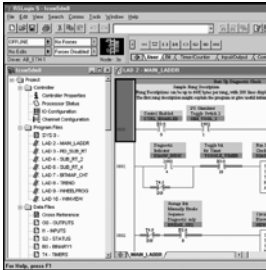


# Import/Export Project Components



## Programming Manual

## Important User Information

Solid state equipment has operational characteristics differing from those of electromechanical equipment. Safety Guidelines for the Application, Installation and Maintenance of Solid State Controls (publication SGI-1.1 available from your local Rockwell Automation sales office or online at <http://literature.rockwellautomation.com>) describes some important differences between solid state equipment and hard-wired electromechanical devices. Because of this difference, and also because of the wide variety of uses for solid state equipment, all persons responsible for applying this equipment must satisfy themselves that each intended application of this equipment is acceptable.





In no event will Rockwell Automation, Inc. be responsible or liable for indirect or consequential damages resulting from the use or application of this equipment.

The examples and diagrams in this manual are included solely for illustrative purposes. Because of the many variables and requirements associated with any particular installation, Rockwell Automation, Inc. cannot assume responsibility or liability for actual use based on the examples and diagrams.

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

Reproduction of the contents of this manual, in whole or in part, without written permission of Rockwell Automation, Inc., is prohibited.

Throughout this manual, when necessary, we use notes to make you aware of safety considerations.

<b>WARNING</b> 	Identifies information about practices or circumstances that can cause an explosion in a hazardous environment, which may lead to personal injury or death, property damage, or economic loss.
<b>IMPORTANT</b>	Identifies information that is critical for successful application and understanding of the product.
<b>ATTENTION</b> 	Identifies information about practices or circumstances that can lead to personal injury or death, property damage, or economic loss. Attentions help you identify a hazard, avoid a hazard, and recognize the consequence
<b>SHOCK HAZARD</b> 	Labels may be on or inside the equipment, for example, a drive or motor, to alert people that dangerous voltage may be present.
<b>BURN HAZARD</b> 	Labels may be on or inside the equipment, for example, a drive or motor, to alert people that surfaces may reach dangerous temperatures.

Allen-Bradley, Rockwell Automation, RSLogix 5000, SoftLogix, and TechConnect are trademarks of Rockwell Automation, Inc.

Trademarks not belonging to Rockwell Automation are property of their respective companies.

	<b>Chapter 1</b>	
<b>Import and Export Logic Components</b>	Introduction . . . . .	5
	Export Components . . . . .	6
	Considerations . . . . .	6
	Import Components . . . . .	8
	Import Configuration. . . . .	9
	Configure Component Import . . . . .	9
	Configure References Import . . . . .	12
	Configure References Example. . . . .	14
	Undefined References . . . . .	15
	Attention Flags. . . . .	16
	Import . . . . .	16
	Import Fails or is Canceled . . . . .	16
	Import Considerations. . . . .	16
	Online Import . . . . .	17
	Motion . . . . .	18
	Safety . . . . .	18
	Security. . . . .	18
	Language Switching . . . . .	19
	Search and Replace. . . . .	21
	<b>Chapter 2</b>	
<b>Import and Export Rungs</b>	Introduction . . . . .	23
	Export Considerations. . . . .	23
	Import Considerations. . . . .	24
	<b>Chapter 3</b>	
<b>Import and Export Routines</b>	Introduction . . . . .	25
	Export Considerations. . . . .	25
	Import Considerations. . . . .	25
	. . . . .	26
	<b>Chapter 4</b>	
<b>Import and Export Programs and Equipment Phases</b>	Introduction . . . . .	27
	Export Considerations. . . . .	27
	Import Considerations. . . . .	27
	<b>Chapter 5</b>	
<b>Import and Export User-defined Types</b>	Introduction . . . . .	31
	Export Considerations. . . . .	31
	Import Considerations. . . . .	31
	<b>Chapter 6</b>	
<b>Import and Export Add-On Instructions</b>	Introduction . . . . .	33
	Export Considerations. . . . .	33
	Import Considerations. . . . .	34

## **Import and Export Tags**

### **Chapter 7**

Introduction .....	37
Export Considerations.....	37
Import Considerations.....	37

## Import and Export Logic Components

### Introduction

This chapter explains import and export of logic components.

Topic	Page
Export Components	6
Import Components	8
Import Configuration	9
Import Considerations	16
Language Switching	19
Search and Replace	21

The ability to perform online edits has existed in RSLogix 5000 software and the Logix family of controllers for many versions. Users have also been able to export and import rungs, Add-On Instructions, and user-defined data types offline as well. With version 17, the extended import/export functionality now provides a greater degree of flexibility during development and deployment of projects. It supports the ability to export a component of a RSLogix 5000 project and then import that component into either the same or a different project, both offline or online (even in Remote Run mode). Import components to create something new in a project or to update something that already exists. You can export and import the following logic components, both offline and online:

- Rungs
- Routine
- Program/equipment phase
- User-defined data type (UDT)/user-defined string type
- Add-On Instruction
- Trend

Using the extended import/export feature, changes or additions to a project that may be impractical to do with online editing can be completed offline and then imported in one operation into a running controller. Engineering collaboration is also facilitated in that many engineers can work on one or more of the project components independently offline and then the changes merged into a single project by using import.

With online editing, you modify or create a component and then modify other components referenced by that component in the project sequentially. Importing online differs from online edits in that it provides the ability to create and delete components, as well as update referenced components, in one step.

## Export Components

Follow these steps to export components.

1. In the RSLogix 5000 controller organizer select the component to export.

---

**TIP** For rung export, select the rungs in the ladder routine editor.

---

2. Right-click and choose the export component menu item.

To export a	Select
Set of rungs	Export Rungs
Routine	Export Routine
Program	Export Program
Equipment Phase	Export Equipment Phase
User-defined data type	Export Data Type
User-defined string type	Export Data Type
Add-On Instruction	Export Add-On Instruction
Trend	Export Trend

3. Enter a file name for the export file.
4. Edit the default description or add a description if desired

A default description is supplied for some component exports.

---

**TIP** The file description can be viewed in a tooltip during import when selecting which file to import.

---

5. Click Export.

## Considerations

Keep these considerations in mind when exporting components.

- You may get more than you expected in an L5X file. Some components referenced by the exported component will also be exported.

When exporting a	The exported component includes the	The exported file also includes referenced
Set of rungs	Rungs	Controller-scoped tags, program-scoped tags, user-defined data types, user-defined string types, Add-On Instructions
Routine	Routine	Controller-scoped tags, program-scoped tags, user-defined data types, user-defined string types, Add-On Instructions
Program	Program, routines, and program-scoped tags	Controller-scoped tags, user-defined data types, user-defined string types, Add-On Instructions
Equipment Phase	Equipment phase, routines, phase-scoped tags, and controller-scoped PHASE tag	Controller-scoped tags, user-defined data types, user-defined string types, Add-On Instructions
User-defined data type	User-defined data type	User-defined data types, user-defined string types, Add-On Instructions
User-defined string type	User-defined string type	
Add-On Instruction	Add-On Instruction, Logic routine and any scan mode routines, and parameter and local tags	User-defined data types, user-defined string types, Add-On Instructions
Trend	Trend	

For example, if you export a rung with an instruction that uses (references) a bit in a tag that is a user-defined type tag, the export content includes the referenced tag and the referenced user-defined type along with the rungs. When you import the rung, you can choose whether to import the referenced tag and user-defined type as well.

- Only the offline data values of tags will be exported to the L5X file, even when the project is online with the controller. When exporting components that include tags while online, the software will prompt to upload tag values before exporting the component.

**IMPORTANT**

Whether exporting while offline or online with the controller, only offline data values of tags are exported to the L5X file.

- You can export a component that has Test Edits, but the resulting file cannot be imported. If an L5X file with Test Edits is selected for import, then the import will be aborted.

## Import Components

Follow these steps to import components.

