

APPLICATION NOTE

THIS INFORMATION PROVIDED BY AUTOMATIONDIRECT.COM TECHNICAL SUPPORT IS PROVIDED "AS IS" WITHOUT A GUARANTEE OF ANY KIND. These documents are provided by our technical support department to assist others. We do not guarantee that the data is suitable for your particular application, nor do we assume any responsibility for them in your application.

Product Family: Cmore

Number: AN-EA-004

Subject: Using Cmore HMI with Allen Bradley PLCS

Date Issued: 1020-2011

Revision: A

1. Compatible Allen Bradley PLCs	2
2. Supported Data Types for SLC, MicroLogix and PLC5.	
a. Compatible Data Type Chart	
b. Syntax for entering SLC, MicroLogix and PLC5 addressing into C	-more
programming software.	
3. Cable Diagrams	9
4. Control/Compact/FlexLogix Tag support	
a. Terminology (Data types and memory types)	
b. Tag types	
I. Atomic	
II. Pre-defined	13
III. Arrays	
IV. User-defined	14
V. I/O Tags	15
VI. Aliasing	15
VII. Program scope Tags	16
5. Getting Logix Tags into C-more programming software	17
a. Importing RSLogix .L5K file (Standard method)	17
 Importing through C-more .CSV file (used for bringing in I/O Tags 	with Aliases)
c. Manual Entry	
6. Importing SLC, MicroLogix and PLC5 Symbols	
Step-by-Step, Exporting Symbols from RSLogix500 and Importing into	C-more:30
7. Optimizing communications between C-more and Allen Bradley PLC	Cs 32
a. Optimization for non-Tag based PLCs (SLC and MicroLogix)	
 b. Optimization for Tag based PLCs (Control/Compact/FlexLogix PL 	Cs) 33
8. Version History	33



1. <u>Compatible Allen Bradley PLCs</u>

C-more supports 5 Allen Bradley PLC drivers: DF1 Full Duplex, DF1 Half Duplex, DH485, EtherNet/IP Client driver and EtherNet/IP Server driver (generic).

A compatibility matrix is shown below that illustrates which Allen Bradley PLCs are supported by C-more with the appropriate communications protocol.

C-more [©] Allen-Bradley [®] Support	DF1 Full Duplex	DF1 Half Duplex	DH485	Generic EtherNet/IP Server (I/O Messaging)	EtherNet/IP Client	EtherNet/IP Tag-Based Client	Tag-Based DF1 Full Duplex	Tag-Based DF1 Half Duplex
SLC™ 5/01, 5/02			Yes					
SLC™ 5/03	Yes	Yes	Yes		Yes 1,3			
SLC™ 5/04	Yes	Yes	Yes ²		Yes ^{1,3}			
SLC™ 5/05 (Series A OS501 FRN5 & Higher)	Yes	Yes	Yes ²		Yes ¹			
MicroLogix™ 1000, 1200, 1500	Yes	Yes	Yes ²		Yes ^{1,3}			
MicroLogix™1100	Yes	Yes	Yes ²		Yes ¹			
PLC-5	Yes							
ControlLogix®				Yes ¹		Yes	Yes	Yes
CompactLogix®				Yes ¹		Yes	Yes	Yes
FlexLogix				Yes ¹		Yes	Yes	Yes
Ethernet available on full featured C-more™ panels onlyR units AlC module from Allen-Bradley® required for this connection NET - ENI module from Allen-Bradley® required for this connection	do not support Et n	hernet						

Listed below are explanations as to when each protocol should be used:

<u>DF1 Full Duplex:</u> Used for point to point serial connections. If you are only using 1 C-more panel and 1 PLC, this is the most economical method of connecting C-more to an AB PLC.

DF1 Half Duplex: This is an economical method for connecting 1 C-more panel to multiple PLCs. Note that none of the Allen Bradley PLCs have RS-422 or RS-485 for their serial connection so a RS-232 to RS-422/485 converter (such as the FA-ISOCON) is required for each PLC.

<u>Generic EtherNet/IP Server</u>: This driver is atypical in that the messaging is initiated by the PLC instead of C-more. The C-more panel is treated basically like a Remote I/O drop. It contains a block of Input Words and Output Words and the PLC either reads or writes this data back and forth. The PLC tags are not directly referenced as in the EtherNet/IP Client driver. This method is sometimes preferred by some customers for different applications.

EtherNet/IP Client: This driver is the more typical method of reading and writing data back and forth from the PLC. The C-more panel directly references the PLC tags and initiates the messages going back and forth. If your application involves using multiple C-more panels and/or multiple PLCs, this is the best method for accomplishing this.

<u>DH485</u>: This driver is mostly for legacy systems where it is desired to add a C-more panel to an existing network. It is possible with DH485 to have multiple C-more panels and multiple PLCs



on the same network but the performance is much less than EtherNet/IP. DH485 is not as robust as DF1 or EtherNet/IP.

Supported Data Types for SLC, MicroLogix and PLC5. a. Compatible Data Type Chart

	SLC	MicroLogix	PLC5
I = Input	*X	Х	Х
O = Output	*X	Х	Х
S = Status	Х	Х	Х
B = Binary	Х	Х	Х
T = Timer	Х	Х	Х
C = Counter	Х	Х	Х
R = Control	Х	Х	Х
N = Integer	Х	Х	Х
F = Float	Х	Х	Х
**L = Long		Х	
**ST = String	Х	Х	

* I/O access not supported for SLC5/01 and 5/02

** **<u>NOTE</u>**: As of version 2.10, Long and String data types were not yet supported. They should be available by Q3, 2008. Refer to the release notes of the C-more software on our website to see when these data types are supported.

b. Syntax for entering SLC, MicroLogix and PLC5 addressing into C-more programming software.

ag Name - Add		×
No. 🖊 ┥ 68	▶ ₩	
Tag Information		
Device Name:	DEV001	
Tag Name:	N7:0	
Tag Data Type:	Signed int 16	
Characters:	40 Retentive	
PLC Address		
Memory Type	Address [7:0	
	Add Close Help	

Syntax for Integer data type, file 7, word 0 (N7:0).



These documents are provided by our technical support department to assist others. We do not guarantee that the data is suitable for your particular application, nor do we assume any responsibility for them in your application.

ng Name - Edit No. 🖊 ┥ 1	 ► ₩	
Tag Information		
Device Name:	DEV001	•
Tag Name:	N7:0/0	
Tag Data Type:	Discrete	•
Characters:	10 -	E Retentive
PLC Address		
Memory Type	Address	Bit
	7:0	

Syntax for Integer data type, file 7, word 0 and bit 0 (N7:0/0).

lo. 🖊 ┥ 🛛		
ag Information		
Device Name:	DEV001	-
Tag Name:	B3:0	
Tag Data Type:	Signed int 16	
Characters:	10 E Retentive	9
PLC Address		
Memory Type	Address	
B	3:0	
		197725

Syntax for Binary data type, file 3, word 0 (B3:0).

