



Re-use Ladder Code Investment
With Partial Import/Export

**New tools in RSLogix 5000 let
you re-use application code
more easily and decrease the
costs of a project**



Re-use Ladder Code Investment With Partial Import/Export

Improving Productivity with New Tools

About 40% of the cost associated with control system development is related to application code creation and configuration. Because this expense does not appear on a Bill-of-Materials, it is commonly overlooked as is the opportunity for potential savings. In most control systems, there is a reasonable amount of duplication that requires the same task to be performed repetitively. RSLogix™ 5000, version 13, provides new tools that help to improve productivity by allowing you to reuse previously-written code more easily.

When working in the RSLogix 5000 programming environment, you can use familiar drag & drop and cut/copy/paste Windows® programming tools to duplicate or move code from one part of a project to another or between projects. While these tools are extremely valuable in maintaining high productivity, Version 13 of RSLogix 5000 offers a new Ladder Diagram (LD) Partial Import/Export tool that is a step above these customary Windows operations.

This paper explains how you can use the Partial Import/Export feature in RSLogix 5000 to:

- easily export ladder diagram code.
- take advantage of the exported file's XML format to customize the code.
- easily reimport the customized code to another project.
- optimize the new project with tools such as user-defined data types (UDTs).
- reap the benefits of code reuse within and across projects, avoided typing mistakes and building a common code library.