**WARNING**

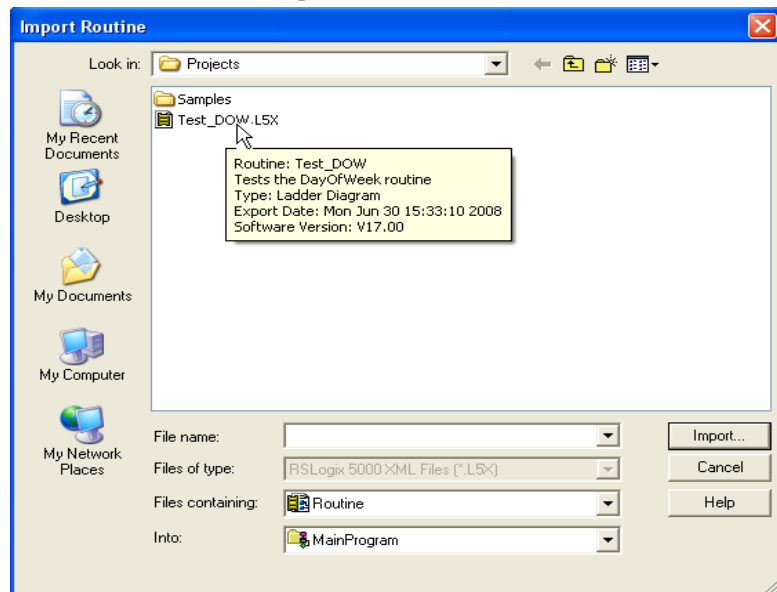


Importing components into a controller while online can affect controller operation. Use caution when modifying logic while online in Remote Run or Run mode.

1. Right-click the item to import a component into and choose the import component menu item.

To import a	Right-click	And choose
Set of rungs	A rung or set of rungs within a Ladder routine editor	Import Rungs
Routine	A program or equipment phase	Import Routine
Program	A task, the Controller Fault Handler folder, the Power-Up Handler folder, or the Unscheduled Programs/Phases folder	Import Program
Equipment Phase	A task or the Unscheduled Programs/Phases folder	Import Equipment Phase
User-defined data type	User-Defined folder	Import Data Type
User-defined string type	Strings folder	Import String Type
Add-On Instruction	Add-On Instructions folder	Import Add-On Instruction
Trend	Trends folder	Import Trend

2. Select the file to import.



3. Click Import.

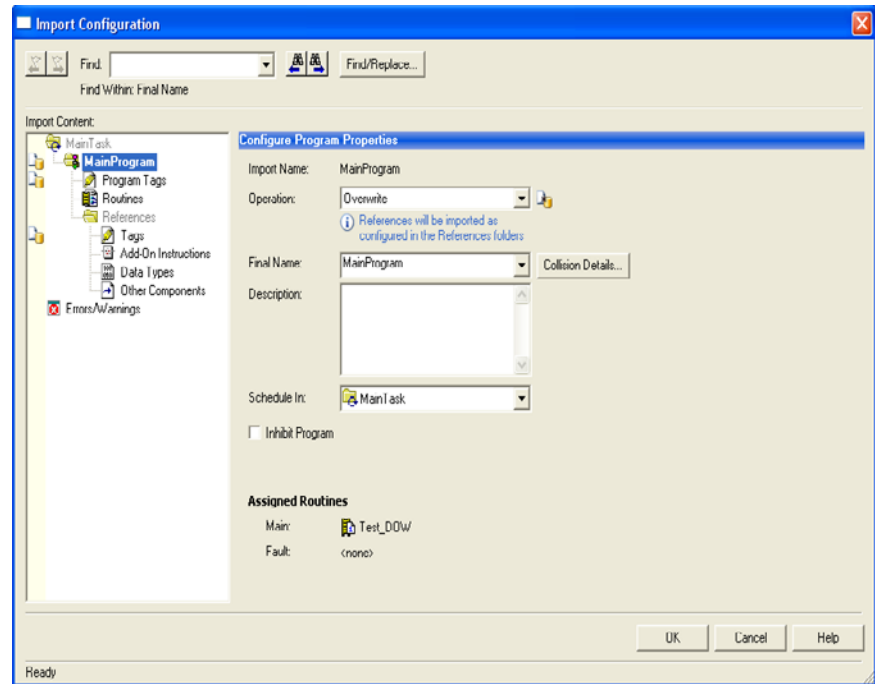


## Import Configuration

This section explains how to configure the import.

### Configure Component Import

When you import a component, the previously exported information is brought into the project based on the import configuration you select. You determine how the imported component, a program for example, connects to the existing project by either overwriting an existing component of the same type or creating a new one.



#### TIP

Trends do not require any import configuration, therefore the Import Configuration dialog box is not displayed and the trend is imported immediately.

An import component collides with an existing component if a component of the same type, scope, and with the same name already exists in the project. By configuring the Final Name of the imported component, you determine whether the imported component collides with an existing component or if it does not (if a component of the same type, scope and with the same name does not exist in the project). For a collision, you can use the existing project component or you can overwrite it. For a non-collision, you can create a new project component or you can discard, and not import, the component from the import file. In any case, any imported logic references are updated to be the same as the Final Name.

By configuring the Operation for the component, you determine whether it is imported from the file or if an existing component is used. Incompatibility of a collision or other situations, for example, project security privileges, may affect what Operations are permitted for a component.




**Configuration Options**

<b>Configuration</b>	<b>Description</b>
Import Name	The Import Name is read only and displays the name of the component as read from the import file.
Operation	<p>The Operation determines whether the component from the import file is imported. Not all operations are applicable to a given component in all situations. Sometimes there is only one possible operation permitted in a particular situation.</p> <p>Select:</p> <p>Use Existing (collisions) to use the existing component in the project. The component definition in the import file will not be imported. If the component collides and there is no definition for the component in the import file, then this is the only Operation available.</p> <p>Overwrite (collisions) to import the component definition from the import file and overwrite the existing component definition in the project.</p> <p><b>Important:</b> If you select Overwrite for a reference tag, the tag data that is different is written to the offline project only, even if you are online with the controller.</p> <p>Create (non-collisions) to import the component from the import file creating a new component in the project.</p> <p>Discard (non-collisions) to discard the component and not import it. If the main component from the import file (the program during a program import, for example) is discarded, the references may still be imported. If a referenced component is discarded, any imported logic references will be undefined in the project and may not verify after import.</p> <p>Undefined (non-collision) is the only Operation available if the component does not collide and there is no definition for the component in the import file. The imported logic will not verify after import.</p> <p>Delete (non-collision) indicates that this component will be deleted from the project on import.</p>
Final Name	<p>The Final Name determines the name of the component as it will appear in the project after the import (except in the case of a Delete Operation).</p> <p>You can create or avoid collisions by changing the Final Name. If you change the Final Name to be different from the Import Name, all logic references in the import content are updated.</p> <p>If you change the Final Name of a component during configuration so that it collides with a different existing component after it was colliding with another, any changes you have already made to the Operation are preserved.</p>

**Configuration Options**

<b>Configuration</b>	<b>Description</b>
Description	<p>The Description that is displayed depends on the Operation. If the Operation is:</p> <ul style="list-style-type: none"> <li>• Overwrite or Create, the description is initially the description from the import file. It may be edited during import configuration.</li> <li>• Discard, the description is initially the description from the import file. It is read-only and may not be edited.</li> <li>• Use Existing or Delete, the description is the existing component description. It is read-only and may not be edited.</li> <li>• Undefined, no description is displayed,</li> </ul> <p>The description for members of an expanded tag follow the same rule the tag follows. Only members of the tag that have descriptions are displayed when the tag is expanded.</p>

When there is a collision, the components colliding may be identical or they may be different. If they are the same, the Operation defaults to Use Existing. If they are different, imported components default to Overwrite while referenced components default to Use Existing. Select Overwrite to use the component definition from the import file or Use Existing to use the component in the project. Icons on the Import Configuration dialog box (next to the Operation control) show whether colliding components are identical or different.