Tag Name - Add		×
No. 🖊 ┥ 68	► H	
Tag Information		
Device Name:	DEV001	•
Tag Name:	B3:0/0	
Tag Data Type:	Discrete	•
Characters:	40 *	Retentive
PLC Address		
Memory Type	Address	Bit
B	3:0	0 💌

Syntax for Binary data type, file 3, word 0, bit 0 (B3:0/0).



These documents are provided by our technical support department to assist others. We do not guarantee that the data is suitable for your particular application, nor do we assume any responsibility for them in your application.

Tag Name - Add	×
No. 🖌 ┥ 68	► H
Tag Information	
Device Name:	DEV001
Tag Name:	F8:0
Tag Data Type:	Floating PT 32
Characters:	40 × Retentive
PLC Address	
Memory Type	Address
F	8:0
	Add Close Help

Syntax for Floating data type, file 8, word 0 (F8:0).

Tag Name - Add	×
No. 🖌 🔺 68	▶ ₩
Tag Information	
Device Name:	DEV001
Tag Name:	1:0
Tag Data Type:	Signed int 16
Characters:	40 💌 🗖 Retentive
PLC Address	
Memory Type	Address
	:0
	Add Close Help

Syntax for Input data type, slot 0 (I:0). File number is fixed at 1. Entering the file number (1:x) in the address field is also accepted by the C-more programming software. At the present time (Version 2.10 and earlier); using the expanded slot.word format is not supported. Please refer to the release notes for the C-more programming software on our website as new versions are released to look for this feature.

ag Name - Add		
No. 🖌 🖌 68	▶ ₩	
Tag Information		
Device Name:	DEV001	▼
Tag Name:	1:0/0	
Tag Data Type:	Discrete	•
Characters:	40 ×	E Retentive
PLC Address		
Memory Type	Address	Bit
	:0	0 💌
		E.

Syntax for Input data type, slot 0, bit 0 (I:0/0). File number is fixed at 1. Entering the file number (1:x) in the address field is also accepted by the C-more programming software.



THIS INFORMATION PROVIDED BY AUTOMATIONDIRECT.COM TECHNICAL SUPPORT IS PROVIDED "AS IS" WITHOUT A GUARANTEE OF ANY KIND. These documents are provided by our technical support department to assist others. We do not guarantee

These documents are provided by our technical support department to assist others. We do not guarantee that the data is suitable for your particular application, nor do we assume any responsibility for them in your application.

Tag Name - Add	×
No. 🖌 ┥ 68	▶ ₩
Tag Information	
Device Name:	DEV001
Tag Name:	0:0
Tag Data Type:	Signed int 16
Characters:	40 Retentive
PLC Address	
Memory Type	Address 0
	Add Close Help

Syntax for Output data type, slot 0 (O:0). File number is fixed at 0. Entering the file number (0:x) in the address field is also accepted by the C-more programming software. At the present time (Version 2.10 and earlier); using the expanded slot.word format is not supported. Please refer to the release notes for the C-more programming software on our website as new versions are released to look for this feature.

10. M 4 13		
ag Information		
Device Name:	DEV001	•
Tag Name:	0:0/0	
Tag Data Type:	Discrete	•
Characters:	40 *	E Retentive
LC Address		
Memory Type	Address	Bit
0	:0	0 -

Syntax for Output data type, word 0, bit 0 (O:0/0). File number is fixed at 0. Entering the file number (0:x) in the address field is also accepted by the C-more programming software.



These documents are provided by our technical support department to assist others. We do not guarantee that the data is suitable for your particular application, nor do we assume any responsibility for them in your application.

Tag Name - Add		x
No. 🖊 ┥ 68	▶ ₩	
Tag Information		
Device Name:	DEV001	
Tag Name:	T4:0.PRE	
Tag Data Type:	Signed int 16	
Characters:	40 Retentive	
Characters:	40 × Retentive	
Characters: PLC Address Memory Type T	40 Address Flag 40	

Syntax for Timer data type, file 4, word 0, preset member (T4:0.PRE). The .ACC member is also supported.

No. 🖌 🔺 🗍	► H	
Tag Information		
Device Name:	DEV001	•
Tag Name:	T4:0.TT	
Tag Data Type:	Discrete	•
Characters:	10 -	E Retentive
PLC Address		
Memory Type	Address	Flag
T	4:0	TT 💌
	Apply	Close Help

Syntax for Timer data type, file 4, word 0, timer timing flag (T4:0.TT). The .EN and .DN members are also supported.

Tag Information	F	
Device Name:	DEV001	<u> </u>
Tag Name:	C5:0.ACC	
Tag Data Type:	Signed int 16	-
Characters:	10 -	E Retentive
PLC Address		
	A 11	Flag

Syntax for Counter data type, file 5, word 0, accumulated member (C5:0.ACC). The .PRE member is also supported.



These documents are provided by our technical support department to assist others. We do not guarantee that the data is suitable for your particular application, nor do we assume any responsibility for them in your application.

lo. 🖊 ┥ 🛛		
Device Name:	DEV001	•
Tag Name:	C5:0.DN	
Tag Data Type:	Discrete	•
Characters:	10 +	🗖 Retentive
PLC Address		
Memory Type	Address	Flag
C 🔹	5:0	DN 💌

Syntax for Counter data type, file 5, word 0, done flag (C5:0.DN). The .CU, .CD, .OV and .UN members are also supported.

Tag Name - Add			×
No. 🖊 ┥ 68	▶ ₩		
Tag Information			
Device Name:	DEV001		•
Tag Name:	S:42		
Tag Data Type:	Signed int 16	•	
Characters:	40 *	🗖 Re	tentive
PLC Address			
Memory Type	Address		
S 💌	:42		

Syntax for Status data type, word 42 (S:42). File number is fixed at 2.

Tag Name - Add	×
No. 🖊 ┥ 68	▶ ₩
Tag Information	
Device Name:	DEV001
Tag Name:	S:0/0
Tag Data Type:	Discrete
Characters:	40 Retentive
PLC Address	
Memory Type	Address Bit
S 💌	:0
	Add Close Help

Syntax for Status data type, word 0, bit 0 (S:0/0). File number is fixed at 2.



