Allen-Bradley

Create Tag Descriptions Automatically with User-Defined Data Types

Document a Logix5000 project in significantly less time by using user-defined data types combined with the pass-through of descriptions.

Allen-Bradley
RELIANCE ELECTRIC
DODGE
ROCKWELL SOFTWARE

Rockwell Automation

Automate Tag Descriptions Using User-Defined Data Types

Tag-Based Memory Makes Projects Self-Documenting

The Allen Bradley Logix5000™ controller's tag-based memory provides an evolution in the development of the Programmable Logic Controller (PLC). The use of appropriately named variables rather than cryptic physical addresses makes application programs self-documenting and thus easier to create and maintain.

- RSLogix™ 5000 software, the programming software for the Logix5000 controllers, lets you create up to a 40-character name for each variable (tag). This lets the tag's name be fairly descriptive of how the tag is used in the application.

- User-defined data types (structures) let you create customized memory records that consolidate multiple fields (members) of data into a single contiguous group with a hierarchical lay-out.

Descriptions Clarify Tag Names

Even with these premier capabilities, the purpose of a tag can still be ambiguous. By adding a description to a tag, RSLogix 5000 software lets you add another 120 characters of information.
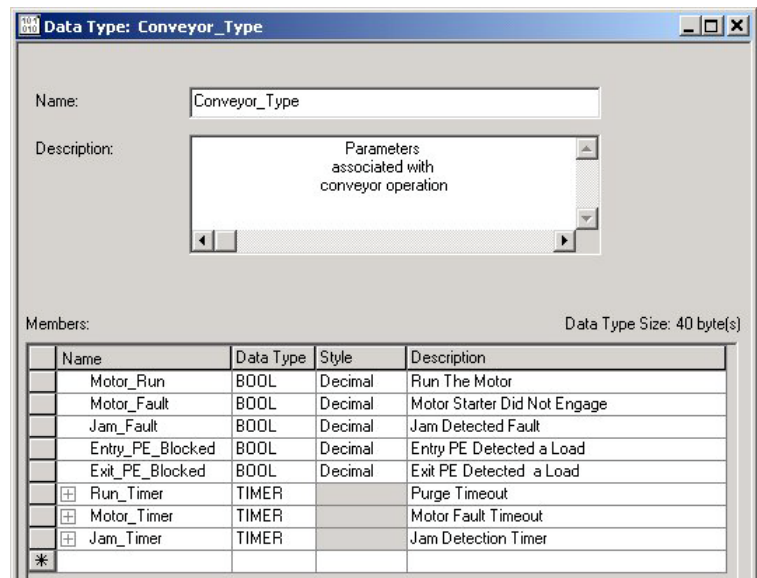
- Because you have to manually enter descriptions, they are often left blank. This can lead to additional expense during startup or when maintaining the system.

- Starting with version 13, RSLogix 5000 software automates the creation of tag descriptions, thus improving the application while minimizing the associated development cost.

Structures Include a Description of Each Member

RSLogix 5000 software, like most programming software, lets you assign both a name and description for each member within a user-defined data type.
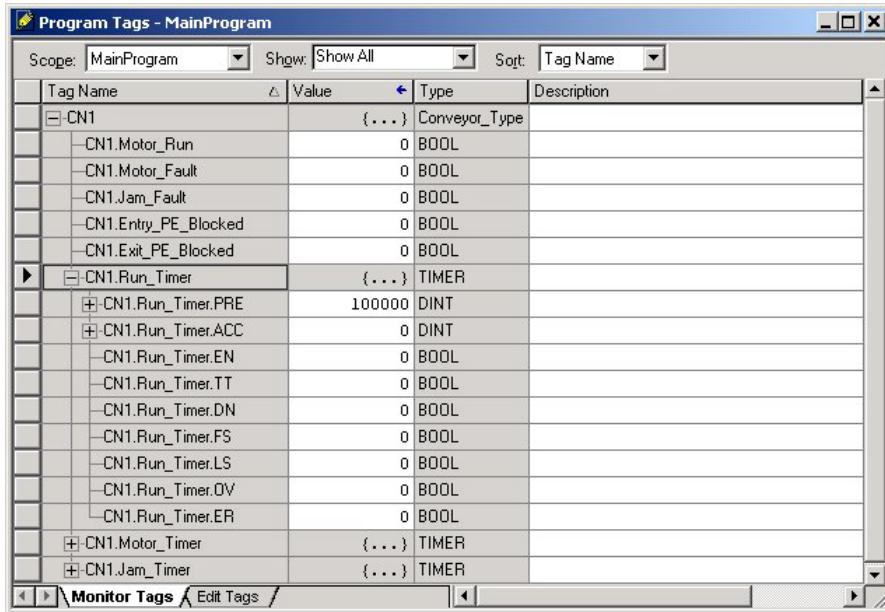
Member name - The program uses the member name to access the associated data.

