

PhaseManager Software

1756 ControlLogix, 1769 CompactLogix, 1789 SoftLogix, 1794 FlexLogix, 20D PowerFlex 700S with DriveLogix



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 [SGL-1.1](#) available from your local Rockwell Automation sales office or online at <http://www.rockwellautomation.com/literature/>) 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.



WARNING: 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.



ATTENTION: 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



SHOCK HAZARD: Labels may be on or inside the equipment, for example, a drive or motor, to alert people that dangerous voltage may be present.



BURN HAZARD: Labels may be on or inside the equipment, for example, a drive or motor, to alert people that surfaces may reach dangerous temperatures.

IMPORTANT Identifies information that is critical for successful application and understanding of the product.

Allen-Bradley, Rockwell Software, Rockwell Automation, SoftLogix, FlexLogix, CompactLogix, ControlLogix, DriveLogix, PhaseManager, Powerflex 700S, Logix5000, Logix5550, PLC-5, SLC 500, SoftLogix5800, FactoryTalk Batch, RSLogix 5000, and RSBizWare Batch are trademarks of Rockwell Automation, Inc.

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

Preface	Summary of Changes	5
	Additional Resources	5
	Purpose of This Manual	6
	Who Should Use This Manual	7
	How To Use This Manual	7
	 Chapter 1	
Introduction	PhaseManager Overview	9
	State Model Overview	12
	Equipment States	13
	State Transitions	14
	Manually Change State	15
	Ownership	15
	Comparison of Other State Models	16
	 Chapter 2	
PhaseManager Quick Start	Purpose of This Chapter	17
	Equipment	17
	Create an Equipment Phase	18
	Create a State Routine	18
	Manually Step Through the States	19
	Configure the Initial State for an Equipment Phase	22
	 Chapter 3	
Guidelines	Purpose of This Chapter	23
	Equipment Model Guidelines	24
	Example 1: Tank	25
	Example 2: Smart Belt	25
	State Model Guidelines	26
	State Model Worksheet	28
	Example 1: Add Water	29
	Example 2: Space Parts	30
	Equipment Code Guidelines	31
	Example 1: Add Water to a Tank	32
	Example 2: Smart Belt	33
	Execution Guidelines	34
	Example 1: Add Water to a Tank	39
	Example 2: Smart Belt	40
	Transition Guidelines	41
	Example 1: Tank	45
	Example 2: Smart Belt	46
	Example 3: Jam Detection	47
	State Completion Guidelines	48
	Example 1: Add Water to a Tank	50
	Example 2: Smart Belt	50

	Equipment Interface Tag Guidelines	51
	Additional Resources	52
	Example 1: Add Water to a Tank	53
	Example 2: Smart Belt	54
	Example 2: Smart belt, Continued	55
	Alias Tag Guidelines	56
	Example	56
	Additional Resources	56
	AppendixA	
PHASE Data Type	Introduction	57
	Set and Clear Equipment Phase Tag Values	57
	PHASE Data Type	58
	AppendixB	
Configure an Equipment Phase	Introduction	63
	Open the Configuration for an Equipment Phase	63
	Equipment Phase Settings	64
Glossary	67
Index	69

Summary of Changes

This revised document removes Equipment Phase instructions and updates cross-references to the Logix5000™ Controllers Advanced Process Control and Drives Instructions Reference Manual, publication [1756-RM006](#).

For the latest compatible software information, see the Product Compatibility and Download Center at <http://www.rockwellautomation.com/rockwellautomation/support/pcdc>.

Additional Resources

These documents contain additional information that concern-related products from Rockwell Automation.

Table 1 - Additional Resources

Resource	Description
Logix5000 Controllers Quick Start, publication 1756-QS001	Get started with a Logix5000 controller
Logix5000 Controllers Common Procedures, publication 1756-PM001	Program a Logix5000 controller—detailed and comprehensive information
PhaseManager™ User Manual, publication LOGIX-UM001	<ul style="list-style-type: none"> • Use equipment phases • Configure a state model for your equipment • Program in a way that is similar to S88 and PackML models
Logix5000 Controllers General Instructions Reference Manual, publication 1756-RM003	Program a specific Logix5000 programming instruction
Logix5000 Controllers Process and Drives Instructions Reference Manual, publication 1756-RM006	
Logix5000 Controllers Motion Instructions Reference Manual, publication MOTION-RM002	
Logix5000 Controllers Import/Export Reference Manual, publication 1756-RM084	Import or export a Logix5000 project or tags from or to a text file
Converting PLC-5 or SLC 500 Logix to Logix5550® Logic Reference Manual, publication 1756-RM085	Convert a PLC-5® or SLC™ 500 application to a Logix5000 project