3. <u>Cable Diagrams</u>

Allen-Bradley:



EA-SLC-232-CBL



EA-PLC5-232-CBL





These documents are provided by our technical support department to assist others. We do not guarantee that the data is suitable for your particular application, nor do we assume any responsibility for them in your application.





These documents are provided by our technical support department to assist others. We do not guarantee that the data is suitable for your particular application, nor do we assume any responsibility for them in your application.





These documents are provided by our technical support department to assist others. We do not guarantee that the data is suitable for your particular application, nor do we assume any responsibility for them in your application.



For more information on wiring diagrams, refer to the C-more Hardware User manual (EA-USER-M).



4. <u>Control/Compact/FlexLogix Tag support</u>

a. Terminology (Data types and memory types)

There are different terms used to describe the different data types in the Allen Bradley Logix PLCs. Below is a table showing the data type syntax in the AB PLC, the data type syntax used for tags in the C-more programming software and the meaning of each.

Logix Memory Type	C-more Tag Data Type	Description
DINT	Signed int 32	Memory area with a value of 32 bit 2s compliment integer -2,147,483,648 to 2,147,483,647
INT	Signed int 16	Memory area with a value of 16 bit 2s compliment integer -32,768 to 32,767
SINT	Ascii String	Memory area with a value of 8 bit used primarily as a character byte location up to 128 characters
BOOL	Discrete	Memory area with a value of 1 bit representing 0 as False and 1 as True
REAL	Floating PT 32	Memory area with a value of 32 bit floating point -1.1754944e- 38 to 3.4028237 e38
STRING	Ascii String	Memory area with a character array up to 82 characters.

b. Tag types

I. Atomic

The Atomic data types are the base data type unit of which all other complex data types are comprised of. The Atomic data types are: BOOL, SINT, INT, DINT and REAL. The syntax for the Atomic data types is simply the tag name itself (no pre cursors or delimiters). An example would be: Tank1_Level.

All Atomic data types are addressable in the C-more programming software and are importable via the L5K file.

II. Pre-defined

Pre-defined data types are complex data types that are made up of 1 or more Atomic data types, Arrays, or other Pre-defined data types. They are commonly called "Structures" in other programming languages such as C. The Pre-defined data types already exist in RS Logix 5000 with every new project that is created. One of the most common Pre-defined data types is the TIMER data type. A TIMER data type is comprised of these members: TIMER:

.PRE (preset member that is a DINT atomic data type)

THIS INFORMATION PROVIDED BY AUTOMATIONDIRECT.COM TECHNICAL SUPPORT



IS PROVIDED "AS IS" WITHOUT A GUARANTEE OF ANY KIND. These documents are provided by our technical support department to assist others. We do not guarantee that the data is suitable for your particular application, nor do we assume any responsibility for them in your application.

.ACC (accumulated member that is a DINT atomic data type)

.EN (enabled member that is a BOOL atomic data type)

.TT (timer timing member that is a BOOL atomic data type)

.DN (done member that is a BOOL atomic data type)

.FS (first scan member that is a BOOL atomic data type)

.LS (last scan member that is a BOOL atomic data type)

.OV (over member that is a BOOL atomic data type)

.ER (error member that is a BOOL atomic data type)

So if you create a TIMER called Cycle in the PLC, you will have these tags:

Cycle.PRE, Cycle.ACC, Cycle.EN, Cycle.TT, Cycle.DN, Cycle.FS, Cycle.LS, Cycle.OV and Cycle.ER.

There are hundreds of Pre-defined data types available in the RS Logix 5000 software.

All Pre-defined data types are addressable in the C-more programming software. The Predefined data types can be imported via the L5K file but there is an L5KPreDefine.txt document that must be modified for this purpose. Refer to section 5.a. for details on how to do this.

III. <u>Arrays</u>

An array is basically a data structure of a sequential set of the same data type. Each element is accessed by an index number. Arrays can be created from Atomic data types or complex data types (Pre-defined or User-defined).

All Array Tags are addressable in the C-more programming software. Array Tags (depending upon what data type array) can be imported via L5K file. You may choose to limit the number of imported elements by adjusting the "Import Array Element Count Limit" field in the C-more programming software import window (shown below).

Import Tag name Database from Excel/CSV I	File	×
File Path: C\temp\example L5k F	ILES\ADC_RSLogix5000_Example.L5K	
-Import Option		
Overwrite Existing Tags	-L5K Import Option	
	Device: DEV001 Add	
	Protocol: Allen-Bradley EtherNet/IP Client Tag-Based (Control/Com	p
	Append Text	
My file contains a Header Row	Import Array Element Count Limit(1-30000):	
1st Row is not imported.	3000	
	-Alias Option	
	Do not import Aliases	
	 Import all Aliases & Tags 	
	C Import only Allases	
	C Import Allases & Lags that do not have Allases	
	Import Cancel Help	

IV. <u>User-defined</u>

User-defined data types are complex data types that are created by the user. They are very much like the Pre-defined structures but the user may choose which data type members the User-defined data type is comprised of. It is a very efficient way to organize data in the PLC.

Here is an example of a User-defined data type:

DataType_A:

DintMember (which is a DINT member of this User-defined data type)



IntArray[10] (which is a INT array of this User-defined data type) BoolArray[32] (which is a BOOL array of this User-defined data type)

If a Tag called "Example1" was added using the "DataType_A" data type, the RS Logix5000 software would create these tags:

Example1.DintMember

Example1.IntArray[0] – Example1.IntArray[9]

Example1.BoolArray[0] – Example1.BoolArray[31]

All User-defined data types are addressable in the C-more programming software. Userdefined data types are also importable via L5K file depending upon the members of the userdefined data type. See section 5.a. for details on this.

V. I/O Tags

I/O Tags that are created automatically when an I/O module is added to the configuration. They are typically members of a complex data type. The complex data types differ based upon the module being configured and the various configuration options that the user has selected.

These data types are not directly importable into C-more via the L5K file but must be imported using the standard C-more Tag importation. There is a macro function available to help make I/O tags importation easier (see Section 5.b.).

VI. Aliasing

Aliasing is a method to 'de-reference' that user tag from the memory location in the PLC. It could be said that Tag names themselves do this very thing but Aliasing creates another abstraction layer from tags such as I/O tags that can and will change very often. Aliasing is very useful for HMIs, where the data point is needed but the constant hassle of changing the name can become tedious. An example of an Alias in the RS Logix 5000 software is shown below:

Ø	🖉 Controller Tags - Alias_Example(controller) 🛛 🗖 🔀						
Se	Scope: Alias_Example(contr Show: Show All Sout: Alias For						
	Ρ	Tag Name	Alias For 🛛 🛆	Туре	Style 🔺		
		—-Local:1:I		AB:1756_DI:I:0			
	E-Local:1:I.Fault			DINT	Binary		
	+ Local:1:I.Data			DINT	Binary		
		+-Local:1:C		AB:1756_DI:C:0			
			Local:1:I.Data	DINT	Binary 🗾 👻		
4	Monitor Tags Edit Tags						

In the example above, "Photoeye_Status" is an Alias to the I/O tag "Local:1:I.Data".