<b>Situation</b>	<b>Icon</b>
The component appears only in the import file (non-collision).	
The component appears only in the project (non-collision).	
The component collides with a component in the project and there are differences.	
The component collides with a component in the project and they are identical.	No icon

To see the differences between the import component and the colliding project component, click Collision Details and a Collision dialog box will appear. Use the information in the Collision dialog box to determine whether to overwrite or use the existing component.

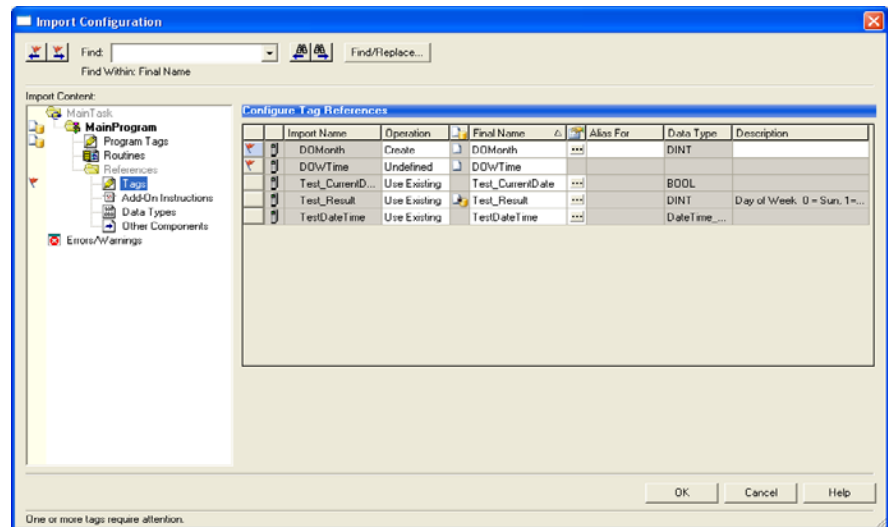
For components that do not collide, click Properties for more information on the component.

**IMPORTANT**


A component you select for import that collides with an existing component defaults to overwriting the project component unless the components are identical, in which case the project component is used. If the project component is used, references may still be imported.


### Configure References Import

In addition to determining how the component is imported, you also need to configure how references are imported. The references are all of the other components used by the exported component at the time it was exported. For example, a program L5X file contains the definition of all the controller-scoped tags, user-defined data types, user-defined string types, and Add-On Instructions referenced by the program at the time the program was exported. During import configuration, you determine what the references from the imported component connect to after import. You can connect a reference to an existing component in the project and not import the referenced component from the import file. Or, you can use the referenced component in the import file by either creating a new component in the project from the definition in the import file or by overwriting an existing component of the same type in the project.



As with imported components, if the Final Name of a referenced component (such as a tag) matches a component of the same type and scope in the project, these components collide. If there is no collision, the component is identified as needing to be created (Operation is Create).

Icons in the Differences column () on the Import Configuration dialog box show whether components collide with differences or if the component only exists in the import file or the project.

Click the icon in the Details column () and a Collision dialog box will appear showing the differences between the import component and the colliding project component. Use the information in the Collision dialog box to determine whether to overwrite or use the existing component.

For components that do not collide, click the same icon for more information on the component.

---

**TIP**

A referenced component that collides with an existing component defaults to using the project component (Operation is Use Existing), even if the definitions of the components are different. To import the referenced component definition from the import file, Overwrite must be selected.

---

Connecting a referenced component to a component of the same type, but a different name, in the project will cause any reference to the component in the imported logic to be updated to the new name. For example, if you connect a Boolean reference tag A to B[3].ACCUM.4 (where B is an existing tag in the project that is an array of structures), then all references in the imported logic that used to say A will now say B[3].ACCUM.4. Another way to connect reference tags to existing tags in the project is to modify their Alias For field. This is permitted when the reference tag is being created or the reference tag is overwriting an existing tag (offline only), regardless of whether it was originally an alias tag. The Import Configuration dialog box lets you configure all the references you need to connect during the import process.

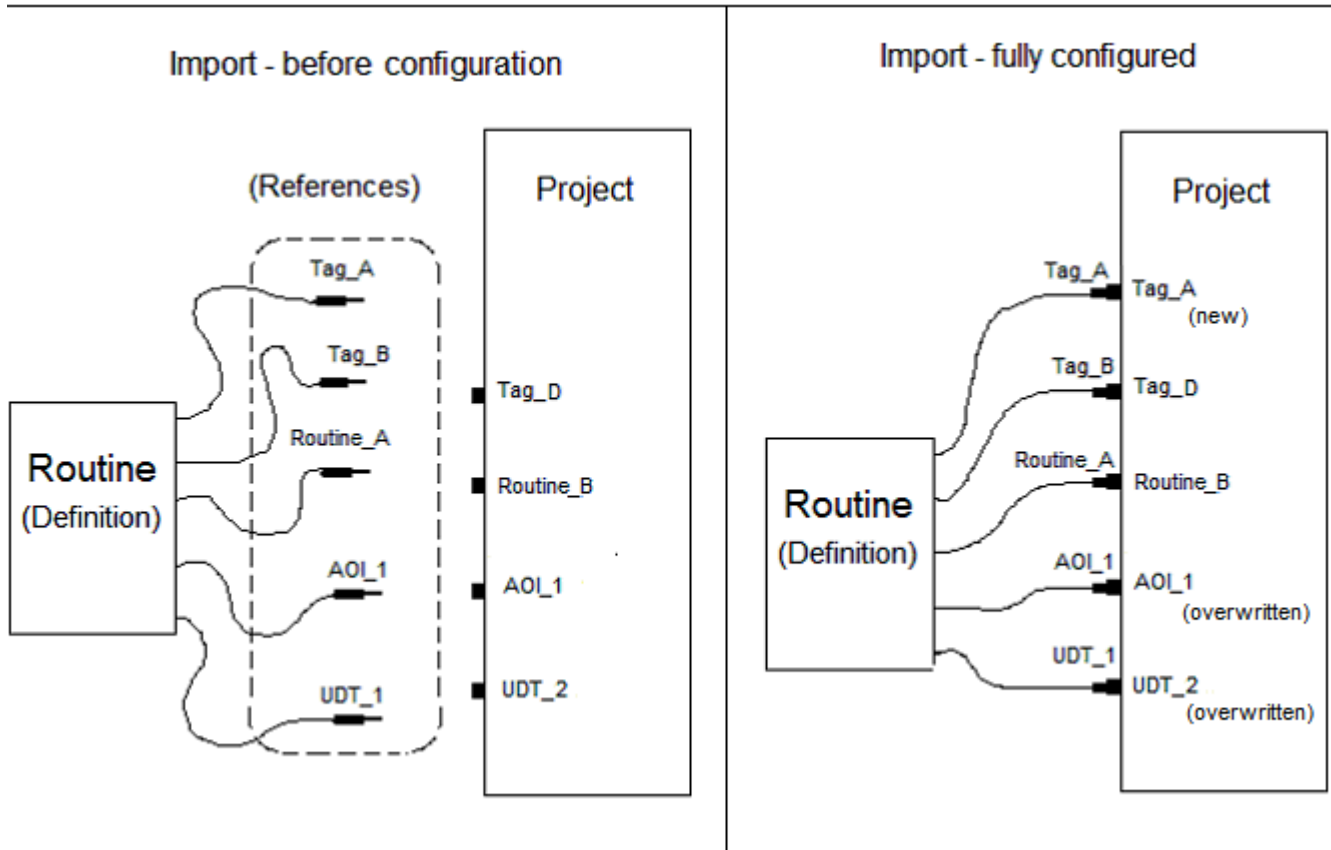
---

**IMPORTANT**

It is important to note that even if a component has an Operation of Discard, Use Existing, or Undefined, if the Final Name is different from the Import Name then the imported logic references to that component will be updated to reflect the Final Name. For example, if you enter an invalid name as the Final Name for a tag reference, the Operation will be set to Discard but the imported logic references to that tag will still be updated to the invalid Final Name.

