Industrial Automation

Tech Note 12 Mapping to RSLogix 500 PLCs

Abstract:

This document describes how to map Crimson® tags to RSLogix 500 based processors.

Products:

G3 Series HMI / G3 Kadet HMI / Graphite® HMI / Graphite® Controllers / Modular Controller / Data Station Plus / ProducTVity Station

Problem Solved: Mapping tags to RSLogix 500 PLCs

One difference that can be applied to the majority of the available data types is the way that bit addressing is displayed. RSLogix 500 often uses a "/" to signify a bit within a word or long, Crimson uses a "." to signify a bit within a word or long. All RSLogix 500 based PLCs transfer data in increments of 16 bit words, in order to map a tag to a bit within a word or long, first map the tag to the word or long, and then set the Treat As property to Bit Array Little-Endian and then choose the bit from the Bit Number drop down selection. Refer to section 2.B. in the Crimson 3.0 Quick Start Guide for further instructions on accessing bits within words.

Required Software:
Crimson 2.0 or 3.0

Binary

RSLogix 500

Binary Data File addresses may be shown in one of two formats:

1. B(File):Element/Bit
2. B(File)/Bit.

Crimson

Crimson's mapping is similar to option 1 listed above. The Binary Data Files are transferred a word at a time, in order to access individual bits, a flag tag is required.

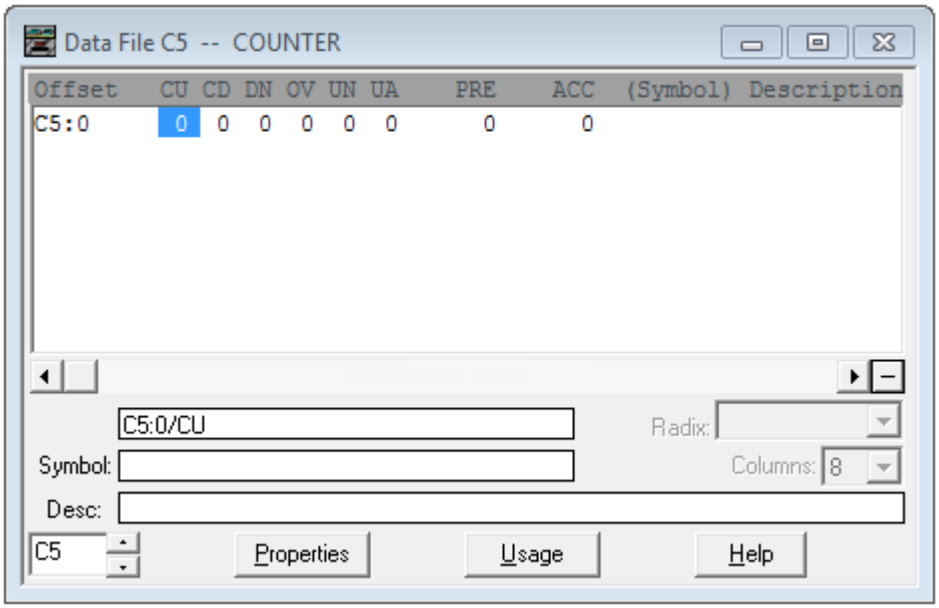
Counter

RSLogix 500

Counter Data File addresses are shown in the following format:

- C(File):Element
- Counters are broken down into 8 pieces:
  - CU
  - CD
  - DN
  - OV
  - UN
  - UA
  - PRE
  - ACC.

- The item selected below is displayed as C5:0/CU in RSLogix 500.