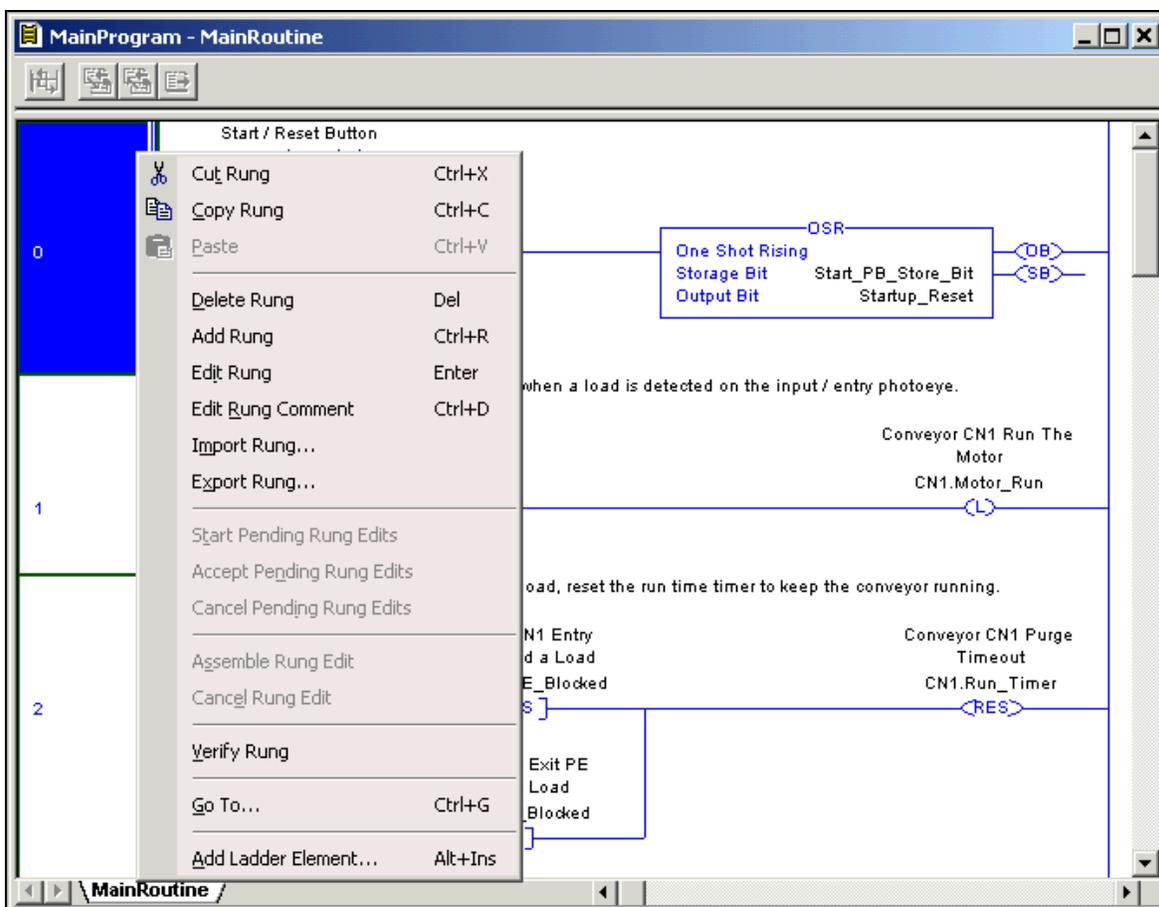
Exporting Ladder Diagram Code

The basic function that LD Partial Import/Export provides is the ability to extract specific rungs of code from a project and store them in an external file. Later, you can import this code back into the same project or into a different project later. The significant difference is that Partial Import/Export is not limited to just the LD rung source code. The exported file includes the definitions for tags, user-defined data types (UDT) or structures, and their associated description comments. This ensures that when the resulting file is imported into a project, all of the necessary tags and UDTs are automatically defined, and the code includes everything required to operate. Extending this even further, RSLogix 5000 provides a dialog listing of all of the tags and UDTs and their associated descriptions so that you can rename them prior to inserting them into a project. This ensures that the code is nearly ready to run and avoids potential tag-naming collisions that can be difficult to locate later.

Re-use Ladder Code Investment With Partial Import/Export

The operation begins in the RSLogix 5000 Ladder Diagram (LD) language editor. When you create the initial logic that you know you want to reuse, you can either build the logic specific to one instance of its use in the control system or build it generically, using wild cards in the names that can be located easily and renamed later. Once the program is developed, you simply:

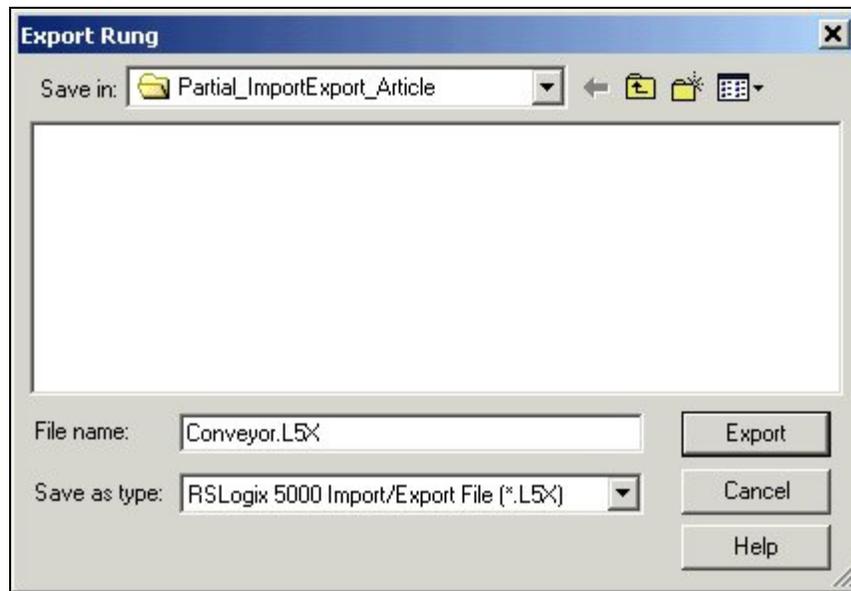
1. Select the rung or rungs of interest.
 - To select a contiguous group of rungs, press and hold the [Shift] key.
 - To select a non-contiguous group (i.e. rungs scattered throughout the file, press and hold the [Ctrl] key.
2. Right-click your mouse.
3. Select "Export Rung . . ." from the pop-up menu that appears.



