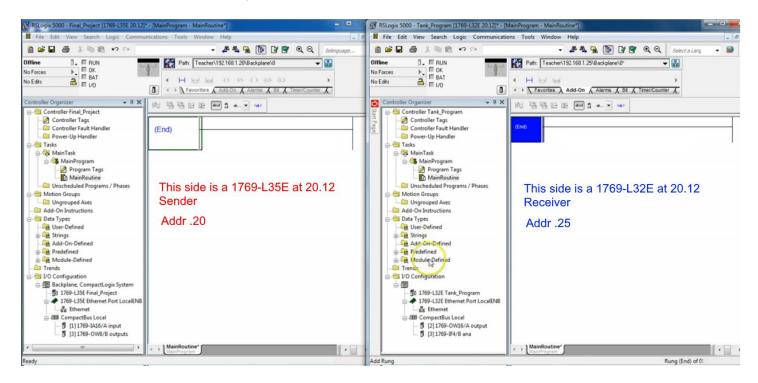
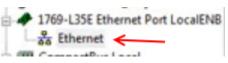
How to program a producer/consumer data messaging in RSLogix 5000 using Ethernet (X)

This is the basic method. There are many ways including message based. Some methods are very complicated. There is a more recent note on this topic in Databox.



First step is to set up the ethernet connection so they know where to look.

On the L35 side select "Ethernet" under the 1769-L35E Ethernet Port LocalENB rack symbol, and select New Module under right click menu.



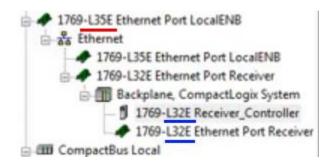
Search for and select the 1769-L32E Ethernet Port 10/100 Mbps ... from the catalog. Basically we are telling the PLC about the other PLC's communications setup.

alog Module Discovery Favor	tes			
Enter Search Text for Module 7	Dipe Clear Filters		Show Filters	¥
Catalog Number	Description	Vendor	Category	*
1769-L18ER-BB18 1769-L18ERM-BB18	CompactLogix 5318ER-BB1B Controller CompactLogix 5318ERM-BB1B Controller	Alen-Bradley Alen-Bradley	Controller Controller	
1769-L23E-QB1 Ethemet 1769-L23E-QBFC1 Ether 1769-L24ER-QB1B			Communication Communication Controller	
1769-L24ER-QBFC1B 1769-L27ERM-QBFC1B	CompactLogix5324ER-QBFC1B Controller CompactLogix5327ERM-QBFC1B Controller	Allen-Bradley Allen-Bradley	Controller Controller	
1769-L30ER 1769-L30ERM	CompactLogix5330ER Controller CompactLogix5330ERM Controller	Allen-Bradley Allen-Bradley	Controller Controller	
1269-L30ER-NSE 1769-L32E Ethemet Port	CompactLook5330ER-NSE Controller 10/100 Moos Ethernet Port on CompactLoop	- Alen-Bradey -5332E Alen-Bradey	Controller Communication	
1769-L33ER 1769-L33ERM 1769-L35E Rhemet Port	CompactLogx5333ER Controller CompactLogx5333ERM Controller 10/100 Moss Ethemet Part on CompactLogi	Alen-Bradley Alen-Bradley Alen-Bradley	Controller Controller Communication	-
209 of 209 Module Types Four	m d		Add to Favori	ites
Close on Create		Creste	Close	Help
1000 - 11100 - 100 - 100 - 100	and a second		-	
ct Major Revision	×			
Select major revision for ne Ethernet Port module being				
		en click Create and se	elect the mate	ching major revisi
Major Revision: 20	•			
OK Care	el Help			
	A A A A A A A A A A A A A A A A A A A			

In the next dialog we will name our new module. The name describes the other PLC which, in this case, is the receiver. We enter the IP address of the other PLC as well. We are telling the L35 what address to go out and look for.