Table 1 - Additional Resources

Resource	Description
CompactLogix™ Controllers User manual, publication 1769-UM007	Use a specific Logix5000 controller
ControlLogix System User Manual, publication 1756-UM001	
DriveLogix™ System 5720 User Manual, publication 20D-UM002	
DriveLogix5730 Controller for PowerFlex® 700S Drives with PhaseII Control User Manual, publication 20D-UM003	
FlexLogix™ Controllers User Manual, publication 1794-UM001	
SoftLogix5800 System User Manual, publication 1789-UM002	
EtherNet/IP Modules in Logix5000 Control Systems User Manual, publication ENET-UM001	Control devices over an EtherNet/IP network
ControlNet Modules in Logix5000 Control Systems User Manual, publication CNET-UM001	Control devices over a ControlNet network
DeviceNet Modules in Logix5000 Control Systems User Manual, publication DNET-UM004	Control devices over a DeviceNet network

Purpose of This Manual

This manual shows you how to configure and program a Logix5000 controller to use equipment phases. It gives you guidance and examples to:

- Lay out your code in sections that include equipment phases.
- Configure a state model for your equipment.
- Program your equipment to run by the state model.
- Use equipment phase instructions to transition to another state, handle faults, create break points, and so forth.

A Logix5000 controller is any of the following:

- 1756 ControlLogix controllers
- 1769 CompactLogix controllers
- 1789 SoftLogix5800 controllers
- 1794 FlexLogix controllers
- 20D PowerFlex 700S with DriveLogix controllers

Who Should Use This Manual

This manual is for employees who program or maintain industrial automation systems.

To use this manual, you must already have experience with the following:

- Programmable controllers
- Industrial automation systems
- Personal computers

How To Use This Manual

As you use this manual, text that is courier identifies information that you must supply based on your application (a variable). For example, 'Right-click name_of_program ...' means that you must identify the specific program in your application. Typically, it is a name or variable that you have defined.

Notes:

Introduction

PhaseManager Overview

PhaseManager™ software adds equipment phases to a controller. An equipment phase makes it easier to write, use, and manage the code for your machine or equipment.