There are several Alias import options available in C-more (shown below). Here is an explanation of these options:

"Do not import Aliases" = This means, simply, that the C-more programming software will not import any Alias tag names.

"Import all Aliases & Tags" = This means that the C-more programming software will import everything that it can. If there are many Alias tags in the project, this method will result in



duplicate tag references since you will have imported the base tag itself, as well as the Alias of that tag.

"Import only Aliases" = This option will only import those Tags that have Aliases. "Import Aliases & Tags that do not have Aliases" = This option will import Alias tags but not the base tag that the Alias refers to and it will import tags that do not have Aliases.

Unfortunately, only importation of Aliases to BOOL I/O tags can be done with the method shown below because the memory size for all other I/O data types cannot be derived from the L5K file. To import Aliases of all non-BOOL I/O tags use the method outlined in section 5.b.

import ray name batabase from Excel/CSV						
File Path: C\temp\example L5k	<pre><files\adc_rslogix5000_example.l5k< th=""></files\adc_rslogix5000_example.l5k<></pre>					
-Import Option						
Overwrite Existing Tags	L5K Import Option					
	Device: DEV001 Add					
	Protocol: Allen-Bradley EtherNet/IP Client Tag-Based (Control/Comp					
	Append Text					
My file contains a Header Row	Import Array Element Count Limit(1-30000):					
1st Row is not imported.	3000					
	Alias Option O not import Aliases Import all Aliases & Tags Import only Aliases Import only Aliases Import Aliases & Tags that do not have Aliases					
	Import Cancel Help					

VII. <u>Program scope Tags</u>

Tags can be in 2 locations in the PLC. They can be either Controller Scope (globally accessible anywhere) or Program Scope (only available within that Program).

Program Scope tags can be accessed via C-more or other HMIs but they require additional syntax in front of the tag name. To access a Program Scope tag, you must enter PROGRAM:program name.tag name where PROGRAM is static text to identify that a Program Scope tag is following.



5. <u>Getting Logix Tags into C-more programming software</u>

a. Importing RSLogix .L5K file (Standard method)

1. Within RSLogix 5000 and with your project file open, Select File>SaveAs

<mark>81</mark>	RSLo	gix 5	000 - 9	Smar	tBelt in Sma	rt_Bel
File	Edit	View	Search	Logic	Communications	Tools
Ē	<u>N</u> ew					Ctrl+N
È	Open					Ctrl+0
	<u>C</u> lose					
	<u>S</u> ave					Ctrl+S
	Save A	<u>\</u> s				
	New C	omnone	ant			•

2. Change the "Save as type" to L5K

File name:	Smart_Belt	Save
Save as type:	RSLogix 5000 Import/Export File (*.L5K)	Cancel
	RSLogix 5000 Project File (*.ACD) RSLogix 5000 Import/Export File (*.L5K)	Help

3. Open the C-more Programming Software and select Import>Tag Name Database



File Edit	View	Tool	Object	Screen	Database	Setup	Pane	sl Wir
Start Pr Open Pi Close Pi	oject roject roject			Ctrl+0	sil	nula	Lar te	iguag Proje
Save Pr Save Pr Save Pr	oject oject A oject A	s s Prote						
Import				۱.	Tag Nar	ne Datal	base	
Export				+	Messag	e Databi	ase	
Simulati Project	on Transfe	er		Ctrl+T	Address Langua	s Book ge Edit		
Dript				Chrlind	Project	Data		
Print Pr	eview			Curr	Library.			
Print Se	tup				Recipe	Sheet		
Project Project	Informa Option.	ation						
1 07126 2 EARui 3 Howm 4 Scree	i.eap n.eap iet.eap nCaptu	reCras	h.eap					
Exit								



4. Import Tag Database into C-more Programming, Import>Tag Database>Select L5K File

Import Tag name Database from Excel/CSV	File	Ň
File Path: C\My Documents\Sm	art_Belt.L5K	
Import Option		
Overwrite Existing Tags	L5K Import Option	
	Device: DEV001 Add	
	Protocol: Allen-Bradley DF1 Full Duplex Tag-Based (Control/Compa	-
	Append Text:	
My file contains a Header Row	Import Array Element Count Limit(1-30000):	
1st Row is not imported.	3000	
	Alias Option	
	Do not import Aliases	
	Import all Aliases & Tags	
	Import only Aliases	
	Import Aliases & Tags that do not have Aliases	
	Import Cancel Help	

Note: If many errors are reported these are probably from tags which are Aliases or Physical I/O. See the application notes mentioned below for information on how to import these separately through another process.

Importing Pre-defined data types:

There are very many Pre Defined data types in the Allen Bradley ControlLogix and CompactLogix PLCs. A Pre Defined data type is a data type that is already created in the RS Logix 5000 software when starting a new project. Some common Pre Defined data types are TIMER and COUNTER.

The complication of importing Pre Defined data type is that there are very, very many Pre Defined data types in the RS Logix 5000 software and within many of these data types, there are many different data type members. Many/most of these members of these Pre Defined data types, while useful in the PLC application, will not be needed in the C-more project. So the complication is to decide which Pre Defined data types should be imported and which members of these data types should or shouldn't be imported.



Defining which data types to import:

There is a file located inside of the root directory of the C-more programming software that can be modified to determine which data types and their members should be imported into the C-more project. The image below shows where the C-more programming software is installed by default. The location of your C-more programming software may be different, depending upon how the software was installed.



The name of this file (as shown above) is L5KPreDefine.txt and by default looks like the image below:

🐌 L5KPreDefine.txt - Notepad	
<u>File E</u> dit F <u>o</u> rmat <u>V</u> iew <u>H</u> elp	
DATATYPE TIMER BOOL EN; BOOL DT; BOOL DN; BOOL FS; BOOL LS; BOOL UV; BOOL ER; DINT PRE; DINT ACC; END_DATATYPE	*
DATATYPE COUNTER BOOL CU; BOOL CD; BOOL ON; BOOL OV; BOOL UN; DINT PRE; DINT PRE; DINT ACC; END_DATATYPE	×

So, by default, the C-more programming software will import the TIMER and COUNTER Pre Defined data types.