Type: Vendor: Parent:	1769-L32E Ethernet Port 10/100 Mbps B Allen-Bradley LocalENB	Ethernet Port on Comp Address / Host Na	
Name: Description:	Receiver	 IP Address: Host Name: 	192 . 168 . 1 . 25
Slot:	1 Align Revision: 20		

On the L35 side we now have this...



Now we will do the same for the L32E PLC, we will register the L35's information.



All tags which are used to communicate between PLCs must be created and stored in the Controller Tags group under Controller.



We will create a tag called "Signal_to_be_sent", it is a **Produced** type tag, this will cause the data type to become DINT. Now click the **Connection** button and confirm the entry for Max Consumers is set to 1.

Jsage:	<normab< th=""><th></th><th></th></normab<>		
Туре:	Produced	Conection	
Alias For:		~	-
Data Tuper	DINT	-	

Connection	Status	1				
Max Consu	mers:	8			Advanc	ed
Send D	ata State	Chan	ge Ever	t To Cons	sumer(s)	

connection State	us	
O Connection S	Status Included	
(i) Connect	tion Status is inclu	ded when the tags data type is a user defined data
Connect type who	tion Status is inclu ose first member is	ded when the tags data type is a user defined data CONNECTION STATUS.
Connect type who Data Type:		ded when the tags data type is a user defined data CONNECTION STATUS.

Now we create a simple program with the contact being a local input and the coil being on the other plc. But first we must specify which bit inside the data type DINT we wish to send. We can only send 1 bit.

	e e	PB1 <local:11.data.0></local:11.data.0>	S	igna	al_t	o_b	e_s	Sent	•	•		î.		
0	e e		V	r. En	ier A	(ante)	(ila:		•	5	how	All Tags		•
				Nar	me							Data Type	Description	-
(End)			0000		Loca	ni: 1:1 ni: 3:C ni: 3:1 ni: 3:0						AB:1769_DI AB:1769_DI AB:1769_DI AB:1769_DI BOOL BOOL))	10
			10		-	a to	be !	Sent				DINT		di i
				6	R	1	2	3	4	5	6	7		Ч
			_	_	_	Used			2	13	14	15		-
						the second second	-	d: Y*	2	21	22	23		
				- 2	24	25	26	27 2	8	29	30	31		

Once we select the bit we see it represented in the tag name above the coil.

0	PB1 <local:1:i.data.0></local:1:i.data.0>	Signal to be Sen 0
(End)		

Above is the Producer side.

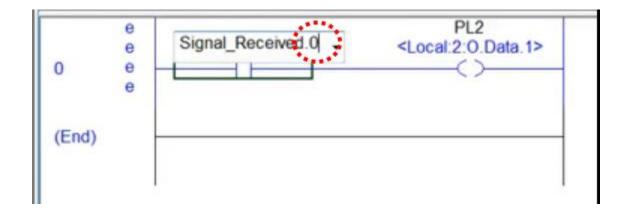
On the consumer (receiver) side we will set up a similar arrangement, but the contactor will be a bit we send from L35 (this is likely the reason we had to select a single bit above). The coil will be a local physical output. Other data types can likely receive more than on bit. Under Controller Tags create a new tag called Signal_Received of type Consumed then click Connection and select the Producer item "Sender_Controller" and then enter the tag name of the tag sending the data on the Sender side.

