

# ***460 Triggering User Guide***

---

*Firmware Version 6.2.3*

## Trademarks

CompactLogix, ControlLogix, & PLC-5 are registered trademarks of Rockwell Automation, Inc. EtherNet/IP is a trademark of the ODVA. MicroLogix, RSLogix 500, and SLC are trademarks of Rockwell Automation, Inc. Microsoft, Windows, and Internet Explorer are registered trademarks of Microsoft Corporation. BACnet® is a registered trademark of American Society of Heating, Refrigerating and Air-Conditioning Engineers (ASHRAE). All other trademarks and registered trademarks are the property of their holders.

## Limited Warranty

Real Time Automation, Inc. warrants that this product is free from defects and functions properly.

EXCEPT AS SPECIFICALLY SET FORTH ABOVE, REAL TIME AUTOMATION, INC. DISCLAIMS ALL OTHER WARRANTIES, BOTH EXPRESSED AND IMPLIED, INCLUDING BUT NOT LIMITED TO IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR APPLICATION. THIS LIMITED WARRANTY GIVES YOU SPECIFIC LEGAL RIGHTS. YOU MAY ALSO HAVE OTHER RIGHTS, WHICH VARY FROM STATE TO STATE.

The examples and diagrams in this manual are included solely for illustrative purposes. Because of the many variables and requirements associated with any particular application, Real Time Automation, Inc. cannot assume responsibility or liability for actual use based on the examples and diagrams. Except as specifically set forth above, Real Time Automation and its distributors and dealers will in no event be liable for any damages whatsoever, either direct or indirect, including but not limited to loss of business profits, income, or use of data. Some states do not allow exclusion or limitation of incidental or consequential damages; therefore, the limitations set forth in this agreement may not apply to you.

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

## Government End-Users

If this software is acquired by or on behalf of a unit or agency of the United States Government, this provision applies: The software (a) was developed at private expense, is existing computer software, and was not developed with government funds; (b) is a trade secret of Real Time Automation, Inc. for all purposes of the Freedom of Information Act; (c) is "restricted computer software" submitted with restricted rights in accordance with subparagraphs (a) through (d) of the Commercial "Computer Software-Restricted Rights" clause at 52.227-19 and its successors; (d) in all respects is proprietary data belonging solely to Real Time Automation, Inc.; (e) is unpublished and all rights are reserved under copyright laws of the United States. For units of the Department of Defense (DoD), this software is licensed only with "Restricted Rights": as that term is defined in the DoD Supplement of the Federal Acquisition Regulation 52.227-7013 (c) (1) (ii), rights in Technical Data and Computer Software and its successors, and: Use, duplication, or disclosures is subject to restrictions as set forth in subdivision (c) (1) (ii) of the Rights in Technical Data and Computer Software clause at 52.227-7013. If this software was acquired under GSA schedule, the U.S. Government has agreed to refrain from changing or removing any insignia or lettering from the Software or documentation that is provided or from producing copies of the manual or media. Real Time Automation, Inc.

© 2018 Real Time Automation, Inc. All rights reserved.

## Overview of Optimized Triggering

The 460 products now allow for the end device to know that all the data from the source has been successfully updated in the application. Without this feature, you would get some of the updated data without knowing that all or some of the data has been updated.

This feature is only available in certain protocols so please be aware. The Allen-Bradley PLC protocol (“ETC”) and the DF1 Master protocol (“DFM” – coming soon) are the two protocols that currently support this optimization feature.

## Overview of How the Allen-Bradley PLC (“ETC”) Triggering Works

Within the Allen-Bradley PLC configuration page, you will now find a configurable tag called the “Optimization Trigger Tag/File Name”. The Tag (Logix PLCs) or File (Non-Logix PLCs) that you specify here MUST be configured as either an INT Tag or a N File.

Once a connection is made to the PLC, the Trigger Tag/File is constantly being read checking for a change of state. Once the Trigger Tag/File is different, then the read scan lines will be executed and after each successful request, the data will be marked valid. Once all of the data has been marked valid, then the ETC’s handshake value will be set equal to the Trigger value.


The ETC’s handshake value should be mapped to the mating protocol for it to make use of and to know that all of the data has been updated.

Note: by default, the 460 gateways will cyclically read the Tag/Files defined. Once the trigger Tag/File is defined, the 460 gateways is only cyclically read the trigger Tag/File and only read the Scan Lines defined once the trigger tag/file has changed values.

## Let’s Look at an Example

First, setup the Allen-Bradley PLC Optimization Tag/File. Reminder that this needs to be defined an INT Tag or a N File and NOT part of the Read or Write Scan Lines. In the below example, we are setting up the “trigger\_tag” as the optimization trigger tag.