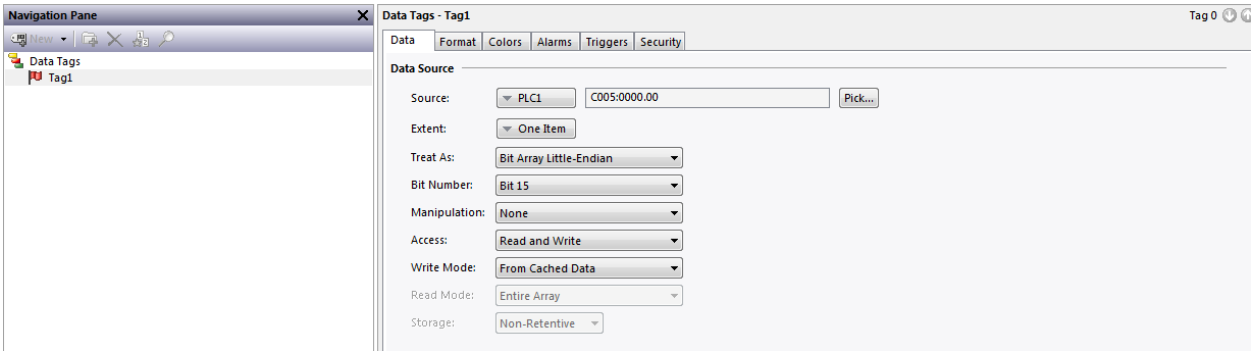
Crimson

Crimson mappings are slightly different that RSLogix 500. It uses the following format:

- C(File):Element, this is the same as RSLogix 500
- Counters are broken down into 3 pieces:
  - 0 – STAT Status word
  - 1 – PRS Counter Preset
  - 2 – ACC Counter Accumulated Value.

- To map a tag to the count up enable bit above (C5:0/CU in RSLogix 500) in Crimson you would map a Flag tag to C005:0000 0 - STAT. Set the Treat As to *Bit Array Little-Endian* and choose Bit 15.

  NOTE: The bit numbers are explained in the RSLogix 500 Instruction Set Help.



Floating Point

RSLogix 500

Floating Point Data File addresses are in the following format:

- F(File):Element, ex. F8:3

Crimson

Crimson mappings are similar, but with leading 0s. ex. F008:0003

Input

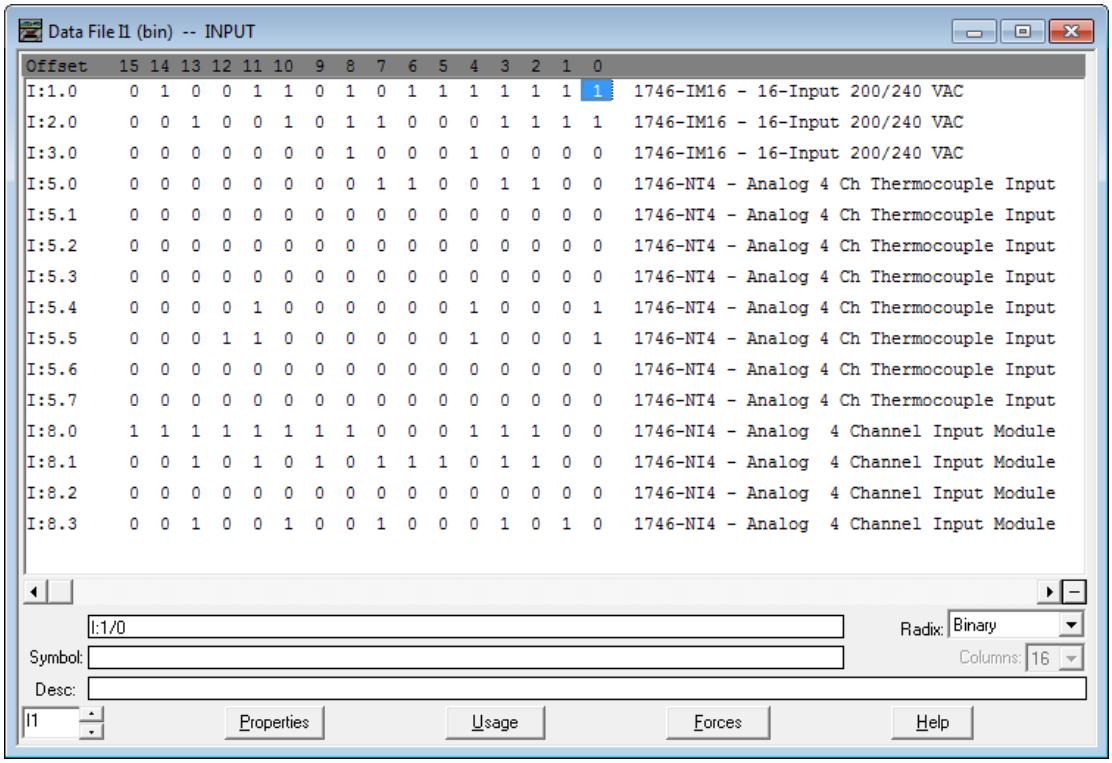
RSLogix 500

Input Point Data File addresses are in the following format:

- I:(Slot).Word/Bit
  - I is for Input
  - Slot is the backplane location of the input module
  - Word is which word of the card contains the data.

    NOTE: RSLogix 500 uses a "/" to signify a bit within a word or long.

- The item selected below is displayed as I:1.0/0 in RSLogix 500.



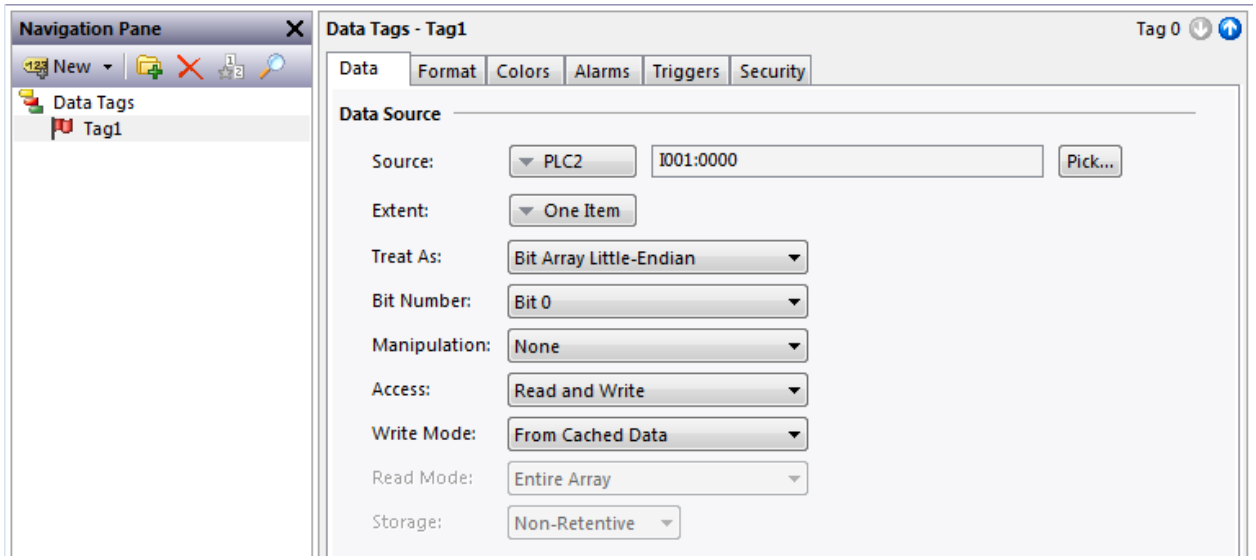
Crimson

Crimson mappings are slightly different than RSLogix 500, it uses the following format:

- I(Slot):Word.Bit
  - I is for Input
  - Slot is the backplane location of the input module
  - Word is which word of the card contains the data.

    NOTE: Crimson uses a "." to signify a bit within a word or long.

- To map a tag to the input above (I:1.0/0 in RSLogix 500) in Crimson you would map a tag to I001:0000. Then, set the Treat As to *Bit Array Little-Endian* and choosing Bit 0.