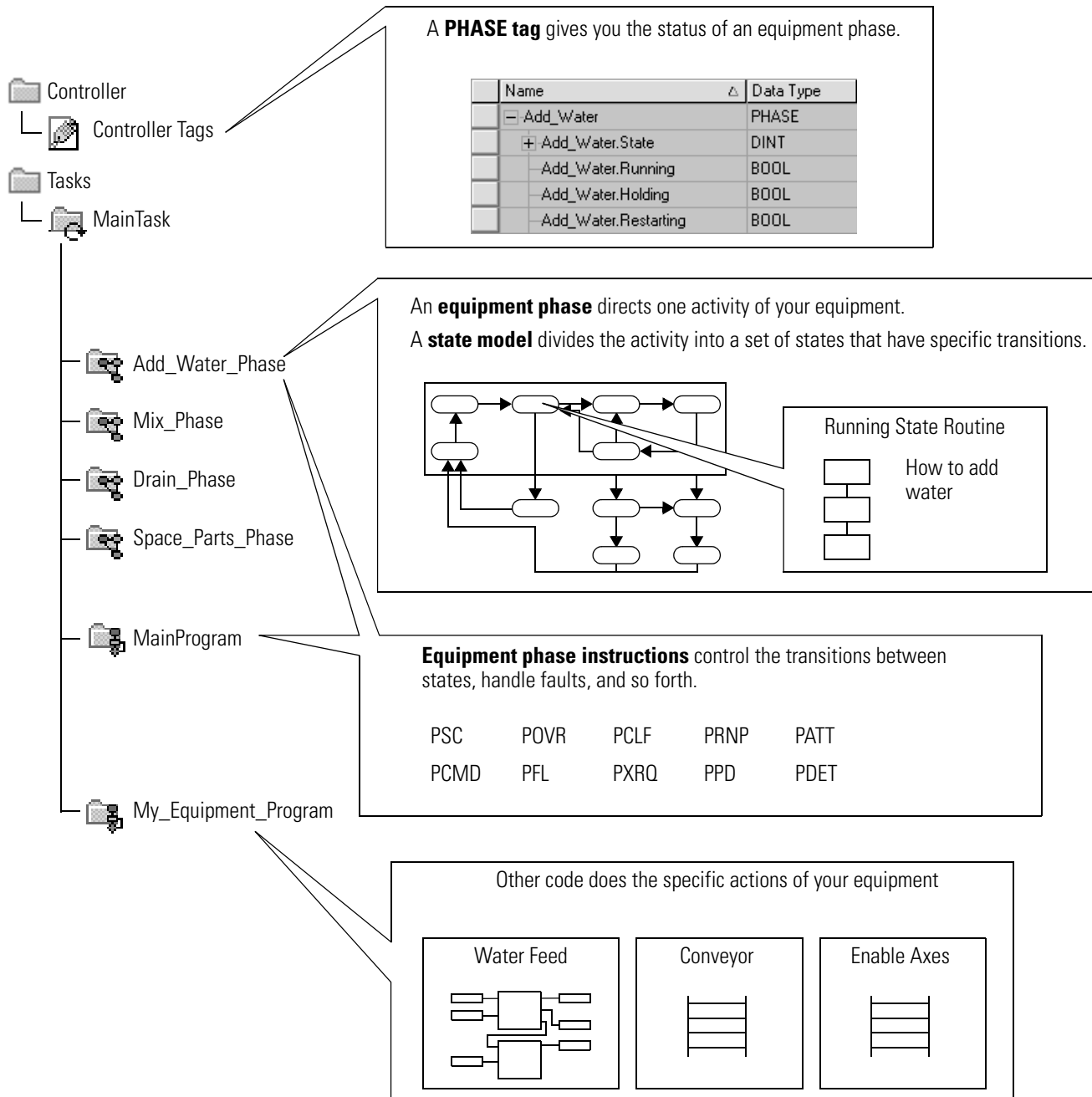


Table 2 - PhaseManager Terms

Term	Description
Equipment phase	<p>An equipment phase is similar to a program:</p> <ul style="list-style-type: none"> • You run the equipment phase in a task. • You give the equipment phase a set of routines and tags. <p>An equipment phase differs from a program in these ways:</p> <ul style="list-style-type: none"> • The equipment phase uses a state model. • Use an equipment phase to do one activity of your equipment.
State model	<p>A state model divides the operating cycle of your equipment into a set of states. Each state is an instant in the operation of the equipment. It's the actions or conditions of the equipment at a given time.</p> <p>The state model of an equipment phase is similar to these state models:</p> <ul style="list-style-type: none"> • U.S. standard ISA S88.01-1995 and its IEC equivalent IEC 61512-1-1998, commonly referred to as S88 • PackML, which was previously under the supervision of OMAC but is now a working group within ISA
State machine	<p>The controller has an embedded state machine for the equipment phase. This machine makes it a lot easier to use the state model. The state machine:</p> <ul style="list-style-type: none"> • Calls the main routine (state routine) for an acting state. • Manages the transitions between states with minimal coding. <p>You code the transition conditions. When the conditions are true, the state machine transitions the equipment to the next required state.</p> <ul style="list-style-type: none"> • Makes sure that the equipment goes from state to state along an allowable path. <p>For example, if the equipment is in the Complete or Stopped state, the equipment phase makes sure that it goes only to the Resetting state. This functionality simplifies the amount of interlocking that you have to do.</p>
Equipment phase instructions	<p>Specific instructions that you use to control an equipment phase. See Logix5000™ Controllers Advanced Process Control and Drives Instructions Reference Manual, Publication 1756-RM006.</p>
PHASE tag	<p>When you add an equipment phase, RSLogix 5000® software makes a tag for the equipment phase. The tag uses the PHASE data type. Use the tag to:</p> <ul style="list-style-type: none"> • See which state the equipment phase is in. • Hold a failure code for the equipment phase. • Hold an index for your steps. • Hold the unit ID. • See the status of an external request to FactoryTalk® Batch software. • See if FactoryTalk Batch software has new parameters for the equipment phase. • Create producing and standby states. <p>See Appendix A for more information about the PHASE data type.</p>

PhaseManager software helps you write the code for your equipment in a structured way. This structure results in the same behavior for all equipment across a plant.

PhaseManager Questions and Answers

Question	Answer
How can I get the highest performance possible from my equipment?	<p>You have to measure equipment performance to improve it. The state model gives you a way to measure the status of your equipment. With that data, you can calculate the efficiency and performance measures that you want.</p> <p>If you use PhaseManager software across your plant, you have consistent data from equipment to equipment.</p>
How can I cut the cost of integrating my equipment into the plant?	Clear structure and consistent tags make it a lot easier to plug the equipment into your plant and configure communication right away. Equipment up and down that line share data that uses the same tag names. And all equipment communicates with higher-level systems in the same way.
How can I make it easier to maintain the code?	A state model helps you lay out the general functions of your equipment. We found that programmers prefer a state model as the heart of their code. A state model serves as a map for the code. With a clear structure, you know just where to look for the piece of code that you want.
How can I give my operators a clean, intuitive HMI?	A state model lets you make all your equipment behave the same. Your HMIs can then show consistent equipment conditions across the plant. When an HMI says that the equipment is in an idle, run, or hold state, your operators know exactly what the message means.