To add more Pre Defined data types, you can type them in using the format shown above or you can open up a master template file called: L5KPreDefine_template.txt. This file can be downloaded from Automationdirect.com on the Free Software, Upgrades section of the tech support page. It will be listed under the Tools and Utilities section. This file can also be found in the Help folder from the downloaded Install zip file or on the C-more Programming Software CD.

Here is an example of how to add the SELECT data type to the list of the Pre Defines that will be imported. Open up the L5KPreDefine_template.txt file and scroll down to the DATATYPE SELECT heading as shown below:

1	_ 5KP re	Define_t	empla	te.txt - Notepad	
Eile	<u>E</u> dit	F <u>o</u> rmat	View	Help	
END)_DAT	BOOL BOOL BOOL BOOL BOOL BOOL BOOL ATYPE		WLEADINV; ZETALEADINV; HIGHLOWLIMSINV; TIMINGMODEINV; RTSMISSED; RTSTIMEINV; RTSTIMESTAMPINV; DELTATINV;	<u>~</u>
DAT	D_DAT	E SELE BOOL REAL REAL BOOL BOOL REAL ATYPE	СТ	ENABLEIN; IN1; IN2; SELECTORIN; ENABLEOUT; OUT;	
DAT	D_DAT	E SELE BOOL REAL BOOL BOOL REAL ATYPE	CTABI	LE_NEGATE ENABLEIN; IN; NEGATEENABLE; ENABLEOUT; OUT;	_
4					▼ //

Use the cursor and highlight the entire section for the SELECT data type beginning at DATATYPE and ending at END_DATATYPE.

📕 L5KP	reDefine_t	templa	te.txt - Notepad	
<u>E</u> ile <u>E</u> di	: F <u>o</u> rmat	View	Help	
	BOOL		WLEADINV;	
	BOOL		ZETALEADINV;	
	BOOL		HIGHLOWLIMSINV;	
	BOOL		DISMISSED.	
	BOOL		PTSTIMETNV/	
	BOOL		RTSTIMESTAMPINV:	
	BOOL		DELTATINV;	
END_DA	TATYPE			
DATATY	PE SELE	СТ		
	BOOL		ENABLEIN;	
	REAL		IN1;	
	REAL		IN2;	
	BOOL		SELECTORIN;	
	BOOL		ENABLEOUT;	
			001;	
	TATTFE			
DATATY	PE SELE	CTAB	LE NEGATE	
	BOOL		ENABLEIN;	
	REAL		IN;	
	BOOL		NEGATEENABLE;	
	BOOL		ENABLEOUT;	
	REAL		001;	
END_DA	TATYPE			•
•				

Click on Edit and Copy to copy the highlighted area. Then click over in the L5KPreDefine.txt file and go to the bottom and click on Edit and Paste to make the L5KPreDefine file look like below:



These documents are provided by our technical support department to assist others. We do not guarantee that the data is suitable for your particular application, nor do we assume any responsibility for them in your application.

📕 L	5KPre	Define.txt	- Notepad					×
<u>F</u> ile	<u>E</u> dit	Format \	<u>∕</u> iew <u>H</u> elp					
END.	_DAT	BOOL BOOL BOOL BOOL DINT DINT ATYPE	FS; LS; OV; ER; PRE; ACC;					
DAT	ATYP _DAT	E COUNT BOOL BOOL BOOL BOOL BOOL DINT DINT ATYPE	ER CD; DN; OV; UN; PRE; ACC;					
DAT.	ATYP _DAT	E SELEC BOOL REAL REAL BOOL BOOL REAL ATYPE	T EN IN IN SE EN OU	ABLEIN; 1; 2; LECTORIN; ABLEOUT; T;				•
-								

If you wish to import all the members of the SELECT data type shown above then all that needs to be done is to save and close the L5KPreDefine.txt file and re start the Cmore programming software.

You can also remove the members of the data type that are needed in the C-more project as shown below:

📕 L5KPreDefine.b	ct - Notepad	
<u>F</u> ile <u>E</u> dit F <u>o</u> rmat	<u>V</u> iew <u>H</u> elp	
BOOL BOOL BOOL BOOL BOOL BOOL DINT END_DATATYPE	EN; TT; DN; FS; LS; OV; ER; PRE; ACC;	
DATATYPE COUN BOOL BOOL BOOL BOOL DINT DINT END_DATATYPE	TER CU; CD; DN; OV; UN; PRE; ACC;	
DATATYPE SELE REAL REAL REAL END_DATATYPE	CT IN1; IN2; OUT;	•

It is important to remember the format required in the L5KPreDefine.txt file: DATATYPE Desired data type name as shown in RS Logix 5000 Memory type (IE BOOL, DINT, etc...) member name (as shown in RS Logix); END_DATATYPE

The semi colon is necessary after each member name of the data type.



b. Importing through C-more .CSV file (used for bringing in I/O Tags with Aliases)

The information below is obtained from Application Note: <u>AN-EA-002</u>. It also requires a spreadsheet called <u>IO Import Utility.xls</u> that can be found by clicking on the filename. These documents can be found in the Tech support application note section of our website (<u>http://support.automationdirect.com/technotes.html</u>).

Starting with <u>C-more Programming Software Version 2.1</u> it will import Alias Tags and Boolean I/O Tags. These can be imported directly using the L5K file (generated by "Save-As" from within RSLogix.) Only none Boolean I/O Tags will need any special handling by the user. This document describes a method to generate those tags for C-more quickly and easily without having to retype them. This process will require licensed installs of the following software; RSLogix5000, Microsoft Excel, and the C-more Programming Software.

It is desirable to assign Alias Tag Names to any I/O that is accessed by the Operator Panel. This can reduce the total number of tags residing in the Panel and also prevent the need to make changes in the panel if the I/O is remapped at a later time (ex: card slot number change).

Within the RSLogix Tag Viewer:

- 1. Open the Controller Tags window by double-clicking in the Controller Organizer
- 2. Select the "Edit Tags" tab along the lower edge of the window



- 3. (optional) Sort the tags by right-clicking the Tag Name field then "Sort"
- 4. Find your I/O Tags starting with "Local: ..."
- 5. Expand the tree to display all the tags that you require (only visible tags will be copied, you must expand any User Define Data Types to their basic elements)



These documents are provided by our technical support department to assist others. We do not guarantee that the data is suitable for your particular application, nor do we assume any responsibility for them in your application.