Connection 9	tatus			
Producer:	Sender_Co	ontroller	•	
Remote Data:	Signal_to_	be_Sent		
	(Tag Name	or Instance Numb	er)	
RPI:	20.0	🔹 ms		
🕢 Use Unica	st Connectio	n over EtherNet/IF	6	

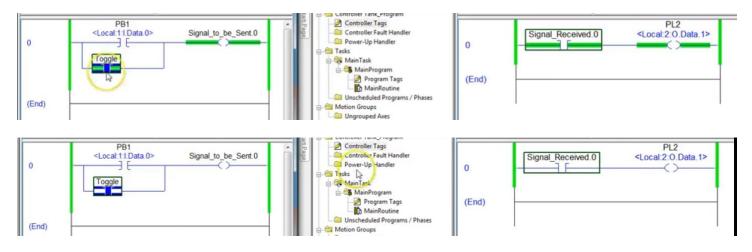
lame:	Signal_Received		Conte -
Description		*	Cancel
		-	nep
Usage:	cnormab	Ψ	
Туре:	Consumed • Co	onnection	
	Consumed - Co	onnection	
Alias For:	Consumed • Co	w	
Type: Alias For: Data Type: Scope:		•	
Alias For Data Type:	DINT	•	

Make sure you specify the proper bit on the contactor tag name.

Τ	Sig	nal_	Re	ceiv	ved	•			<	Local	PL2 2:0.Data.1	>
t	V. (7/2/7/	Vətan	1736	P		-	how	ALI	ag:	-	•
	1	lame						:=	Data	Туре	Description	-
	1 8	-Loc	al2	¢					AB:	1769_DO	les .	11
t	10	-Loc	al2	1					AB:	1769_DO		
	11 9	Loc	at2	0					AB:1	769_DO		
L	1 9	-Loc	al 3	¢					AB:	769_IF4		=
	1 1	-Loc	al3	1					AB:	769_IF4	have a second	11
		PL2	2						800	IL.		Ш
	9	Sig	na. F	Recei	ved				DINT			1
		Q	1	2	3	4	5	6	7			μ
		20	Us.	d: N		12	13	14	15			
		16	DAN	IT Us	ied: \	10	21	22	23			
		24	25	26	27	28	29	30	31			



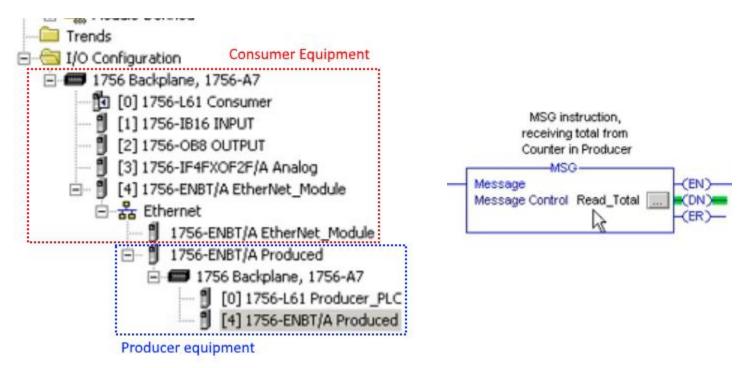
Now download each program to its respective controller. Put the controllers in remote run. Run the test cases.



Use a message command to send data of larger types. Data going the other way works the same.

How to set up a Producer/Consumer Message Exchange (X)

How to set up a message instruction on a ControlLogix platform? The machine we are working on is the Consumer. The second machine is referred to as Produce in the hardware tree.



Message instructions must have a name, the example above is named Read_Total. Keep in mind this name is not a tag, it, the name, does not transmit or receive data. Now click the eclipse to bring up the configuration dialog box.

Configuration Tab

Message Type:	CIP Data Table Read		-	
Source Element:	Counter_1.ACC)		
Number Of Elements:	1			
Destination Element	Total_ACC_Count	- 1		New Tag
	Tracecourt	/		tow rag.
·	e Waiting 🔾 Start	• Done	Done Length: 1	

Message Type: tells the application what type of equipment you are connecting to, offers generic as well.

Source Element: tag name on the other machine you wish to link to. You can include the dot attributes as well, for ex. .ACC.

Number of Elements: size of the data table.

Destination Element: tag name of destination tag (on same PC).