State Model Overview

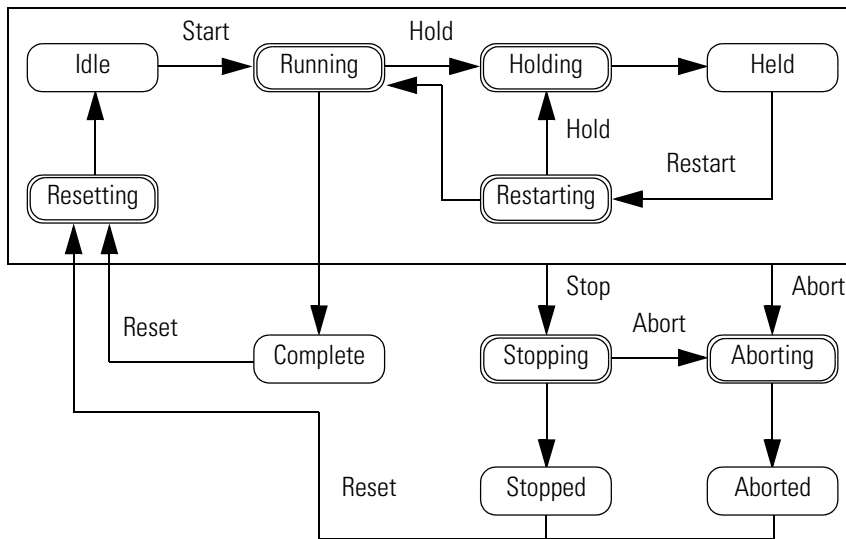
A state model divides the operating cycle of your equipment into a series of states. Each state is an instant in the operation of the equipment. It's the actions or conditions of the equipment at a given time.

In a state model, you define what your equipment does under different conditions, such as run, hold, and stop. You are not required to use all states for your equipment. Use only the states that you want.

There are two types of states.

Type of State	Description
Acting	Does something or several things for a certain time or until certain conditions are met. An acting state runs one time or repeatedly.
Waiting	Shows that certain conditions are met and the equipment is waiting for the signal to go to the next state.

PhaseManager software uses the following states.



Your equipment can go from any state in the box to the stopping or aborting state.

Acting

Acting states represent the things that your equipment does at a given time.

Waiting

Waiting states represent the condition of your equipment when it is in-between acting states.

One common objection to a state model is that it doesn't fit all equipment. You could hear or think: 'My equipment is complex. There's much synchronization and many things happen in parallel.'

Keep in mind that a state model views your equipment at a general level. Different equipment does different things and needs specific code for everything it does. A state model simply gives you a higher-level framework for your code.

- The state model defines the general behavior, commands, and status of the equipment.
- You program the details of the equipment within that framework.

Equipment States

The use of a state model can sound like a significant change for programmers. But it simply represents another way to view the same control problem.

With a state model, you define the behavior of your equipment and put it into a brief functional specification. In this way, you show what happens and when it happens.

For this State	Ask
Stopped	What happens when you turn on power?
Resetting	How does the equipment get ready to run?
Idle	How do you tell that the equipment is ready to run?
Running	What does the equipment do to make product?
Holding	How does the equipment temporarily stop the production of product without making scrap?
Held	How do you tell if the equipment is safely holding?
Restarting	How does the equipment resume production after holding?
Complete	How do you tell when the equipment is done with what it had to do?
Stopping	What happens during a normal shutdown?
Aborting	How does the equipment shutdown if a fault or failure happens?
Aborted	How do you tell if the equipment is safely shut down?