Member description - The description helps to define the purpose of the member.

Data Type: Conveyor_Type

Name: Conveyor_Type

Description: Parameters associated with conveyor operation

Members: Data Type Size: 40 byte(s)

Name	Data Type	Style	Description
Motor_Run	BOOL	Decimal	Run The Motor
Motor_Fault	BOOL	Decimal	Motor Starter Did Not Engage
Jam_Fault	BOOL	Decimal	Jam Detected Fault
Entry_PE_Blocked	BOOL	Decimal	Entry PE Detected a Load
Exit_PE_Blocked	BOOL	Decimal	Exit PE Detected a Load
⊞ Run_Timer	TIMER		Purge Timeout
⊞ Motor_Timer	TIMER		Motor Fault Timeout
⊞ Jam_Timer	TIMER		Jam Detection Timer
✳			

With other development software, you see the description for a member only if you look at the definition of the user-defined data type; the descriptions don't carry through to any tags that are based on the user-defined data type.
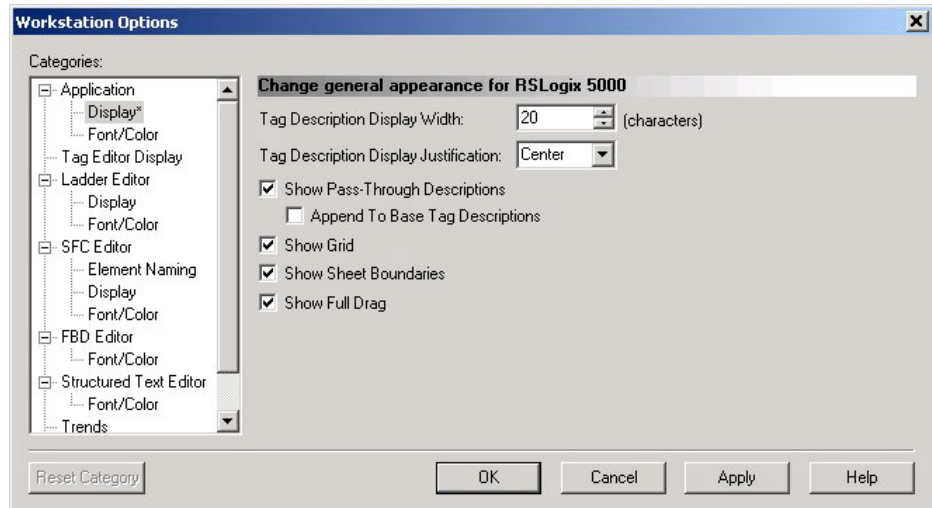

To get specific documentation for a tag, each of its members requires its own description.

In most cases, the descriptions you enter for a tag's members are based on the descriptions that you already entered in the user-defined data type. This means that each time you create a tag that uses the user-defined data type, you must manually enter an almost identical description for each member of the tag.

**RSLogix 5000
Software Reduces the
Documentation Time**

RSLogix 5000 software now differs from other software in the industry because it lets you reuse documentation work you have already done. The new *Show Pass-Through Descriptions* feature dramatically reduces your documentation and development time while improving your documentation.

When turned on, the *Show Pass-Through Descriptions* feature lets RSLogix 5000 software look within a tag and/or its associated user-defined data type for an available description.