1	Controller Tags - IO_Testing(controller)										
S	Scope: ID_Testing(controlle - Show All - Sort: Tag Name -										
	P Tag Name 🛆 /	Alias For	Base Tag	Туре	Style	Description					
	MOTOR_STATUS_1	Local:1:1.Data.0	Local:1:1.Data.0	BOOL	Decimal						
	MOTOR_STATUS_2	Local:1:1.Data.1	Local:1:1.Data.1	BOOL	Decimal						
	MOTOR_STATUS_3	Local:1:I.Data.2	Local:1:I.Data.2	BOOL	Decimal						
	MOTOR_STATUS_4	Local:1:I.Data.3	Local:1:I.Data.3	BOOL	Decimal						
	-Local:1:I			AB:1756_DI:I:0							
				DINT	Binary						
	- Local:1:I.Data			DINT	Binary						
	Local:1:I.Data.0			BOOL	Decimal						
	-Local:1:I.Data.1			BOOL	Decimal						
	-Local:1:I.Data.2			BOOL	Decimal						
	-Local:1:I.Data.3			BOOL	Decimal						
	-Local:1:I.Data.4			BOOL	Decimal						
	Local:1:I.Data.5			BOOL	Decimal						

6. Select the first I/O tag you want by clicking the blank square on the left

Controller Tags - 10_Testing(controller)										
Scope: I0_Testing(controlle V Show: Show All V Sout: Tag Name V										
P Tag Name 🛆	Alias For	Base Tag	Туре	Style	Description					
MOTOR_STATUS_1	Local:1:1.Data.0	Local:1:1.Data.0	BOOL	Decimal						
MOTOR_STATUS_2	Local:1:1.Data.1	Local:1:1.Data.1	BOOL	Decimal						
MOTOR_STATUS_3	Local:1:1.Data.2	Local:1:1.Data.2	BOOL	Decimal						
MOTOR_STATUS_4	Local:1:1.Data.3	Local:1:1.Data.3	BOOL	Decimal						
-Local:1:I			AB:1756_DI:I:0							
Eocal:1:I.Fault			DINT	Binary						
- Local:1:I.Data			DINT	Binary						
Local:1:I.Data.0			BOOL	Decimal						
Local:1:I.Data.1			BOOL	Decimal						
Local:1:I.Data.2			BOOL	Decimal						
Local:1:I.Data.3			BOOL	Decimal						
Local:1:I.Data.4			BOOL	Decimal						
Local:1:1 Data 5			8001	Decimal						

- 7. Hold the shift key down
- 8. Select the last I/O tag desired (Page Down or Mouse Down as needed)

Controller Tags - 10_Testing(controller)											
Scope: IO_Testing(controlle - Show: Show	Scope: IO_Testing(controlle - Show: Show All - Sott: Tag Name -										
P Tag Name 🛆	Alias For	Base Tag	Туре	Style	Description						
MOTOR_STATUS_1	Local:1:1.Data.0	Local:1:I.Data.0	BOOL	Decimal							
MOTOR_STATUS_2	Local:1:1.Data.1	Local:1:I.Data.1	BOOL	Decimal							
MOTOR_STATUS_3	Local:1:1.Data.2	Local:1:I.Data.2	BOOL	Decimal							
MOTOR_STATUS_4	Local:1:1.Data.3	Local:1:I.Data.3	BOOL	Decimal							
-Local1:I			AB:1756_DI:I:0								
			DINT	Binary							
Local:1:I.Data			DINT	Binary							
Local:1:I.Data.0			BOOL	Decimal							
Local:1:I.Data.1			BOOL	Decimal							
Local:1:I.Data.2			BOOL	Decimal							
Local:1:I.Data.3			BOOL	Decimal							
Local:1:I.Data.4			BOOL	Decimal							
Local:1:1.Data.5			BOOL	Decimal	-						

- 9. Release the shift key (those tags should remain selected)
- 10. Edit > Copy
 - Note: If you only select I/O tags the Edit > Copy feature will not be available, you must include a standard tag within your selected block to be copied.



These documents are provided by our technical support department to assist others. We do not guarantee that the data is suitable for your particular application, nor do we assume any responsibility for them in your application.

🎉 R	SLog	jix 500	00 - 10_	Testi	ng [1756-l
File	Edit	View	Search	Logic	Communica
酉	кŲ	<u>U</u> ndo			Ctrl+Z
_ <u> </u>	¢ы	<u>R</u> edo			Ctrl+Y
Offli					
No F	ž	Cu <u>t</u>			Ctrl+X
No E	Ē)	⊆ору			Ctrl+C

- 11. Open Excel and the file, IO Import Utility.xls
- 12. Select the "Paste Version XX Tags Here" worksheet for either RSLogix V13 or V15(V16)
- Note: For other versions you may have to move columns around to align the correct fields

2	A B	C	D		E	F	G	Н
1	RSLogix Version 13, Sele	ect Cell A3	then Past	e _	Click here v	vhen c	omplete	Clear All
2	P <u>Tag Name</u>	Alias For	Base Tag	Type		Style	Description	
3								
4				1				
5				1				

13. Edit > Paste

	A B	C	D	E	F	G	Н	
1	RSLogix Version 13, Select Cell A3 then Paste Click here when complete							
2	P Tag Name	Alias For	Base Tag	<u>Type</u>	Style	Description		
3	MOTOR_STATUS_1	Local:1:I.Data.0	Local:1:1.Data.0	BOOL	Decimal		~	
4	MOTOR_STATUS_2	Local:1:I.Data.1	Local:1:I.Data.1	BOOL	Decimal			
5	MOTOR_STATUS_3	Local:1:I.Data.2	Local:1:I.Data.2	BOOL	Decimal			
6	MOTOR_STATUS_4	Local:1:I.Data.3	Local:1:I.Data.3	BOOL	Decimal			
7	Local:1:1			AB:1756_DI:I:0				
8	Local:1:I.Fault			DINT	Binary			
9	Local:1:I.Data			DINT	Binary			
10	Local:1:I.Data.0			BOOL	Decimal			
11	Local:1:I.Data.1			BOOL	Decimal		с 	
12	Local:1:I.Data.2			BOOL	Decimal		1	
13	Local:1:I.Data.3			BOOL	Decimal			
14	Local:1:I.Data.4			BOOL	Decimal			
15								

14. Delete any unnecessary tags

- 15. Repeat Steps 3 through 14 for any Alias Tags you require (using Sort by: Alias)
- 16. After you have Copy-Pasted all your I/O and Alias tags, Click on "Click here when complete"
- 17. If your tags are anything except Controller Scope then you'll need to generate separate files for each PLC program. C-more will allow you to import multiple CSV files, or you can combine them into one file before importing.
- 18. Click the "Continue" button to create tags for C-more.
- 19. File > Save-As > then change file type to CSV and Save



These documents are provided by our technical support department to assist others. We do not guarantee that the data is suitable for your particular application, nor do we assume any responsibility for them in your application.