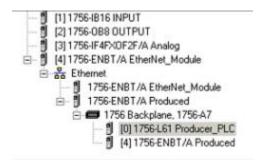
Red is the local tag which will receive the data acquired from the other computer on the network.

Blue is the tag on the other networked computer that we will read, we will transfer its data from the other computer to our local computer and the tag Total_ACC_Count will receive that data.

Communications Tab

Path: Producer_PLC Producer_PLC			Browse
Communication Metho	Channel	Destination	
Connected	I⊄ Cache	Connections +	
Enable 🔾 Enable	Waiting OStart	🔹 Done	Done Length: 1

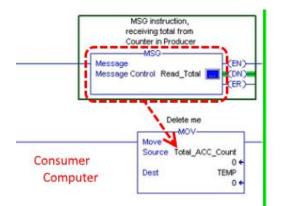
Path: brings up the network tree so you can select the other hardware. In this case we are selecting the **Producer** PLC.



<u>Tag Tab</u>

essage Con	iguration - Read_1	otal			×
Configuration	Communication T	ag			
Name:	Read_Total				
Description:	rece	G instruction, eiving total from nter in Producer	<u>×</u>		
Time	Base		<u>*</u>	l	
Type: Data Type:	MESSAGE				
Scope:	Consumer				
) Enable	O Enable Waiting	 Start 	🔵 Done	Done Length: 1	
) Error Cox Error Path: Error Text:	Extend	ed Error Code:		Timed Out •	
		OK	Cancel	Apply	Help

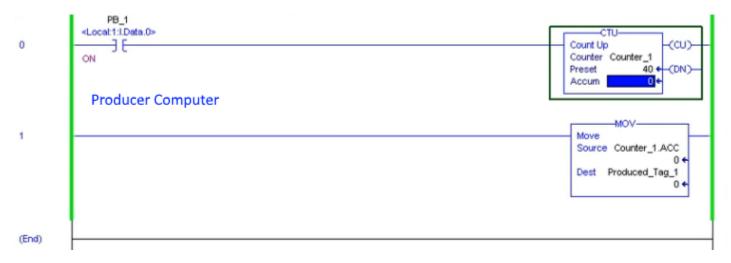
Name: the name given to the IO block. This is not the data tag!



In this example our MSG FB will retrieve the data on the other machine

With the other machine set up appropriately (and with a tag named Counter_1.ACC, the data will be extracted, moved into Total_ACC_Count, and the transferred to TEMP.

Code from Producer computer:



Producer_PLC Communication Method CIP C DH+ Channel Destination Link:	onliguration Communicatio	m Tag		_
Communication Method C DP C DH+ Channel Destination Link: 0 = C DP With Source Link: 0 = Destination Node 0 = (Oct Source ID Source Link: 0 = Destination Node 0 = (Oct Connected Cache Connections + Enable © Enable Waiting © Start © Done Done Length: 1 Error Code: Extended Error Code: Timed Out + or Path:	Path: Producer_PLC			Browse
Image: Construction Constructicon Construction Construction Construction Const				
Connected Cache Connections C		nnel	Destination Lini	0 =
Enable I Enable Waiting I Start I Done Done Length: 1 Error Code: Timed Out + or Path:	€ CIP W/th Source ID Sou	rceLink 🛛	Destination Not	ie 🛛 🛨 (Octa
Error Code: Extended Error Code: Timed Out + or Path:		122233 03		
Error Code: Extended Error Code: Timed Out + or Path:	Connected	Cache	Connections 🔶	
Error Code: Extended Error Code: Timed Out + or Path:	Connected	Cache	Connections 🔶	
or Path:				
			🔮 Done 🛛 🛛	
rex. Error messages win appear in this area.	Enable 🔾 Enable Wait Error Code: E	ting ② Start	🔮 Done 🛛 🛛	
	Enable ③ Enable Wait Error Code: E or Path:	ting O Start Extended Error Code:	🔮 Done 🛛	Timed Out +