RSLogix 5000 LD Editor Context Menu

Re-use Ladder Code Investment With Partial Import/Export

Once you select "Export Rung...", RSLogix prompts you for a file name into which to write the LD code; the file is created with an ".L5X" extension. After you select a file name, RSLogix 5000 scans the LD rungs to determine which tags are referenced and evaluates the tags to determine if any UDTs are used. Finally, the UDTs, tags, rungs and descriptions are written into the file.

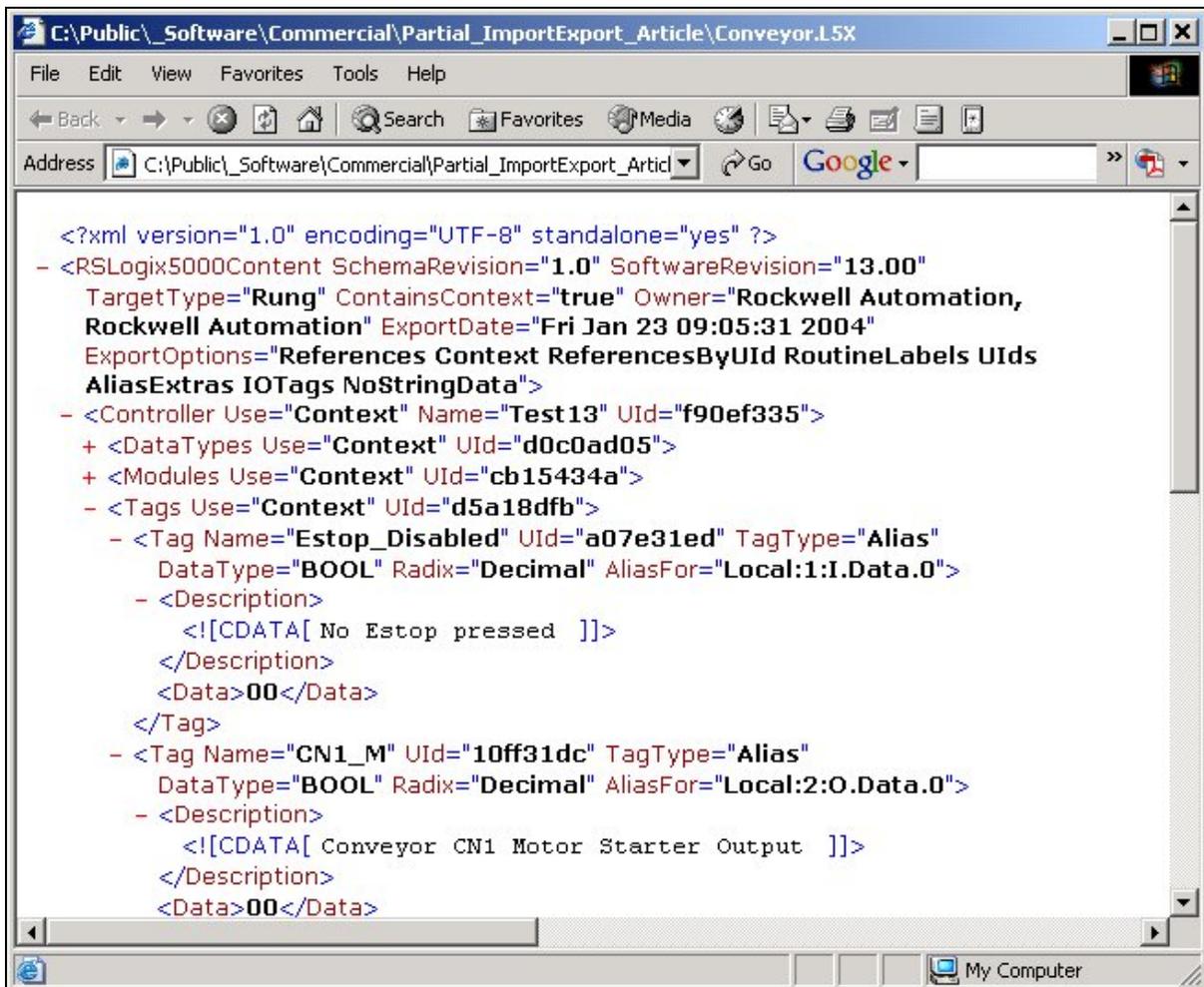


RSLogix 5000 LD Rung Export File Selection

Leveraging XML Format in Your Exported Text File

The exported information is written into an ASCII text file that leverages Extensible Markup Language (XML) tag formatting. The contents of this file is partially based on the full project export file (".L5K" extension). However, changes were made to improve the structure and to make sure it conforms to the XML standard. In addition to formatting, the text representation for the LD in the partial export file is represented a little differently than similar text in the full project export. While the rungs and instructions are still represented in neutral text, the instructions in the L5X file use a numeric ID to represent the tags being referenced. This numeric ID differs from the full project export that includes the name for each specific tag. The tags are defined toward the beginning of the L5X file with a similar ID. The benefit is that if you decide to edit the file with a text editor and rename the tag, the reference to the tag remains linked. In other words, you are no longer required to search through the file and rename each reference to the tag within the logic. A single rename