---

## Configure References Example



For example, if you import a routine, the routine itself is the main import component. References that need to be connected during the import could include program-scope tags, controller-scope tags, user-defined data types, any routines called by JSR instructions, and any Add-On Instructions in the routine logic.

In this example, a routine export file contains the definitions of Tag\_A, Tag\_B, AOI\_1, and UDT\_1. The routine also contains a reference to Routine\_A, but doesn't include the logic in Routine\_A.

During import configuration, connect each:

- reference tag in the import routine to a program or controller-scoped tag in the project.
- reference routine in the import routine to a routine in the project.
- user-defined data type defining a reference tag in the import routine to a user-defined data type in the project.
- Add-On Instruction in the import routine to an Add-On Instruction in the project.

These tags, user-defined data types, and Add-On Instructions might already exist in the project or they can be created when you import the routine.

- Tag\_A is set to Create, so the imported logic referencing Tag\_A will reference the newly created Tag\_A in the project.
- Tag\_B has its Final Name changed to Tag\_D and is set to Use Existing, so it will not be created, and imported logic referencing Tag\_B will instead reference existing Tag\_D.
- Routine\_A has its Final Name changed to Routine\_B, so imported JSRs to Routine\_A will use the existing Routine\_B.
- AOI\_1 is set to Overwrite an existing Add-On Instruction called AOI\_1, so not only will imported logic and tags use AOI\_1, but also any existing references to AOI\_1 in the project will now use the new definition of AOI\_1 from the imported routine.
- UDT\_1 is set to Overwrite an existing user-defined data type called UDT\_2, so imported logic and tags will use UDT\_2 and the existing UDT\_2 will be overwritten (except for the name) with the definition from UDT\_1. Logic and tags from the project will now use the new definition of UDT\_2.

## Undefined References

In the RSLogix 5000 editor, it is possible to create logic that references tag, user-defined data type, or Add-On Instruction components that do not yet exist. These are considered undefined references. If you export that logic, there is no component definition to export for that undefined reference. On import, these tags, user-defined data types, or Add-On Instructions appear with an Operation of Undefined during Import Configuration unless the component exists in the project (in which case the Operation is Use Existing). Similarly, when logic references any type of component other than a tag, user-defined data type, user-defined string type, or Add-On Instruction (for example, a routine that is referenced by a JSR instruction), the definition for these referenced components are not included in the import file and they also appear with an Operation of Undefined if the component does not exist in the project.

During import configuration, these undefined references can be configured as well. Since there is no definition for these components in the L5X file, you cannot overwrite a project component or create a new component during the import process, but you can use an existing component for that reference. You configure which component in the project you want to connect the reference to or leave the reference undefined. An undefined reference that collides with an existing component defaults to using the project component.

## Attention Flags

Attention (red) flags bring to your attention import situations that may be unintended or may lead to a project that will not verify successfully after import. What they describe may or may not prevent the import from proceeding. If online, and the project does not verify successfully when accepting or finalizing edits, then the import fails. Not all attention flags cause a project not to verify. The red flag will indicate the potential problem and you must decide what you want to do. You can ignore the attention message or try to change the Final Name or Operation of the component to resolve the issue.

---

**TIP**

Hovering the mouse pointer over the attention flag will display a tooltip explaining the situation.

---

## Import

Click OK to complete the import.

## Import Fails or is Canceled

When an import fails or is canceled, all changes caused by the import are discarded, leaving the project in the same state it was in before the import, even if the project is online with the controller. If the import fails, the Import Configuration dialog box is displayed with the Errors/Warnings Pane selected. To attempt to import again, inspect the Errors and Warnings, reconfigure the import, and click OK again.

## Import Considerations

This section covers general considerations for all component type imports. See the chapters for specific components (program, routine, rungs, Add-On Instructions, user-defined data types, and tags) for additional export and import considerations for that component, whether it is imported directly or as a reference during another component import.



## Online Import

Topic	Consideration
Tag Data Values	<p>When overwriting tags while online, data values are not written to the online project. Data values for overwritten tags while online are written only to the offline project.</p> <p>Data values for tags created while online are written to both the offline and online project.</p>
Locking Controller Edits	<p>When performing an online import the controller will be locked, preventing other workstations connected to the controller from making edits until the import completes. If another workstation has the controller locked, an online import cannot be initiated.</p>
Online Options	<p>When importing programs, equipment phases, routines, or rungs online, the logic can be imported as:</p> <ul style="list-style-type: none"> <li>• Pending Edits that exist only in the offline project and are not downloaded to the controller</li> <li>• Accepted Edits that are downloaded to the controller where they can be tested, accepted, or canceled.</li> <li>• Finalized Edits that are downloaded to the controller, assembled, and cannot be canceled.</li> </ul> <p>As with normal online editing, with any online option chosen any tags, data types, or Add-On Instructions that are created or modified will be downloaded immediately. In addition any task, program, or routine properties that are modified will be downloaded immediately. These changes will remain in the controller, even if logic edits are later canceled. Pending Edits and Accepted Edits options do not apply to Add-On Instructions and user-defined types so Online Options are not presented during an Add-On Instruction or user-defined type import.</p>
Asynchronous Import	<p>Import operations are not synchronized with the Program Scan (just as online editing operations are not synchronized). For most objects, this does not matter because:</p> <ul style="list-style-type: none"> <li>• existing online tag data values are never overwritten.</li> <li>• new tags and their initial data values are written to the controller before the new logic executes.</li> <li>• the switch-over to the new logic is synchronized with the program scan (just like online editing).</li> </ul> <p>In general, although online tag data values are not overwritten, some object attributes can change asynchronously to the Program Scan. Some examples:</p> <ul style="list-style-type: none"> <li>• changed attributes of existing programs or equipment phases (such as changing the Initial Step Index of an equipment phase) will be written to the controller before switch-over, so old logic can execute with the new configuration.</li> <li>• an existing program with no main routine could become configured with a main routine as part of import and that main routine will execute immediately even though the new logic for that routine has not yet been accepted.</li> <li>• changing Logix-based alarm tag configuration (such as alarm messages or associated tag definitions) can go into effect while the old alarm trigger logic is still active, resulting in alarm messages that don't match the condition.</li> </ul>

## Motion

Topic	Consideration
Motion Components Limitations While Online	Motion tags (Axis, Coordinate System, and Motion Group) cannot be created, modified or deleted during an online import. Existing motion tags can be referenced. When doing an offline import, motion tags can be created and modified.

## Safety

Topic	Consideration
Safety Components Limitations While Online	Safety components cannot be created, modified or deleted during an online import.
Safety Components Limitations While Offline	Safety components cannot be created, modified or deleted during an offline import if a Safety Signature exists, the project is Safety locked, a non-recoverable safety fault is present, the Safety Partnership status between the primary controller and the safety partner is any value other than "OK", or if Safety modify permissions are denied.
Scheduled Location	In a Safety project, Safety programs cannot be imported into a Standard task and vice versa.
Safety Programs Imported Into a Standard Project	Safety programs imported into a Standard project will be converted to Standard programs during the import processing and a warning will be emitted to the Errors/Results Pane during initial parsing of the L5X file. In this case, imported logic will not verify after import if it contains instructions that are only valid in a Safety routine.
Controller Fault Handler and Power-Up Handler	A Safety program cannot be scheduled in the Controller Fault Handler or Power-Up Handler folders.

## Security

Topic	Consideration
Project is Secured	If the project is secured, the import process may not be able to create, delete, or modify some components, depending on the security actions that have been granted or denied for those components.

## Language Switching

With RSLogix 5000 software, version 17, you have the option to display project documentation, such as tag descriptions and rung comments for any supported localized language when using the RSLogix 5000 software. You can store project documentation for multiple languages in a single project file rather than in language-specific project files. You define all the localized languages that the project will support and set the current, default, and optional custom localized language. The software uses the default language if the current language's content is blank for a particular component of the project. You can use a custom language to tailor documentation to a specific type of project file user. Once you enable language switching in RSLogix 5000 software for a project, you can dynamically switch between languages in that project as you use the software.