-	ave As			
	Save in:	Conor	💌 🔮 – 🗖 🕅 🕰 🗙 📑 🧱	Tools
	My Recent Documents	iOld 🔮 IO Import U	Jtility.×ls	
	Desktop			
	My Documents			
	My Computer			
	- S	File <u>n</u> ame:	IO Import Utility.xls	Save
	My Network Places	Save as <u>t</u> ype:	Microsoft Office Excel Workbook (*.xls)	Cancel
ect	"OK"		Template (*.xlt) Text (Tab delimited) (*.txt) Unicode Text (*.txt) Microsoft Excel 5.0/95 Workbook (*.xls) Microsoft Excel 97- Excel 2003 & 5.0/95 Workbook (*.xls) CSV (Comma delimited) (*.csv)	
	OK			
195	The select The select The select To save To save	ed file type does n only the active she all sheets, save th	ot support workbooks that contain multiple sheets. et, click OK. em individually using a different file name for each, or choose a file type th OK Cancel	at supports multiple sheets.
ect	"Yes"			
oft I	ixcel			
	O Import Utility.csv To keep this forma To preserve the fe To see what might	may contain feat t, which leaves ou atures, click No. T be lost, click Help	ures that are not compatible with C5V (Comma delimited). Do you wan It any incompatible features, click Yes. Then save a copy in the latest Excel format.	t to keep the workbook in th

- 22. Open the C-more Programming Software and load your project or a new project
- Note: Project must use the Allen-Bradley EtherNet/IP Tag-Based driver
- 23. File > Import > Tag Name Database



THIS INFORMATION PROVIDED BY AUTOMATIONDIRECT.COM TECHNICAL SUPPORT

IS PROVIDED "AS IS" WITHOUT A GUARANTEE OF ANY KIND. These documents are provided by our technical support department to assist others. We do not guarantee that the data is suitable for your particular application, nor do we assume any responsibility for them in your application.

C	C-more Programming Software - [NewProject3.eap - #1 Scre											
C	File	Edit	View	Tool	Object	Screen	D	atabase	Setup	Pane	el Wind	
	Start Project Open Project Ctrl+O Close Project							Languag Languag Languag				
S	Sa Sa Sa	oject oject A: oject A:	s s Prote	cted	Ctrl+5							
	In	nport					۲	Tag Na	ame Data	abase.		
	Export •						 Message Database 			•		
	Si	Simulation						Address Book Language Edit				

24. Standard Import Settings

	File Path:	G:\IO Import Utility.csv						
Import	Option							
	verwrite Existing Ta	206	CSV/XLS Import	Option				
		-2-	From Direct	SOFT				
- - - - -	u Gla containa a Lla	adas Daus	- Device:	Default.				
	y file contains a me	auer now	001100.					
1	st Row is not impor	ted.		Default - Tags will be imported to the Device specified in the Import File 'DeviceName' Column.				
	A	В	Protocol					
1	TAG NAME	TAG DATATYPE						
2	ASCII STRING	ASCII_STRING	Append Text:					
3	BCD INT 16	BCD_INT_16						
4	BCD INT 32	BCD_INT_32	Import Array El	ement Count Limit(1-30000):				
5	DISCREATE	DISCRETE		0000				
6	FLOATING PT 32	FLOATING_PT_32		3000				

25. Your I/O and Alias Tags are now available in the C-more Tag Database for use with graphic display objects.



These documents are provided by our technical support department to assist others. We do not guarantee that the data is suitable for your particular application, nor do we assume any responsibility for them in your application.

Tag Nam Device I	ne Database Name: ALL	×			Fill Down
Displ	lay System Tags ── ┌☑] Highlight	Display Internal Tag	IS 🗌 Highlight Unused Tag	8	
No.	Tag Name	Data Type	PLC Address	Device Name	Attribute
1	LOCAL:1:I.DATA	Signed int 32	DINTLocal:1:I.Data	DEV001	
2	LOCAL:1:I.DATA.0	Discrete	BOOLLocal:1:I.Data.0	DEV001	
3	LOCAL:1:I.DATA.1	Discrete	BOOLLocal:1:I.Data.1	DEV001	
4	LOCAL:1:I.DATA.2	Discrete	BOOLLocal:1:I.Data.2	DEV001	
5	LOCAL:1:I.DATA.3	Discrete	BOOLLocal:1:I.Data.3	DEV001	
6	LOCAL:1:I.DATA.4	Discrete	BOOLLocal:1:I.Data.4	DEV001	
7	LOCAL:1:I.FAULT	Signed int 32	DINTLocal:1:I.Fault	DEV001	
8	MOTOR_STATUS_1	Discrete	BOOLMOTOR_STATUS_1	DEV001	
9	MUTUR_STATUS_2	Discrete	BUULMUTUR_STATUS_2	DEV001	
10	MUTUR_STATUS_3	Discrete	BUULMUTUR_STATUS_3	DEV001	
11	MUTUR_STATUS_4	Discrete	BUULMUTUR_STATUS_4	DEVOUI	
_					
1					
Add	Edit	Delete			
					Help

Not Supported: Only the PreDefined STRING type of 82 characters in length is supported, Custom Strings lengths are not supported. SINT[] Arrays can be used and they will be zero filled, the ".LEN" field won't automatically be updated by C-more as is the case when using the standard STRING data type.

c. Manual Entry

Any tag that is in the Control/Compact/FlegLogix PLC can be manually entered into the Cmore programming software. The easiest way to get tags into the C-more programming software is to use the import methods described above but if there are only a few tags added after the import has already been done, it may be easier and quicker to manually enter those tags.

6. Importing SLC, MicroLogix and PLC5 Symbols.

These series of processors use "Symbols" instead of "Tag Names". These symbols operate much like our Nicknames in DirectSoft. The Symbol is just a name associated with a memory address.



Within RSLogix500 the user is allowed to assign Symbols to Addresses and Symbols can also be organized into Symbol Groups. Additionally there is a Description field associated with each Symbol.

Note: The CSV exported from RSLogix will only contain elements which have either a Symbol Name or a Description field. If an address is used but is unassigned then it will not be listed in the CSV export. Upon Importing the C-more Tag Name will become the Symbol Name, if there is not a Symbol Name the Description Fields will be used as the C-more Tag Name. Name conflicts can't occur with Symbols but are possible with Descriptions; **the user should resolve these before importing**.