operation takes care of all of the instruction references just as with a tag rename operation in RSLogix 5000.



```
<?xml version="1.0" encoding="UTF-8" standalone="yes" ?>
- <RSLogix5000Content SchemaRevision="1.0" SoftwareRevision="13.00"
  TargetType="Rung" ContainsContext="true" Owner="Rockwell Automation,
  Rockwell Automation" ExportDate="Fri Jan 23 09:05:31 2004"
  ExportOptions="References Context ReferencesByUid RoutineLabels UIDs
  AliasExtras IOTags NoStringData">
- <Controller Use="Context" Name="Test13" Uid="f90ef335">
+ <DataTypes Use="Context" Uid="d0c0ad05">
+ <Modules Use="Context" Uid="cb15434a">
- <Tags Use="Context" Uid="d5a18dfb">
  - <Tag Name="Estop_Disabled" Uid="a07e31ed" TagType="Alias"
    DataType="BOOL" Radix="Decimal" AliasFor="Local:1:I.Data.0">
    - <Description>
      <![CDATA[ No Estop pressed ]]>
    </Description>
    <Data>00</Data>
  </Tag>
  - <Tag Name="CN1_M" Uid="10ff31dc" TagType="Alias"
    DataType="BOOL" Radix="Decimal" AliasFor="Local:2:O.Data.0">
    - <Description>
      <![CDATA[ Conveyor CN1 Motor Starter Output ]]>
    </Description>
    <Data>00</Data>
```

Microsoft Internet Explorer viewing a L5X file

The RSLogix 5000 XML schema for the L5X file is fully documented in the same manual as is the full project import/export, [Logix5000 Controllers Import/Export Reference Manual](#), publication number 1756-RM084. An ancillary benefit of using an XML schema is that most office PC tools can view and easily modify the file using standard text editors. Additionally, there are several third-party development libraries available to parse and manipulate XML programmatically. You can build your own tools to modify or generate the text file from scratch before importing it into your RSLogix 5000 project.

