definitions.