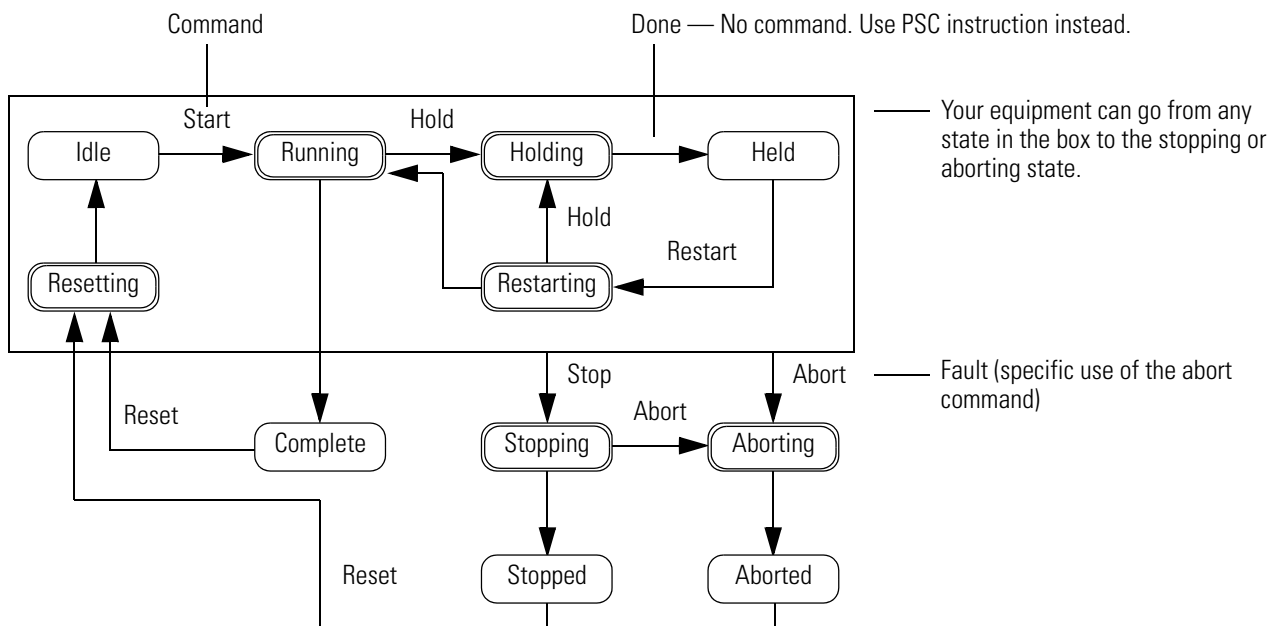
State Transitions

The arrows in the state model show to which states your equipment can go from the state it is in now.

- Each arrow is called a transition.
- A state model lets the equipment make only certain transitions. This functionality gives the equipment the same behavior as any other equipment that uses the same model.

PhaseManager software uses the following transitions.

→ = transition

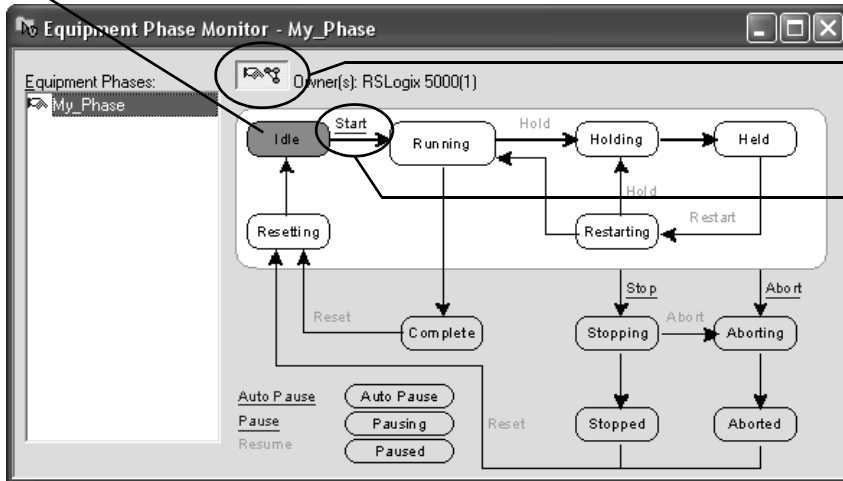


Type of Transition	Description						
Command	<p>A command tells the equipment to start doing something or do something different. For example, the operator pushes the start button to start production and the stop button to shut down.</p> <p>PhaseManager software uses these commands:</p> <table border="0"> <tr> <td>Reset</td> <td>Stop</td> <td>Restart</td> </tr> <tr> <td>Start</td> <td>Hold</td> <td>Abort</td> </tr> </table>	Reset	Stop	Restart	Start	Hold	Abort
Reset	Stop	Restart					
Start	Hold	Abort					
Done	Equipment goes to a waiting state when it's done with what it's doing. You don't give the equipment a command. Instead, you design your code to signal when the equipment is done. The waiting state shows that the equipment is done.						
Fault	A fault tells you that something out of the ordinary has happened. You design your code to look for faults and act if it finds any. Suppose that you want your equipment to shut down as fast as possible if a certain fault happens. In that case, design your code to look for that fault and give the abort command if it finds it.						