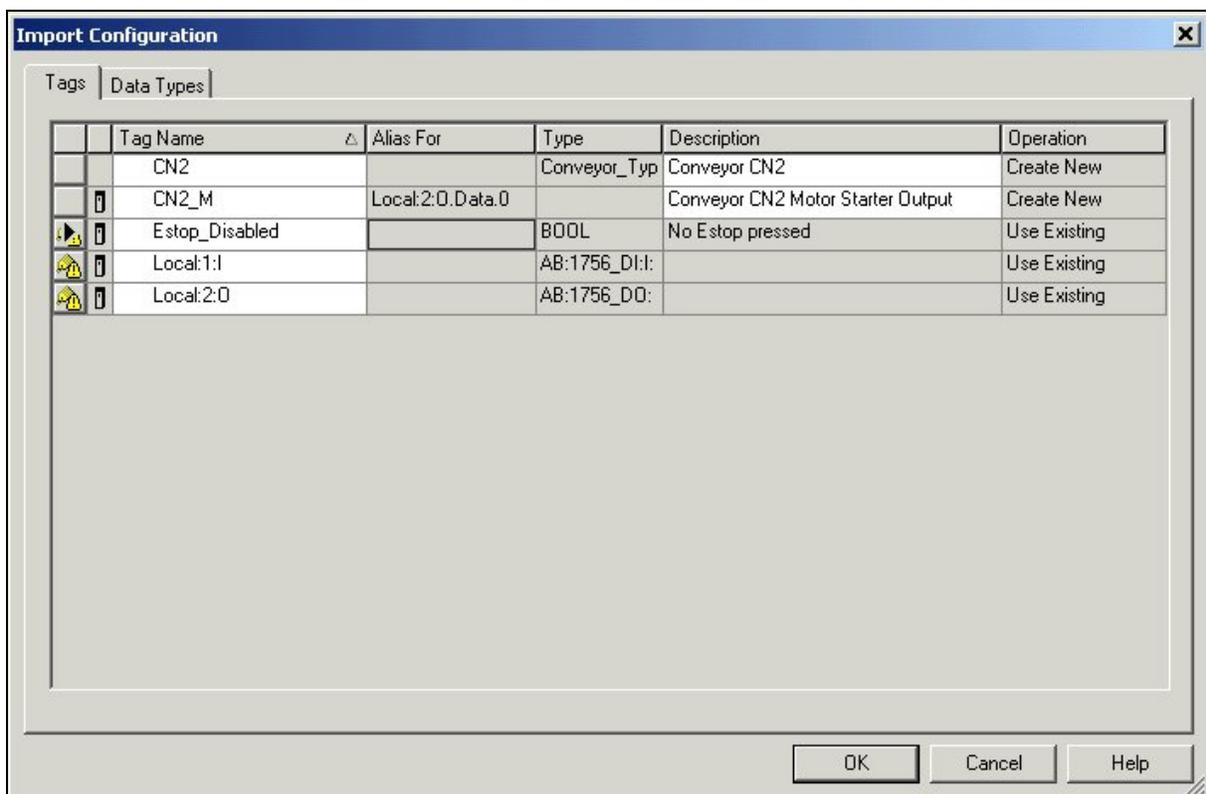
Re-use Ladder Code Investment With Partial Import/Export

Importing Ladder Diagram Code

The greatest benefit of exporting ladder rungs to a file comes when you want to leverage them to develop a project. Importing the code is as easy as exporting it.

1. Navigate in the RSLogix 5000 Ladder Editor to the rung where you want insert the rungs.
2. Right-click your mouse on the rung.
3. Select "Import Rungs..." to begin the import process.

RSLogix5000 prompts you for the name of the L5X file that you want to import. Browse to your L5X file. Once the L5x file is selected, RSLogix 5000 scans it to extract a list of the tags and UDTs that it contains and provide a listing of these tags/UDTs, along with an indication if the tag already exists in the controller. You can override the names contained in the file and even modify the descriptions. If the tag or UDT already exists in the project into which you are importing the file and you decide to keep the original name, the code binds to the existing tags automatically. If, however, you decide to rename anything, new tags or UDTs are created in the project and the new code is inserted with the appropriate references already created.