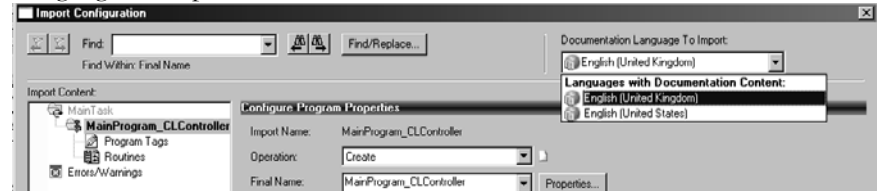
Project documentation that supports multiple languages includes the following:

- Component descriptions in tags, routines, programs, equipment phases, user-defined data types, members of user-defined data types, and Add-On Instructions
- Trends
- Controllers
- Alarm Messages (in configuration of ALARM\_ANALOG and ALARM\_DIGITAL tags)
- Tasks
- Property descriptions for modules in the Controller Organizer
- Rung comments, SFC text boxes, and FBD text boxes

The export and import features of RSLogix 5000 support language switching. Project components can be successfully transferred between projects that have different language switching settings.

If a component was exported to an L5X file from a project that did not have language switching enabled and is then imported into a project that has language switching enabled, all project documentation brought in by the import will be associated with the project's current language.

If a component was exported to an L5X file from a project that had language switching enabled, all language switched comments will be exported with the component. When an L5X file that includes language switched comments is imported, the Import Configuration dialog box includes a Documentation Language to Import selection.



- If the project has language switching enabled, all language switched comments from the L5X file will be imported into the project. This may result in some additional languages now being available in the project after import. Selecting a language in Documentation to Import allows you to edit that language’s comments from the import file during the import configuration process.
- If the project does not have language switching enabled, only one set of localized comments may be imported from the import file. In this case, selecting a language in Documentation to Import determines which language’s comments are imported with components from the import file. While you may edit all of the language’s comments during the import configuration process, only the final language selected when import proceeds will be imported with components.

**IMPORTANT**

Only comments associated with imported components are brought in from the import file. If an existing component is used, import file comments for that component are not imported.

**IMPORTANT**

Multiple localized alarm messages can be stored (in configuration of ALARM\_ANALOG or ALARM\_DIGITAL tags) even if the project does not have language switching enabled. Therefore, all alarm messages will be imported, regardless of the Documentation Language to Import selection on the import configuration dialog.

**IMPORTANT**

The Documentation Language to Import selection does *not* affect the setting of the language defined in the project after import.

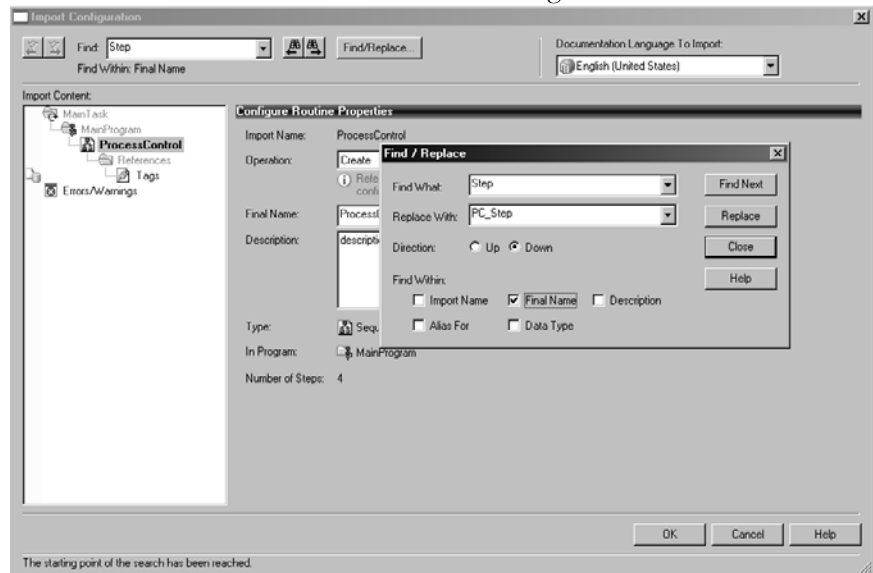
## Search and Replace

Use the Find controls in the Import Configuration dialog box control bar to search for text within the Import Configuration dialog box. Use the Find / Replace dialog box to configure whether the Find will search through the Import Name, Final Name, Description, Alias For, and/or the Data Type fields. As the next item is searched for, the Import Configuration dialog box will automatically switch to different panes within the dialog as needed.

You can replace editable fields during the search. This facilitates development of a component template that you may import repeatedly, replacing template text during each import. When designing a program template, keep in mind that during a program import, the program tag names and descriptions cannot be edited or replaced. However, during import you may edit the names and descriptions of controller-scoped tags referenced from the program. Similarly, when importing an equipment phase, you cannot edit the names and descriptions of tags in the equipment phase and when importing an Add-On Instruction, you cannot edit the names and descriptions of the parameters and local tags.

Follow these steps to find and replace text during the import configuration process.

1. Click Find/Replace in the Import Configuration dialog box.
2. Use the Find Within checkboxes to configure which fields to search.



3. Click Find to search for the next matching item.
4. Click Replace to replace the text in an editable field.

**Notes:**

---

## Import and Export Rungs

### Introduction

This chapter explains import and export of rungs.

<b>Topic</b>	<b>Page</b>
Export Considerations	23
Import Considerations	24

### Export Considerations

A single rung or a contiguous set of selected rungs may be exported to an L5X file.

The export file may also include any program-scoped tags, controller-scoped tags, Add-On Instructions, user-defined data types, and user-defined string types that are referenced by the rungs. The definitions for the referenced tags, Add-On Instructions, and user-defined data types are exported to the L5X file if they exist in the project. In the case of rungs exported from an Add-On Instruction routine, any parameters and local tags referenced will be exported if they exist at the time of export.

## Import Considerations

When importing rungs, you configure how the referenced components are imported during import configuration. By default, referenced components that collide with project components are not imported.

The following are considerations when importing rungs.

### Considerations when Importing Rungs

Topic	Consideration
Pending Edits Exist	If rungs are imported into a program or equipment phase that contains Pending Edits, all Pending Edits in the program will be accepted if Accept Program Edits is selected during import of the rungs. Similarly, all Pending Edits in the program will be finalized if Finalize All Edits In Program is selected during import of the rungs.
Accepted Edits Exist	Rungs may not be imported into a program or equipment phase that contains routines with Accepted Edits or Test Edits. Existing edits must first either be assembled or canceled.
First Scan	When importing rungs into an existing program, the S:FS bit will not be set during the program's next scan. This applies when importing rungs into an existing equipment phase as well.
Collision Handling	<p>If "Overwrite Selected Rungs" is checked in the Import Rungs dialog box, the imported rungs will overwrite the rungs selected in the project.</p> <p>If "Overwrite Selected Rungs" is unchecked, the imported rungs will be inserted before the selected rungs in the project if the ladder editor is in Insert mode or will be inserted after the selected rungs if the ladder editor is in Append mode.</p>
Tag Scope	<p>When rungs are exported from a program or equipment phase and imported into an Add-On Instruction routine, any referenced controller-scoped or program-scoped tags are converted on import. The tag will be converted to a local tag unless local scoped is not allowed (for example, a Motion Group tag can not be a local tag), in which case the tag will be converted to an InOut parameter.</p> <p>When rungs are exported from an Add-On Instruction routine and imported into a routine in a program or equipment phase, the referenced parameters and local tags are converted on import. The parameter or local tag will be converted to a program-scoped or phase-scoped tag unless it is not allowed (for example, a Motion Group tag cannot be a program-scoped tag), in which case it will be converted to a controller-scoped tag.</p>

For considerations for referenced user-defined types, Add-On Instructions, and tags that may be imported with the rungs see:

- [Chapter 5 Import and Export User-defined Types.](#)
- [Chapter 6 Import and Export Add-On Instructions.](#)
- [Chapter 7 Import and Export Tags.](#)



## Import and Export Routines

### Introduction

This chapter explains import and export of routines.