Long

RSLogix 500

Long Data File addresses are in the following format:

- L(File):Element, ex. L9:7

Crimson

Crimson mappings are similar, but with leading 0s. ex. L009:0007

Integer

RSLogix 500

Integer Data File addresses are in the following format:

- N(File):Element, ex. N7:0

Crimson

Crimson mappings are similar, but with leading 0s. ex. N007:0000

Output

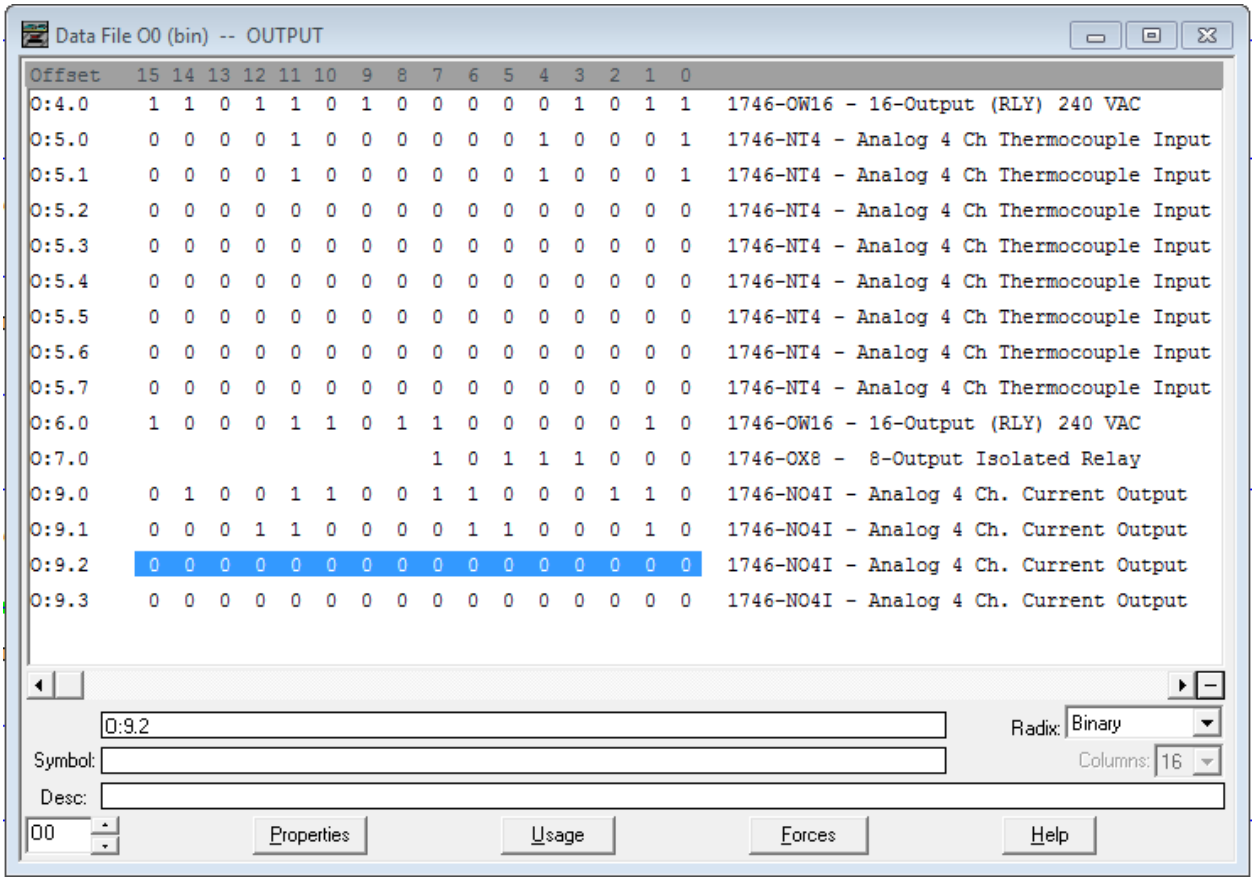
RSLogix 500

Output Point Data File addresses are in the following format:

- O:(Slot):Word/Bit
  - O is for Output
  - Slot is the backplane location of the input module
  - Word is which word of the card contains the data.