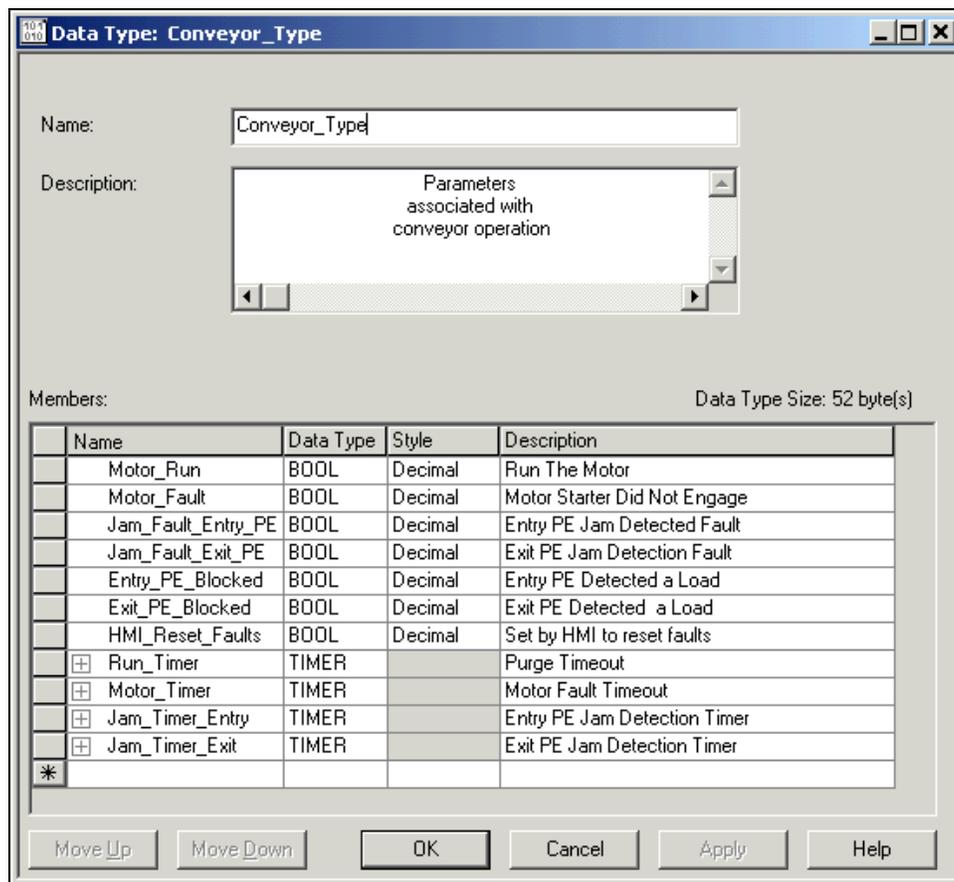


RSLogix 5000 LD Partial Import Configuration

Optimizing the Operation

Generally, Ladder Diagram code is developed using separate references for each individual piece of data. You can consolidate multiple data variables into a single compound structure (i.e., UDTs) in RSLogix 5000. You can reduce the changes you need to make each time you import the code file by leveraging UDTs when defining the initial code module. By combining many of the variables that the code needs into a single UDT, only one tag must be renamed on import, rather than a large number of individually-referenced tags.

For example, a block of code to control a conveyor might require several internal BOOL storage and fault bits and some timers. The UDT is created with all of the necessary data variables (known as members) and a single tag is created that utilizes this UDT. When the code is created, the instructions reference the UDT members in the tag. On import, only the tag's root definition must be renamed. When this occurs, all of the references to UDT members of the tag are updated automatically.