Another caveat of the exported symbols from RS Logix500 is that a symbol name can be assigned to an entire Timer, Control or Counter. The user can then use the 'symbol_name.member' in the instructions of their ladder code without having to explicitly defining a symbol name for each member. But in this situation the symbol name exported from RS Logix500 will be assigned to the entire Timer, Control or Counter address (T4:0, R6:0 or C5:0) and the C-more programming software cannot derive the data type of this symbol name. To help with this situation, there is a file called, "SLCandMicroLogixPreDefine.txt" in the main C-more programming software directory that can allow the user to choose which member of the data type the symbol name will be assigned to. To select which member the C-more programming software will import the symbol name to, simply add ",Default;" to the end of the member. As you can see below, the Control data type Symbol names will not be added by default.

📕 SLCandMicroLogix	PreDefine.txt - Notepad	
<u>File E</u> dit F <u>o</u> rmat <u>V</u> i	ew <u>H</u> elp	
DATATYPE COUNTE BOOL BOOL BOOL BOOL SINT SINT END_DATATYPE	R DN; CU; CD; OV; UN; ACC,Default; PRE;	<u>^</u>
DATATYPE CONTRO BOOL BOOL BOOL BOOL BOOL BOOL BOOL SINT SINT SINT SINT END_DATATYPE	L EN; DN; EU; EU; ER; UL; IN; FD; LEN; POS;	
DATATYPE TIMER BOOL BOOL SINT SINT END_DATATYPE	EN; TT; DN; ACC,Default; PRE;	

Note: This feature is not complete at this time. Unsupported file types (like Strings & Longs at present) should produce valid graceful warnings during import. The user will have to mirror any values stored in unsupported data files to other files which C-more does support (like Integers).



Step-by-Step, Exporting Symbols from RSLogix500 and Importing into Cmore:

1) Within RSLogix 500 and with your project file open, Select Tools>Database>ASCII Export

Tools Window Help								
Options	12 2 4 4 1 V 9 Q Q (
Delete Unused Memory		-0.> -0.0- ABL ABS						
Database	•	ASCII Export						
Security		ASCII Import Native Import						
Visual Basic								
Compare		Delete DB						
FactoryTalk Diagnostics		Delete Unused Addr. Edit Using Excel Edit Device Codes Adjust Rung Offset						
Add-In Manager								
		Convert Rung Attachment						
		Rebuild DB						



2) Import Tag Database into C-more Programming, Import>Tag Database>Select CSV File

Import Tag name Da	tabase from Exc	eVCSV F	ile					
File Path:	C:\SLC500 EXPORTED.CSV							
Import Option								
Overwrite Existing	Tags	ŝ	CSV/XLS Impor	: Option SOFT	A-B SLC	and MicroLog	jix 🛛	
My file contains a l	My file contains a Header Row		Device:	DEV001	× (Add		
All rows are import	All rows are imported.			Default - Tags will be imported to the Device specifi in the Import File 'DeviceName' Column.				
				Allen-Bradley DF1 SLC500 (Full Duplex)				
			Append Text:					
			Import Array El	ement Count Lin	nit(1-30000):			
				3000	2			
-		12						
				In	nport C	ancel	Help	



7. <u>Optimizing communications between C-more and Allen</u> <u>Bradley PLCs.</u>

In most situations, specific steps to optimize the communications between C-more and the PLC will not be necessary. Only in applications where very fast performance is desired or larger C-more projects on the bigger panels (12 inch or 15 inch) may necessitate more careful planning when designing the system.

In order to understand how to optimize the communications between C-more and the PLC, some fundamental knowledge of how and when communications are initiated is necessary:

Writing Data to the PLC: C-more writes data to the PLC infrequently in most applications.

- 1.) Screen objects only write to the PLC when triggered by the user, such as when a Pushbutton object is pressed or released, the enter key is pressed in a Numeric Entry object, the Slider object is moved, etc...
- 2.) The second way that C-more will issue a write to the PLC is through the Event manager "Tag Copy" function. This can be triggered by a value change of a tag, a screen change or time scheduled. See the Event manager functions for more details.
- 3.) The third way that C-more will write to the PLC is through the Panel To PLC tags. These typically only occur on a value change and happen infrequently most of the time.

All in all, writes occur rarely enough in most cases that they typically do not need to be considered when trying to optimize communications.

<u>*Reading Data from the PLC:*</u> There are several different functions that will cause C-more to issue read requests from the PLC.

- 1.) The most common are the objects on the current screen that is being displayed. The only screen objects that will issue read requests on screens other than the current screen being displayed are the trend objects. Most of the time, optimizing the PLC addresses assigned to objects on the same screen will provide the most impact to increased performance.
- 2.) Another function that will issue reads from the panel are alarm events and tag events. Any event created with a tag object addressed to the PLC will constantly read from the PLC no matter which screen is currently being displayed. Careful planning of the addresses used in events will greatly increase communications performance as well.
- 3.) The PLC to Panel tags will also be read constantly with no regard to which screen is currently being displayed. PLC addressing of these tags should also be considered.

a. Optimization for non-Tag based PLCs (SLC and MicroLogix).

The SLC and MicroLogix PLCs are fixed addressed PLCs. Grouping together consecutive addresses of the same data type in the same file without significant gaps between the addresses is the only way to optimize communications for these PLCs. Putting together tags of consecutive addresses on the same screen, in the event manager or in the PLC to Panel screen will greatly reduce the number of requested packets and will therefore greatly increase the amount of updates to the objects visible.



b. Optimization for Tag based PLCs (Control/Compact/FlexLogix PLCs).

The Control/Compact and FlexLogix PLCs are tag based PLCs. The method of communications (both serial and Ethernet) involves sending the actual symbolic tag name in the request sent over the wire. This means that the tag names themselves affect the speed of the communications by creating larger and/or multiple requests to the PLC. Program space tag names are not efficient. Each tag name needs the PROGRAM: header as well as the name of the program in front of the tag name. This makes the size of the request much, much larger hence requiring more requests to get multiple tags. Therefore, Controller scope tags should be used as much as possible.

Another method to reduce the size of the tag name requested without necessarily compromising the descriptive nature of a tag name is to use an Alias name in the C-more software. The alias name could be much shorter than the original tag name in the PLC.

The most effective method to reduce the number of requests from C-more to the PLC is to use arrays in the PLC and then assign the C-more tags to consecutive elements in the array of the PLC. C-more can then request multiple PLC tags with 1 request. This method will, by far, have the most impact in enhanced communication performance between C-more and the PLC.

8. Version History

Version 1.0: Release version.

Technical

Assistance: If you have questions regarding this Application Note, please contact us at 770-844-4200 for further assistance.