Topic	Page
Export Considerations	25
Import Considerations	25

### Export Considerations

A single routine may be exported to an L5X file. Routines of all language types, Function Block Diagram, Sequential Function Chart, Ladder Diagram and Structured Text, may be exported. However, routines may not be exported from an Add-On Instruction container and SoftLogix external routines may not be exported.

The export file may also include any program-scoped tags, controller-scoped tags, Add-On Instructions, user-defined data types, and user-defined string types that are referenced by the routine. The definitions for the referenced tags, Add-On Instructions, and user-defined data types are exported to the L5X file if they exist in the project.

### Import Considerations

When importing a routine, you configure how the referenced components are imported during import configuration. By default, referenced components that collide with project components are not imported.

The following are considerations when importing a routine.

#### Considerations when Importing a Routine

Topic	Consideration
Pending Edits Exist	If a routine is imported into a program or equipment phase that contains other routines with Pending Edits, all Pending Edits in the program will be accepted if Accept Program Edits is selected during import of the routine. Similarly, all Pending Edits in the program will be finalized if Finalize All Edits In Program is selected during import of the routine.
Accepted Edits Exist	A routine may not be imported into a program or equipment phase that contains routines with Accepted Edits or Test Edits. Existing edits must first either be assembled or canceled.
Routine Type	An existing routine may not be overwritten by a routine that is a different routine type.

### Considerations when Importing a Routine

Topic	Consideration
First Scan	When importing a routine into an existing program, the S:FS bit will not be set during the program's next scan. This applies when importing a routine into an existing equipment phase as well.
SFC Routine Execution Configuration	SFC execution settings are configured on a controller project, not per SFC routine. If you export an SFC routine and import it into another project with different SFC execution settings, the functionality of the routine could change. For example, a change in the Last Scan of Active Steps setting could leave physical outputs in an undesired state.
Equipment Phase State Routines	When state routines are created while online with the controller and logic edits are accepted but not tested, the routine will behave as if it was not implemented.

For considerations for referenced user-defined types, Add-On Instructions, and tags that may be imported with the routine see:

- [Chapter 5 Import and Export User-defined Types.](#)
- [Chapter 6 Import and Export Add-On Instructions.](#)
- [Chapter 7 Import and Export Tags.](#)

## Import and Export Programs and Equipment Phases

### Introduction

This chapter explains import and export of programs. All of the topics apply to equipment phases as well; any exceptions are noted.

Topic	Page
Export Considerations	27
Import Considerations	27

### Export Considerations

A single program may be exported to an L5X file. The exported program will include all of its program tags and routines as well. These will be imported with the program automatically.

The export file may also include any controller-scoped tags, Add-On Instructions, user-defined data types, and user-defined string types referenced by the program. The definitions for the referenced tags, Add-On Instructions, and user-defined data types are exported to the L5X file if they exist in the project.

As with other export types, I/O module data type definitions are not exported. I/O module data types are created within a project when the associated I/O module is created in the project. On import, program-scoped tags may not be modified. As a result, if a program-scoped tag aliases an I/O module type tag, the I/O module must first exist in the project in order to import the program. To export the program such that it can be imported into a project with a different module type, first alias the program-scoped tag to a controller-scoped tag of a non-I/O module type and then alias the controller-scoped tag to the I/O module. These program-scoped tags can then be created during the import of the program if necessary.

### Import Considerations

When importing a program, the program-scoped tags and routines are imported as part of the program. The Operation, Final Name, Description, and any other settings of the program-scoped tags and routines cannot be modified, instead the Operations are based on the Operation selected for the program.

Programs cannot overwrite equipment phases, and vice-versa. Programs and equipment phases must have unique names.

You configure how the referenced components are imported during import configuration. By default, referenced components that collide with project components are not imported.

The following are considerations when importing a program.

**Considerations when Importing a Program or Equipment Phase**

Topic	Consideration
Accepted or Test Edits Exist	A program with Accepted Edits or Test Edits may not be overwritten.
Deletes of Program-scoped Tags and Routines during Program Overwrite	When importing a program to replace an existing program, any tags or routines in the existing program that are not in the new program are deleted during import. However, if you are online and Import Logic Edits as Pending or Accept Program Edits is selected in the Online Options dialog, then these tags and routines cannot be deleted because they are referenced by existing logic until edits are finalized. In this situation, although they were identified during import configuration with an Operation of Delete, these tags and routines are not deleted as part of the import. You can delete them in the RSLogix 5000 editor after you finalize edits.
Safety Program Scheduled Location	A Safety program cannot be scheduled in the Controller Fault Handler or Power-Up Handler folders.
Configuration of Equipment Phase State Routines	<p>In the configuration for an equipment phase state routine, if the Complete State Immediately if not Implemented option is checked in RSLogix 5000 software version 17 and greater, an implemented, but empty (no logic), phase state routine behaves the same as an unimplemented phase state routine. The state immediately completes and execution of the phase continues. The phase then enters the next state in the state machine.</p> <p>In RSLogix 5000 software version 16, if an equipment phase enters a state for which a state routine exists, but contains no logic, execution of the phase stops regardless of whether the Complete State Immediately if not Implemented option is checked. The routine does complete, but there is no logic to execute.</p> <p>If you import a new state routine and, in the Online Options dialog, you select:</p> <ul style="list-style-type: none"> <li>• Import Logic Edits as Pending, an empty routine is created in the controller and the pending edits only exist in the offline project.</li> <li>• Accepts Program Edits, an empty routine is created in the controller and the logic is placed in a test edits container in the routine. If you are not actively testing edits, then the routine appears as empty when running.</li> <li>• Finalize All Edits in Program, the routine is created with the new logic and does not appear empty.</li> </ul> <p>In the first two cases, if the Complete State Immediately if not Implemented option is checked, the empty routine will complete immediately and allow phase execution to continue.</p>

**Considerations when Importing a Program or Equipment Phase**

<b>Topic</b>	<b>Consideration</b>
First Scan	<p>If a new program or equipment phase is created in a controller in Remote Run mode, logic in that program/equipment phase will see a value of 1 for the S:FS system flag (First Scan flag) until the main routine has executed once.</p> <p>Any other logic imports (that is overwriting an existing program/equipment phase, or any routine/rung imports) will not result in a value of 1 for S:FS system flag.</p>
Pre-scan	Logic imported while online with the controller in Remote Run mode will not be pre-scanned before it begins to execute.
Program Scheduled Location While Online	An imported program that is configured to overwrite an existing program cannot be scheduled into a different location than the existing program while online with the controller in Remote Run mode; the existing scheduled location will be used
Renamed Tags	<p>When overwriting an existing program and the imported program is modified such that a program-scoped tag has been renamed, during import the existing tag is deleted and then a tag with the new name is created and all logic references are updated to reference the new tag. Therefore, the online tag values are not preserved and the tag values from the imported tag will be downloaded to the controller. To preserve the data values of the renamed tag and minimize the logic changes, rename the program tag in the online project to the new name before importing the modified program.</p>

For considerations for referenced user-defined types, Add-On Instructions, and programs tags and referenced tags that may be imported with a program see:

- [Chapter 5 Import and Export User-defined Types.](#)
- [Chapter 6 Import and Export Add-On Instructions.](#)
- [Chapter 7 Import and Export Tags.](#)

**Notes:**

## Import and Export User-defined Types

### Introduction

This chapter explains import and export of user-defined data types and user-defined string types.

Topic	Page
Export Considerations	31
Import Considerations	31

### Export Considerations

A single user-defined type (either user-defined data type or user-defined string type) may be exported to an L5X file.