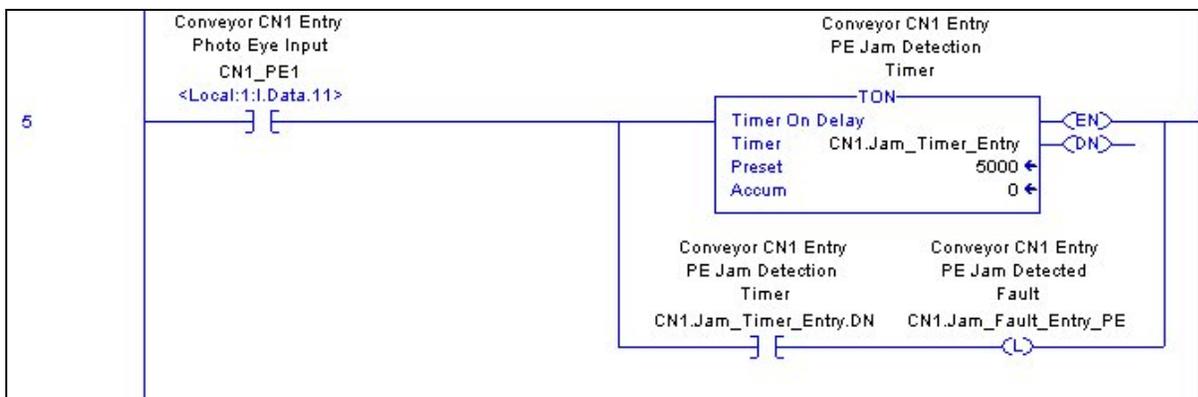


RSLogix 5000 Editor with Conveyor Type Example

Re-use Ladder Code Investment With Partial Import/Export

In this simple example, six internal store bits and four timers are consolidated into one UDT. On import, a single rename operation is needed to instantiate the code. Compare this to the ten separate operations that would have been necessary if each of these variables had been created as a stand-alone tag. If this same block of code were to be used 100 times in the project, it would be much easier to rename 100 UDT tags rather than 1,000 standalone tags. This programming method saves a substantial amount of development time.

An additional benefit of this UDT programming method is that RSLogix 5000 version 13 offers a feature that automatically creates tag descriptions based on the description you enter in the UDT, greatly reducing the amount of documentation you need to create.



Sample rung utilizing UDT based tags for internal timers and bit references

Reaping the Benefits of Partial Import/Export

The Ladder Diagram Partial Import/Export feature has many benefits.

- The first is the ability to reuse LD code across and within projects. This code reuse reduces the amount of development effort required, shortens the project delivery cycle and, ultimately, the system cost.

Using the simple example presented previously, there are nine rungs of code needed to control the conveyor. Within these nine rungs there were 41 instructions and 42 references to 15 different tags. The typical approach would be to duplicate the logic and then manually change all of the references. The majority of this effort can be eliminated easily by using the UDT programming method presented here coupled with the LD partial Import Export capability of RSLogix 5000.

<p>Effort to duplicate manually</p> <p>15 tags x 30 Sec. to create / rename = 7.5 Minutes 42 references x 20 Sec. to rename and validate = 14 Minutes (7.5 + 14) x 100 Instances = ~ 36 Hours of manual effort (15 + 42) x 100 Instances = 5,700 Manual rename operations</p>
<p>UDT Method with LD Partial Import / Export</p> <p>2 Tags x 30 Sec to reassign alias references = 1 Minute 4 references x 15 sec to rename during import = 1 Minute (1 + 1) x 100 Instances = ~ 1.5 hours (2 + 4) x 100 Instances = ~ 600 Manual rename operations</p>
<p>Savings</p> <p>36 Hours manually – 1.5 UDT / Import method= ~ 34 hours</p>

Re-use Ladder Code Investment With Partial Import/Export

- In addition to the productivity benefit, the LD Partial Import/Export feature helps you to avoid the typing mistakes that commonly occur when developing code. If you manage to enter 99% of the renames correctly the first time, you still encounter 57 programming errors (1% x 5,700 references) using the manual method. Compare that to a potential of six errors using the automated method. If each of these mistakes costs you another 20 minutes per error to locate and find, you would save another 17 hours of development ($[(57-6)*20]/60 = 17 \text{ Hours}$). (And, perhaps most significantly, this time is usually during the installation when the cost per hour is the most expensive.)
- Finally, because you can standardize the code for a system to a common library, the system is easier to learn and maintain. This common library benefits the installer because he can utilize different personnel across multiple projects and it benefits the end user because the code within the system is consistent.

Taking This to The Next Level

Besides the tools that are integral to RSLogix 5000, an additional package is available from Rockwell Software to further automate and optimize your development. The "RSAutomation Desktop" software provides a set of template library management tools that expands beyond ladder logic to include HMI and other project-related documentation. Contact your Rockwell Automation sales representative or distributor or visit the Rockwell Software web site for additional information

<http://www.software.rockwell.com/rsautomationdesktop/>

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