- The software looks for the description that most likely is the best fit for the tag member.

- Depending on how the tag is defined, the software pulls the description from the root portion of the tag or from the appropriate member of its user-defined data type.

- Once found, the software automatically shows the description in the tag window and the logic (i.e., the description passes through to the tag).

In the following example, the descriptions for the Conveyor_Type data type (shown earlier) and each of its members pass through to a tag that uses that data type. Both the tag monitor window and the logic editor show the pass-through descriptions.

MainProgram - MainRoutine

Conveyor CN1 Entry
Photo Eye Input
CN1_PE1
<Local:1:I.Data.2>

4

Conveyor CN1 Exit

Jam Detection Timer

TON
Timer On Delay   —(EN)—
Timer   CN1.Jam_Timer   —(DN)—
Preset   5000 ←
Accum   0 ←

Jam Detection Timer   Jam Detected Fault
CN1.Jam_Timer.DN   CN1.Jam_Fault
       ⊣ ⊢   —(L)—

Conveyor CN1 Motor
Starter Output
CN1_M
<Local:2:O.Data.0>

Jam Detected Fault
CN1.Jam_Fault
     ⊣/⊢   —( )—

Program Tags - MainProgram

Scope: MainProgram   Show: Show All   Sort: Tag Name

Tag Name	Value	Type	Description
⊟-CN1	{...}	Conveyor_Type	Parameters associated with conveyor operation
—CN1.Motor_Run	0	BOOL	Run The Motor
—CN1.Motor_Fault	0	BOOL	Motor Starter Did Not Engage
—CN1.Jam_Fault	0	BOOL	Jam Detected Fault
—CN1.Entry_PE_Blocked	0	BOOL	Entry PE Detected a Load
—CN1.Exit_PE_Blocked	0	BOOL	Exit PE Detected a Load
⊟-CN1.Run_Timer	{...}	TIMER	Purge Timeout
⊞-CN1.Run_Timer.PRE	100000	DINT	Purge Timeout
⊞-CN1.Run_Timer.ACC	0	DINT	Purge Timeout
—CN1.Run_Timer.EN	0	BOOL	Purge Timeout
—CN1.Run_Timer.TT	0	BOOL	Purge Timeout
—CN1.Run_Timer.DN	0	BOOL	Purge Timeout
—CN1.Run_Timer.FS	0	BOOL	Purge Timeout
—CN1.Run_Timer.LS	0	BOOL	Purge Timeout
—CN1.Run_Timer.OV	0	BOOL	Purge Timeout
—CN1.Run_Timer.ER	0	BOOL	Purge Timeout
⊞-CN1.Motor_Timer	{...}	TIMER	Motor Fault Timeout
⊞-CN1.Jam_Timer	{...}	TIMER	Jam Detection Timer

Monitor Tags / Edit Tags

Automate Tag Descriptions Using User-Defined Data Types

Pass-Through Descriptions Are Specific to Each Tag

If your application uses the same user-defined data type for many tags, the benefits are even more substantial.

- When multiple tags use the same user-defined data type, pass-through of descriptions is a great help. However, the descriptions may still be too general and lack meaningful documentation about the tag itself.

- To help in this area, RSLogix 5000 software offers an additional feature called *Append to Base Tag Description*. It concatenates the description of the user–defined data type member to the description from the base tag. This creates specific descriptions for each member of a tag that uses a user-defined data type.

In the following example, the software automatically builds a description for each member of the CN1 tag. It starts with the description for the base tag (Conveyor CN1) and then adds the description for the members from the user-defined data type.

Pass-Through Descriptions Adjust to the Type of Tag

The source for the pass-through description varies based on the type of tag, element, or member. Essentially, RSLogix 5000 software searches upward through the tag hierarchy for a description that is as close as possible to the item that needs to be documented.

- The search process begins with the user-defined data type, if one is used. If no user-defined data type is present, it steps up a level.
- If you have nested user-defined data types, it moves up to the root of the nested user-defined data type.
- If the element is an array, it pulls the description from the tag that defines the array.

The following table provides a brief overview of the look-up process for a pass-through description.