The export file may also include any Add-On Instructions, user-defined data types, and user-defined string types referenced by the exported user-defined type. The definitions for the referenced Add-On Instructions and data types are exported to the L5X file if they exist in the project and if 'Include all referenced Add-On Instructions and User-Defined Types' is checked during export.

User-defined data type references may also be exported when a program, routine, set of rungs, or Add-On Instruction is exported.

### Import Considerations

When importing a user-defined type, you configure how the referenced components are imported during import configuration. By default, referenced components that collide with project components are not imported.

User-defined types cannot overwrite Add-On Instructions. User-defined types and Add-On Instructions must have unique names.

The following are considerations when importing a user-defined type.

**Considerations when Importing a User-defined Type**

Topic	Consideration
Tag Data	<p>Imported tags that reference a user-defined data type in the import file may be affected if the user-defined data type is not imported as well. In this case, the imported tag's data may be converted if the existing data structure is different and tag data may be lost.</p> <p>If an existing user-defined data type is overwritten, project tag data may be converted if the data structure is different and tag data may be lost.</p>
Data Type Modification While Online	<p>A user-defined data type that is referenced in the project may not be overwritten. If the existing user-defined data type is not referenced, it may be overwritten while online.</p>
Final Name Change	<p>If the Final Name of a user-defined type reference is modified during import configuration, all logic, tags, Add-On Instructions, and other user-defined types in the import that reference the user-defined type will be updated to reference the new name. As a result, the edit date of any Add-On Instructions that references the user-defined type will be updated.</p>

For considerations for referenced Add-On Instructions that may be imported with the user-defined type see [Chapter 6 Import and Export Add-On Instructions](#).



## Import and Export Add-On Instructions

### Introduction

This chapter explains import and export of Add-On Instructions.

Topic	Page
Export Considerations	33
Import Considerations	34

### Export Considerations

A single Add-On Instruction may be exported to an L5X file. The exported Add-On Instruction will include all of its parameters, local tags, and routines as well. These will be imported with the Add-On Instruction automatically.

The export file may also include any Add-On Instructions, user-defined data types, and user-defined string types referenced by the Add-On Instruction. The definitions for the referenced Add-On Instructions and data types are exported to the L5X file if they exist in the project and if 'Include all referenced Add-On Instructions and User-Defined Types' is checked during export.

Add-On Instruction references may also be exported when a program, routine, set of rungs, or user-defined data type is exported.

## Import Considerations

**ATTENTION**



The EditedDate attribute of an Add-On Instruction must be updated if the Add-On Instruction is modified by editing an L5K or L5X file.

When Add-On Instruction logic is changed by manually editing an L5X file using a text editor and the EditedDate attribute is not changed, and if the Add-On Instruction is imported while online, the Add-On Instruction will be imported into the offline project but will not be updated in the controller.

Also, during import of an Add-On Instruction L5X file while online and overwriting an existing Add-On Instruction, if the the name or description of one of the Add-On Instruction Local tags is changed during the import operation (in the Import Configuration dialog box), the Add-On Instruction will be imported into the offline project but will not be updated in the controller.

In these cases, the changes to the Add-On Instruction will be seen at the local programming station, but will not be seen in the operation of the equipment. The equipment will continue to run in the defined state prior to the import of the Add-On Instruction.

---

When importing an Add-On Instruction, the parameters, local tags, and routines are imported as part of the Add-On Instruction. The Operation, Final Name, Description and any other settings of the Add-On Instruction's parameters, local tags, and routines cannot be modified, instead the Operations are based on the Operation selected for the Add-On Instruction.

You configure how the referenced components are imported during import configuration. By default, referenced components that collide with project components are not imported.

Add-On Instructions cannot overwrite user-defined types. Add-On Instructions and user-defined types must have unique names.

The following are considerations when importing an Add-On Instruction directly or as a reference.

### Considerations when Importing an Add-On Instruction

Topic	Consideration
Tag Data	<p>Imported tags that reference an Add-On Instruction in the import file may be affected if the Add-On Instruction is not imported as well. In this case, the imported tag's data may be converted if the existing Add-On Instruction's data structure is different and tag data may be lost.</p> <p>If an existing Add-On Instruction is overwritten, project tag data may be converted if the Add-On Instruction's data structure is different and tag data may be lost.</p>
Logic	<p>Imported logic that references the Add-On Instruction in the import file may be affected if the Add-On Instruction is not imported. If an existing Add-On Instruction is used for the imported logic reference and the parameter list of the Add-On Instruction in the project is different, the project may not verify or it may verify but not work as expected.</p> <p>If an existing Add-On Instruction is overwritten, logic in the project that references the Add-On Instruction may be affected. The project may not verify or may verify but not work as expected.</p>
Add-On Instructions While Online	<p>An Add-On Instruction cannot be overwritten during import while online with the controller, though a new Add-On Instruction may be created while online.</p>
Final Name Change	<p>If the Final Name of an Add-On Instruction is modified during import configuration, the edit date of the imported Add-On Instruction will be updated. In addition, all logic, tags, user-defined data types, and other Add-On Instructions in the import file that reference the Add-On Instruction will be updated to reference the new name. As a result, the edit date of any Add-On Instruction that references the Add-On Instruction will be updated.</p>

For considerations for referenced user-defined types that may be imported with the Add-On Instruction see [Chapter 5 Import and Export User-defined Types](#).

**Notes:**

## Import and Export Tags

### Introduction

This chapter explains import and export of referenced tags.

Topic	Page
Export Considerations	37
Import Considerations	37

### Export Considerations

Tags are not exported to an L5X file themselves, but they are exported to the L5X file as part of a program, equipment phase or Add-On Instruction export (program-scoped tags with a program export, equipment phase-scoped tags with an equipment phase export, and parameters and local tags with an Add-On Instruction export).

Tags may also be exported to an L5X file as references from another component being exported (controller-scoped tag references with a program or equipment phase export or program-scoped and controller-scoped tag references with a rung or routine export). The definitions for the referenced tags are exported to the L5X file by default if they exist in the project.

When exporting logic, especially if it is intended for general use, be aware that logic that references a bit member of a tag or member of a user-defined data type tag can not be replaced during import to reference a different bit or user-defined data type member. If you expect to connect the logic reference to a different bit or member of a tag, consider editing the logic before export such that the reference is to a full tag name (and alias if need be) so that the reference can be connected to the desired tag on import.

### Import Considerations

When importing a program, the program-scoped tags are imported with the program automatically (same for equipment phase). When importing an Add-On Instruction, the parameters and local tags are imported with the Add-On Instruction automatically.

During program or equipment phase imports, referenced controller-scoped tags may also be imported. For routine and rung imports, referenced tags may be imported as well. You configure how referenced tags are imported during import configuration. By default, referenced tags that collide with project components are not imported.

**Considerations when Importing Tags**

Topic	Consideration
Tag Data	<p>Imported tags that reference an Add-On Instruction or user-defined data type in the import file may be affected if the Add-On Instruction or user-defined data type is not imported as well. In this case, the imported tag's data may be converted if the existing data structure is different and tag data may be lost.</p> <p>If an existing Add-On Instruction or user-defined data type is overwritten, project tag data may be converted if the data structure is different and tag data may be lost.</p> <p>If the data is not convertible, it will be overwritten with the default values for the type. When array dimensions are changed, existing array members retain their values and descriptions and new members have the default values and description (usually 0 and no description).</p>
Consumed Tags	<p>Consumed tags cannot be imported from an L5X file. They will be converted to base tags and a warning will be emitted to the Errors/Results Pane during initial parsing of the L5X file.</p>
Tag Values While Online	<p>When importing into a controller while online, if existing tags are being overwritten by imported tags, the tag values will not be written to the controller. Tag values are written only to the offline project. The tag values in the controller will maintain their current values but other tag attributes will be written to the controller.</p> <p>Tags that are created during import will have their values written to the controller.</p>
Tag Scope	<p>An import tag maintains the scope of the tag as it was when exported if the tag initially collides with a different scoped tag in the project. In that case, an attention (red) flag will be present on the tag indicating the scope collision. However, If you change the Final Name of a tag so that it subsequently collides with a tag of a different scope in the project, the imported tag will be changed to the scope of the existing tag.</p> <p>You can resolve the attention flag that appears on initial collision due to a scope issue by changing the Final Name to avoid the collision with that tag or, if the import component is routine or rungs, you can change the scope of the import tag by right-clicking on the tag row and selecting Toggle Tag Scope.</p>