    NOTE: RSLogix 500 uses a "/" to signify a bit within a word or long.

- The item selected below is displayed as O:9.2 in RSLogix 500.



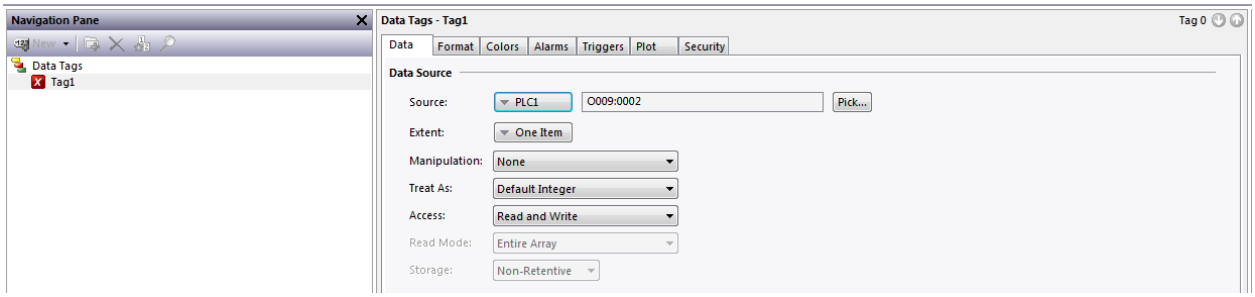
Crimson

Crimson's mapping is slightly different than RSLogix 500, it uses the following format:

- O(Slot):Word.Bit
  - O is for Output
  - Slot is the backplane location of the input module
  - Word is which word of the card contains the data.

    NOTE: Crimson uses a "." to signify a bit within a word or long.

- To map a tag to the input above (O:9.2 in RSLogix 500) in Crimson you would map a tag to O009:0002.



String

RSLogix 500

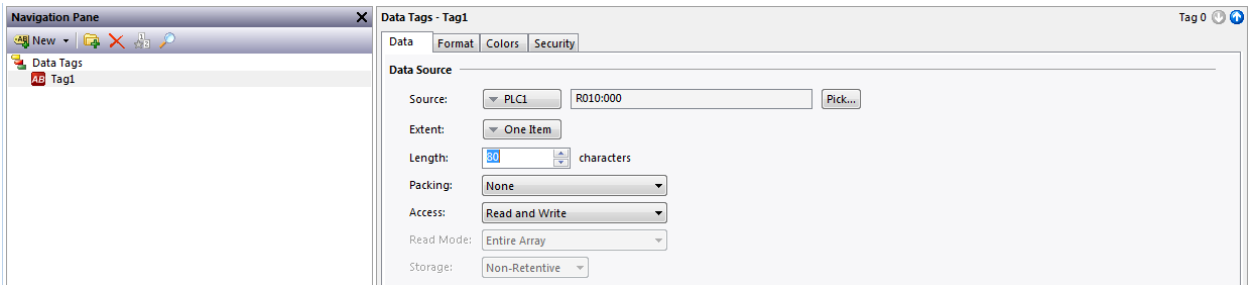
String Point Data File addresses are in the following format:

- ST(File):Element ex. ST10:0
- Strings are broken down into 2 pieces
  - LEN
  - String Text.

Crimson

Crimson only has a single character space available, so strings are referred to as R registers in Crimson; the File and Element values also have leading zeros. ex. R010:0000

- After mapping the Crimson tag to the String register, the Length parameter MUST be set to *80* characters.