Manually Change State

RSLogix 5000 software has a window that lets you monitor and command an equipment phase.

State that the equipment phase is in right now



To change states manually.

1. Take ownership of the equipment phase.

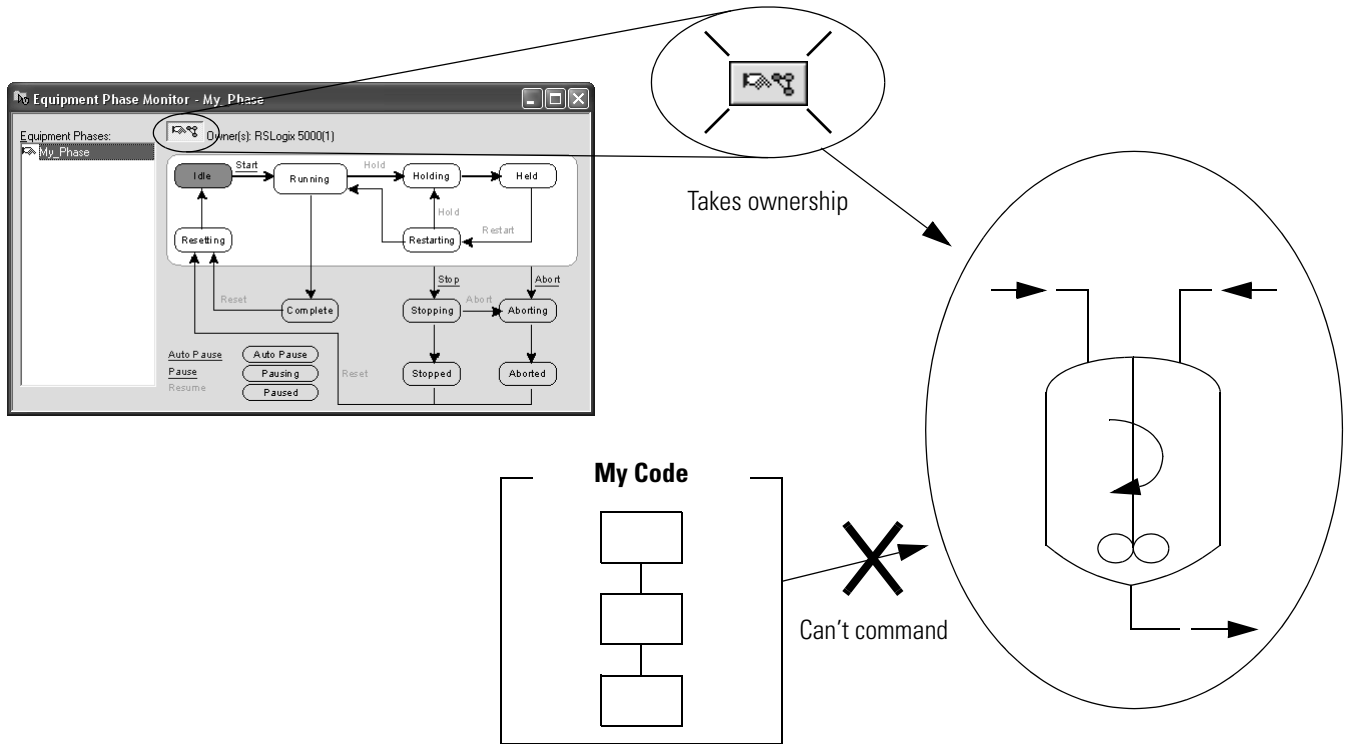
2. Give a command.

Ownership

Ownership locks out programs or FactoryTalk Batch software from giving commands to an equipment phase.

If this component owns the equipment phase	Then
RSLogix 5000 software	Sequencers can't give commands to the equipment phase. These sequencers include: <ul style="list-style-type: none"> Internal sequencer — program in the controller. External sequencer — FactoryTalk Batch software.
Internal sequencer — program in the controller	Other sequencers can't give commands to the equipment phase.
External sequencer — FactoryTalk Batch software	Other sequencers can't give commands to the equipment phase.

Exception: Use an Equipment Phase Override Command (POVR) instruction to give a hold, stop, or abort command regardless of ownership.



See the Logix5000™ Controllers Advanced Process Control and Drives Instructions Reference Manual, publication [1756-RM006](#) for the following information:

- Equipment Phase Command (PCMD)
- Equipment Phase Override Command (POVR)
- Attach to Equipment Phase (PATT) instruction

Comparison of Other State Models

The following table compares the PhaseManager software state model to other common state models.