**A**

- Add-On Instruction** 33
  - export 6
  - export considerations 33
  - import 8
  - import considerations 34
- attention flags** 16

**C**

- canceled import** 16
- comment**
  - translated 19
- component**
  - configuration 9
  - export 5, 6
  - import 5, 8
- configuration**
  - attention flags 16
  - components 9
  - example 14
  - import 9, 12
  - options 10, 28
  - references 12
  - undefined references 15
- considerations**
  - export 6
  - import 16
  - language switching 19
  - motion 18
  - online import 17
  - safety 18
  - security 18

**D**

- description**
  - export 6
  - translated 19
- documentation language** 19

**E**

- equipment phase** 27
  - export 6
  - export considerations 27
  - import 8
  - import considerations 27
- export**
  - Add-On Instruction 33
  - component description 6
  - components 5, 6
  - considerations 6

**export (continued)**

- equipment phase 27
- L5X 5, 6
- local tag 37
- parameter 37
- program 27
- routine 25
- rungs 23
- tag 37
- user-defined data type 31
- user-defined string type 31

**export considerations**

- Add-On Instruction 33
- equipment phase 27
- local tag 37
- parameter 37
- program 27
- routine 25
- rungs 23
- tag 37
- user-defined data type 31
- user-defined string type 31

**F**

- failed import** 16
- find**
  - text 21

**I**

- import**
  - Add-On Instruction 33
  - canceled 16
  - components 5, 8
  - configuration 9, 12
  - considerations 16
  - equipment phase 27
  - failed 16
  - L5X 5, 8
  - local tag 37
  - parameter 37
  - program 27
  - routine 25
  - rungs 23
  - tag 37
  - user-defined data type 31
  - user-defined string type 31
- import considerations**
  - Add-On Instruction 34
  - equipment phase 27
  - local tag 37
  - parameter 37
  - program 27

**import considerations (continued)**

routine 25  
rungs 24  
tag 37  
user-defined data type 31  
user-defined string type 31

**L****L5X**

export 5, 6  
import 5, 8

**language switching 19****local tag**

export considerations 37  
import considerations 37

**M**

**motion 18**

**O**

**online import 17**

**P****parameter**

export considerations 37  
import considerations 37

**program 27**

export 6  
export considerations 27  
import 8  
import considerations 27

**R****references**

configuration 12  
undefined 15

**replace**

text 21

**routine 25**

export 6  
export considerations 25  
import 8  
import considerations 25

**rungs 23**

export 6  
export considerations 23  
import 8  
import considerations 24

**S**

**safety 18**

**search and replace 21**

**security 18**

**T****tag 37**

export considerations 37  
import considerations 37

**tooltip**

attention flags 16

**trend**

export 6  
import 8

**U**

**undefined references 15**

**user-defined data type 31**

export 6  
export considerations 31  
import 8  
import considerations 31

**user-defined string type 31**

export 6  
export considerations 31  
import 8  
import considerations 31



# How Are We Doing?

Your comments on our technical publications will help us serve you better in the future. Thank you for taking the time to provide us feedback.

You can complete this form and mail (or fax) it back to us or email us at [RADocumentComments@ra.rockwell.com](mailto:RADocumentComments@ra.rockwell.com).

Pub. Title/Type Import/Export Project Components/Programming Manual

Cat. No. Logix Controllers Pub. No. 1756-P019B-EN-P Pub. Date November 2008 Part No. N/A

Please complete the sections below. Where applicable, rank the feature (1=needs improvement, 2=satisfactory, and 3=outstanding).

<b>Overall Usefulness</b> 1    2    3	How can we make this publication more useful for you?		
<b>Completeness</b> (all necessary information is provided)    1    2    3	Can we add more information to help you?		
	procedure/step	illustration	feature
	example	guideline	other
	explanation	definition	
<b>Technical Accuracy</b> (all provided information is correct)    1    2    3	Can we be more accurate?		
	text	illustration	
<b>Clarity</b> (all provided information is easy to understand)    1    2    3	How can we make things clearer?		
<b>Other Comments</b>	You can add additional comments on the back of this form.		

Your Name \_\_\_\_\_  
 Your Title/Function \_\_\_\_\_  
 Location/Phone \_\_\_\_\_

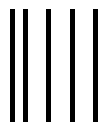
Would you like us to contact you regarding your comments?  
 No, there is no need to contact me  
 Yes, please call me  
 Yes, please email me at \_\_\_\_\_  
 Yes, please contact me via \_\_\_\_\_

Return this form to: Rockwell Automation Technical Communications, 1 Allen-Bradley Dr., Mayfield Hts., OH 44124-9705  
 Fax: 440-646-3525    Email: [RADocumentComments@ra.rockwell.com](mailto:RADocumentComments@ra.rockwell.com)

PLEASE FASTEN HERE (DO NOT STAPLE)

Other Comments

PLEASE FOLD HERE



NO POSTAGE  
NECESSARY  
IF MAILED  
IN THE  
UNITED STATES

PLEASE REMOVE

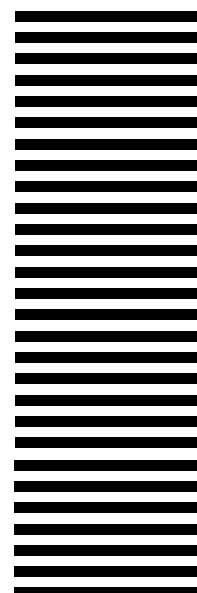
**BUSINESS REPLY MAIL**

FIRST-CLASS MAIL PERMIT NO. 18235 CLEVELAND OH

POSTAGE WILL BE PAID BY THE ADDRESSEE

**Rockwell  
Automation**

1 ALLEN-BRADLEY DR  
MAYFIELD HEIGHTS OH 44124-9705



## Notes:

# Rockwell Automation Support

Rockwell Automation provides technical information on the Web to assist you in using its products. At <http://support.rockwellautomation.com>, you can find technical manuals, a knowledge base of FAQs, technical and application notes, sample code and links to software service packs, and a MySupport feature that you can customize to make the best use of these tools.

For an additional level of technical phone support for installation, configuration, and troubleshooting, we offer TechConnect support programs. For more information, contact your local distributor or Rockwell Automation representative, or visit <http://support.rockwellautomation.com>.

## Installation Assistance

If you experience a problem within the first 24 hours of installation, please review the information that's contained in this manual. You can also contact a special Customer Support number for initial help in getting your product up and running.

United States	1.440.646.3434 Monday – Friday, 8am – 5pm EST
Outside United States	Please contact your local Rockwell Automation representative for any technical support issues.

## New Product Satisfaction Return

Rockwell Automation tests all of its products to ensure that they are fully operational when shipped from the manufacturing facility. However, if your product is not functioning and needs to be returned, follow these procedures.

United States	Contact your distributor. You must provide a Customer Support case number (call the phone number above to obtain one) to your distributor in order to complete the return process.
Outside United States	Please contact your local Rockwell Automation representative for the return procedure.

[www.rockwellautomation.com](http://www.rockwellautomation.com)

### Power, Control and Information Solutions Headquarters

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, Vorstlaan/Boulevard du Souverain 36, 1170 Brussels, Belgium, Tel: (32) 2 663 0600, Fax: (32) 2 663 0640

Asia Pacific: Rockwell Automation, Level 14, Core F, Cyberport 3, 100 Cyberport Road, Hong Kong, Tel: (852) 2887 4788, Fax: (852) 2508 1846