If the tag or array is:	Then for this:	The software looks for a description in this order:
not based on a user-defined data type	Array	n/a
	element of an array	tag that defines the array
	bit of an element of an array	1. element 2. tag that defines the array
based on a user-defined data type	single tag	user-defined data type
	member of a single tag	1. member in the user-defined data type 2. tag
	Array	user-defined data type
	element of an array	1. user-defined data type 2. tag that defines the array
	member of an element of an array	1. member in the user-defined data type 2. element 3. user-defined data type 4. tag that defines the array

Keep in mind that the software performs additional checks if:

- the member is nested more than one level of a user-defined data type.
- the *Append to Base Tag Description* feature is turned on.

The source description that is concatenated via the *Append to Base Description* varies depending on the type of tag.

This tag type:	Uses this base description:
non-array tag elements	description from the base tag
array tag elements	description from the base tag
user-defined data type member of an element of an array	element description, if one is present; otherwise it defaults back to the base tag's description.

Use Pass-Through Descriptions As a Starting Point

RSLogix 5000 software gives you many aids for working with pass-through descriptions:

- To override a pass-through description, simply enter a description for the tag element you want to change.

- The tag window uses different colors for pass-through versus normal descriptions. This helps you see the difference between a description that is being passed through from another source and one that is directly attached to the tag itself.

If you see a description in this color:	Then the description is a:
gray	pass-through description
black	normal (manually-entered) description

- To use a pass-through description as the basis for a normal description, click the right mouse button on the tag and choose *Paste Pass-Through*.

- To quickly determine where to go to change the source of a pass-through description, simply point your mouse to the description. A pop-up tool-tip lists the source of the description.

Pass-Through Descriptions Save Hours of Development Time

Pass-through descriptions can bring very substantial development savings. Designs that use user-defined data types see the most significant benefit.

- In the very simple conveyor example shown earlier, there were 8 members in the user-defined data type.

- Each description took about 20 seconds to enter and confirm.

- If you used this user-defined data type for 100 tags, this would amount to about 4 hours and 27 minutes of savings (20 Seconds * 8 Members * 100 tags).

Extending this to a normally sized system might result in user-defined data types with many more members. Additionally, the number of tag instances that could benefit from these user-defined data type descriptions would be substantially larger. As the sizes and quantities increase, so do the savings.

Several systems integrators who reviewed this feature feel it will save them hundreds of hours of development time. Similarly, end users of these systems expect that the improved documentation will reduce startup time, simplify maintenance, and reduce system downtime.

www.rockwellautomation.com

Corporate Headquarters
Rockwell Automation, 777 East Wisconsin Avenue, Suite 1400, Milwaukee, WI, 53202-5302 USA, Tel: (1) 414.212.5200, Fax: (1) 414.212.5201

Headquarters for Allen-Bradley Products, Rockwell Software Products and Global Manufacturing Solutions
Americas: Rockwell Automation, 1201 South Second Street, Milwaukee, WI 53204-2496 USA, Tel: (1) 414.382.2000, Fax: (1) 414.382.4444
Europe/Middle East/Africa: Rockwell Automation SA/NV, Vorstlaan/Boulevard du Souverain 36, 1170 Brussels, Belgium, Tel: (32) 2 663 0600, Fax: (32) 2 663 0640
Asia Pacific: Rockwell Automation, 27/F Citicorp Centre, 18 Whitfield Road, Causeway Bay, Hong Kong, Tel: (852) 2887 4788, Fax: (852) 2508 1846

Headquarters for Dodge and Reliance Electric Products
Americas: Rockwell Automation, 6040 Ponders Court, Greenville, SC 29615-4617 USA, Tel: (1) 864.297.4800, Fax: (1) 864.281.2433
Europe/Middle East/Africa: Rockwell Automation, Brühlstraße 22, D-74834 Elztal-Dallau, Germany, Tel: (49) 6261 9410, Fax: (49) 6261 17741
Asia Pacific: Rockwell Automation, 55 Newton Road, #11-01/02 Revenue House, Singapore 307987, Tel: (65) 6356-9077, Fax: (65) 6356-9011