<input checked="" type="checkbox"/> Enable	<b>Allen-Bradley PLC 1</b>	
Device Label <input type="text" value="ETC01"/>	IP Address <input type="text" value="10.1.100.99"/>	
Controller Slot <input type="text" value="0"/> 0-49	PLC Type <input type="text" value="CompactLogix"/>	Update Type <input type="text"/>
Comms Mode <input type="text" value="Connected (Class 3 Explicit)"/>		
# of Read Scan Lines <input type="text" value="6"/> 0-150	# of Write Scan Lines <input type="text" value="0"/> 0-150	
Optimized Trigger Tag/File Name (16-Bit Int)	<input type="text" value="trigger_tag"/>	
<input type="button" value="Generate Scan Lines"/>		




Second, setup the Read Scan Lines for the data that you would like transferred to the mating protocol.

Third, configure the mating protocol, in the below example, it is the Web Interface (WI) protocol acting as a server. Setup the data points that you require to be moved over and then add an additional point of data. This additional point of data will be used to tell its protocol that all of the data has been updated.

In the below image, the HandShake data point is going to be used to let its Client know that the data has been updated.

<input checked="" type="checkbox"/>	5	G2N0005	INT (32-bit)	DEV01	GROUP01
<input checked="" type="checkbox"/>	6	G2N0006	INT (32-bit)	DEV01	GROUP01
<input checked="" type="checkbox"/>	7	HandShake	UINT (16-bit)	DEV01	GROUP01

<< 1 >>  
 View



Forth, configure the mating protocol, in the below example, it is the Web Interface (WI) protocol acting as a client. On the WI protocol there is a dropdown for Triggered therefore the HandShake data point described above is not needed.

Update Method	Triggered
---------------	-----------

## Mapping Configuration to Support Triggering

For the ETC protocol optimization and the mating protocol handshaking to be setup properly, you will need to follow the below instructions to do so.

First, make sure that all your data that has been setup is mapped in both directions, excluding the data points that have set aside as the Handshake data points. Once that has been confirmed, please make sure that the Auto-Configuring option is set to “Manual Configure”. To do this, go the “Display Data” page and click on the “Edit Mapping” button. Also accessible from the left-hand navigation called “Data Mapping”. Then, go ahead and verify that it says, “Manual Configure”.

Second, then add two mappings. These two additional mappings are to be used for the Triggering as well as the Handshake data points.

**Data Mapping Configuration**
Help

Manual Configure

# of Mappings to Configure: 8 0-1000


Set Max # of Mappings

<< 1 >>


Navigate to the second to last mapping that is available, enable it, and in the Source Group dropdown, select the Trigger tag that was configured in the Optimized Trigger data point configured within the protocol. In the Destination Group dropdown, select the ETC “Trigger 0” and then in the Start dropdown menu, select the correct offset.

**Trigger 0:** used if you want to have all devices configured sync'd together. As in, if you have 3 PLC's configured in the Allen-Bradley PLC page and have a single data point mark all of the data invalid across all the PLC's.

**Trigger 1...32:** used if you want to trigger individual PLC's. If you want to trigger PLC 1, then you will map the data to Trigger 1. If you want to trigger PLC 3, then you will map the data to Trigger 3.


Mapping 7		
<input checked="" type="checkbox"/> Enable		
Source	<input type="checkbox"/> Enable Manipulation	Destination
Group: ETC01 trigger_tag (Int16) ▼ Start: trigger_tag ▼ End: trigger_tag ▼		Group: ETC Trigger 0 (UInt16) ▼ Start: Trigger 1 ▼ End: Trigger 1

Mapping 8		
<input checked="" type="checkbox"/> Enable		
Source	<input type="checkbox"/> Enable Manipulation	Destination
Group: ETC Handshake 0 (UInt16) ▼ Start: Handshake 1 ▼ End: Handshake 1 ▼		Group: DEV01 HandShake (UInt16) ▼ Start: HandShake ▼ End: HandShake

Navigate to the last mapping that is available, enable it, and in the Source Group dropdown, select the Handshake tag that corresponds to the protocol that is using the Optimized Trigger data point configured within the protocol. In the Destination Group dropdown, select the mating protocols data point that was configured, in the above example, it is the Web Interface (WI) protocol acting as a server, it was the WI's HandShake data point.

In the below example, it is the Web Interface (WI) protocol acting as a client. You would map handshake 1 to WI Upload Trigger.

Mapping 9		
<input checked="" type="checkbox"/> Enable		
Source	<input type="checkbox"/> Enable Manipulation	Destination
Group: ETC Handshake 0 (UInt16) ▼ Start: Handshake 1 ▼ End: Handshake 1 ▼		Group: WI Upload Trigger (UInt16) ▼ Start: ▼ End:

## Overall Behavior

The following steps is a summary to what has been discussed in the above sections:

1. Setup
  - a. Configure an Optimized Trigger data point
  - b. Configure the Read Scan Lines

- c. Configure the Mating protocol's data points
  - d. Add the Mating protocol's data point for the Handshake from the other protocol
  - e. Map all data points together except for the Handshake data point
  - f. Manually add two mappings
  - g. Configure the first manually added mappings to be Triggering
  - h. Configure the second manually added mappings to be Handshake or WI Upload Trigger (WI Client setup)
2. Logic
- a. Update ETC's data
  - b. Change the Trigger value
  - c. Within the Mating protocol, WI Client, monitor for the Handshake data point to change values.
  - d. Once Handshake changes values, all other data has been successfully updated in the data table