Status

RSLogix 500

Status Data File addresses are in the following format:

- S:Element, ex. S:42 (Clock Calendar Seconds)

Crimson

Crimson mappings are similar, but with leading 0s. ex. S:0042

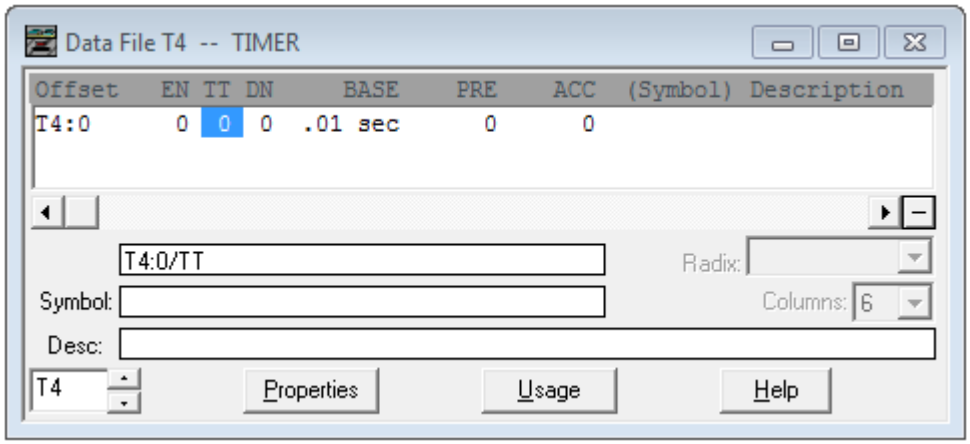
Timer

RSLogix 500

Timer Data File addresses are in the following format:

- T(File):Element ex. T4:0
- Timers are broken down into 6 pieces:
  - EN
  - TT
  - DN
  - BASE
  - PRE
  - ACC.

- The item selected below is displayed as T4:0/TT in RSLogix 500.

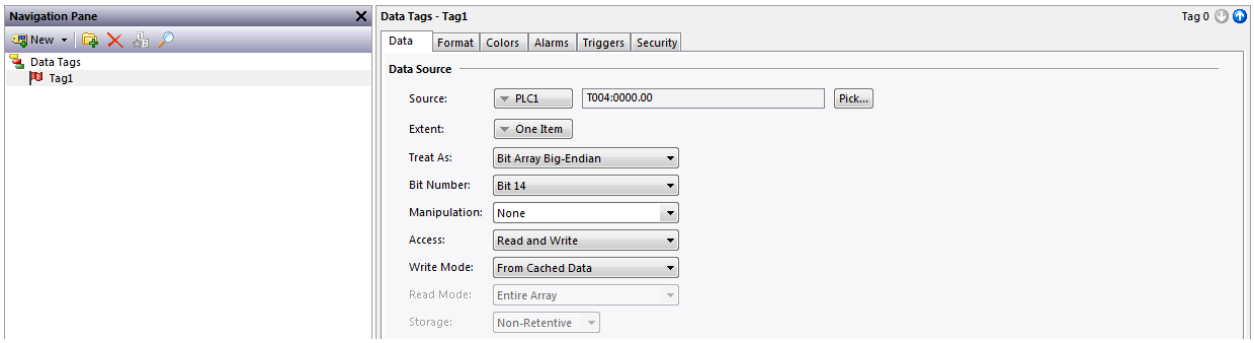


Crimson

Crimson's mapping is slightly different that RSLogix 500, it uses the following format:

- T(File):Element, this is the same as RSLogix 500
- Timers are broken down into 3 pieces:
  - 0 – STAT Status word
  - 1 – PRS Timer Preset
- 2 – ACC Timer Accumulated Value.To map a tag to the count up enable bit above (T4:0/TT in RSLogix 500) in Crimson you would map a Flag tag to T004:0000 0 - STAT. Set the Treat As to *Bit Array Little-Endian* and choose Bit 14.

NOTE: The bit numbers are explained in the RSLogix 500 Instruction Set Help.



For more information: http://www.redlion.net/support/policies-statements/warranty-statement