S88	PackML	PhaseManager Software
Idle	Starting > Ready	Resetting > Idle
Running > Complete	Producing	Running > Complete
Pausing > Paused	Standby	Subroutines, breakpoints, or both.
Holding > Held	Holding > Held	Holding > Held
Restarting	None	Restarting
Stopping > Stopped	Stopping > Stopped	Stopping > Stopped
Aborting > Aborted	Aborting > Aborted	Aborting > Aborted

PhaseManager Quick Start

Purpose of This Chapter

Use this quick start to:

- Get an introduction to how an equipment phase runs.
- Monitor an equipment phase.
- Manually tell an equipment phase to go to another state.

Use this quick start when you want to:

- Try out PhaseManager™ software for the first time.
- Test an equipment phase by manually stepping through its states.

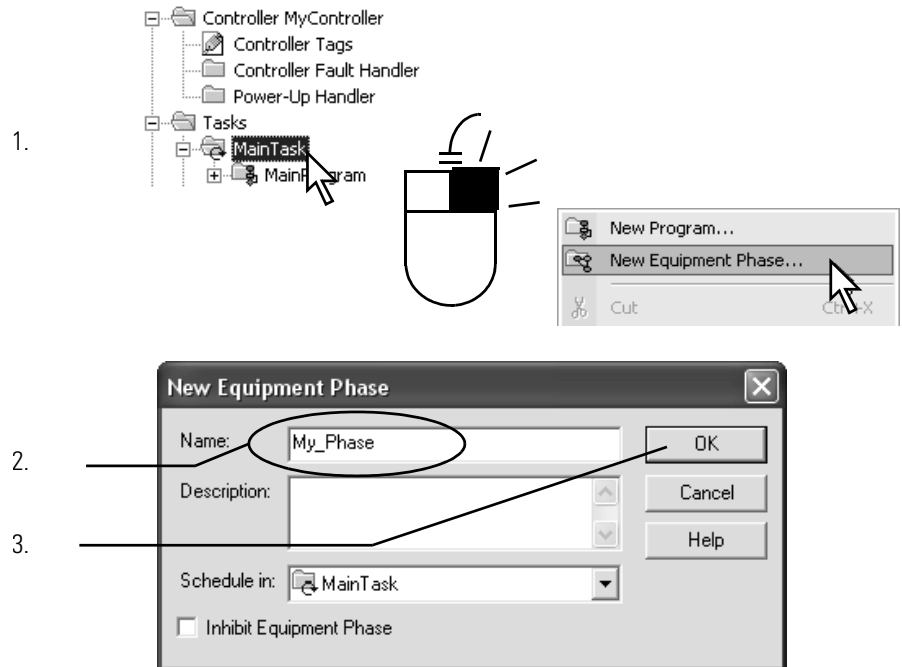
Topic	Page
Create an Equipment Phase	18
Create a State Routine	18
Manually Step Through the States	19
Configure the Initial State for an Equipment Phase	22

Equipment

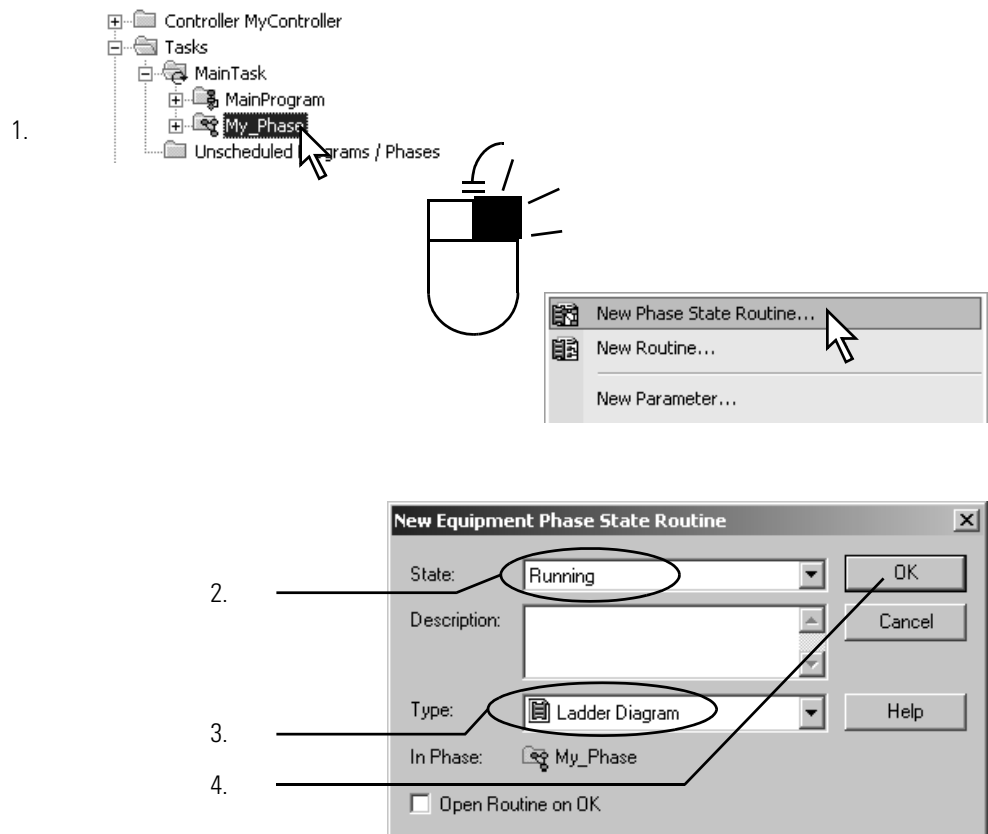
To use this quick start, you need:

- A Logix5000™ controller. See the preface if you aren't sure which controllers are Logix5000 controllers.
- Firmware, revision 18.0 or later, for the controller
- A power supply for the controller
- A communication path to the controller:
 - Communication card or built-in port
 - Corresponding communication cable
- RSLogix 5000® software, version 18.0 or later

Create an Equipment Phase



Create a State Routine

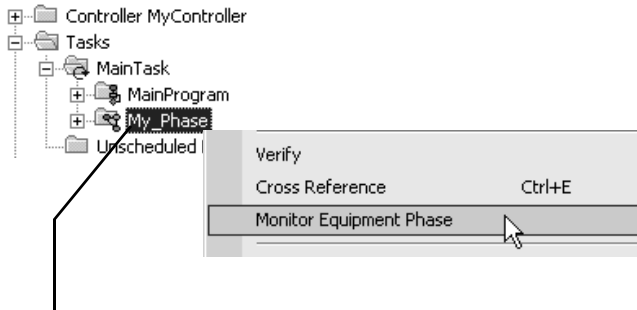


Manually Step Through the States

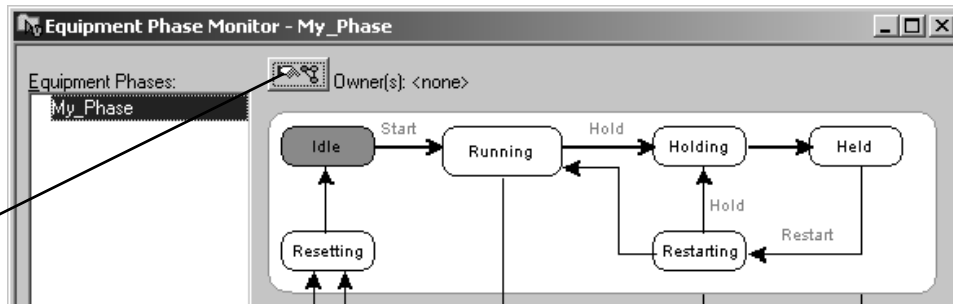
Before you do this procedure, do the following:

- Download the project to the controller.
- Put the controller in Run or Remote Run mode.

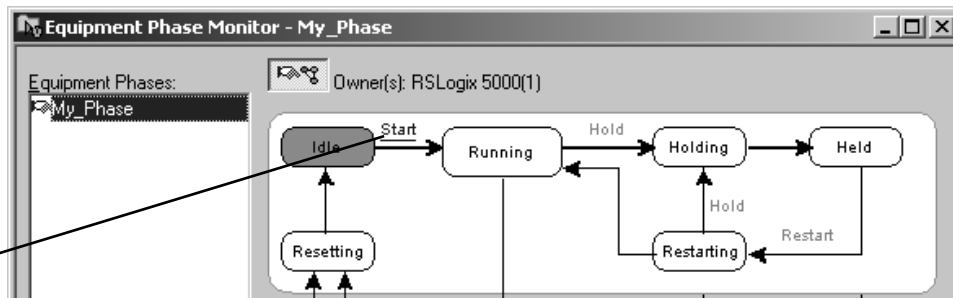
Step	Notes
------	-------



1. Right-click the equipment phase and choose Monitor Equipment Phase.

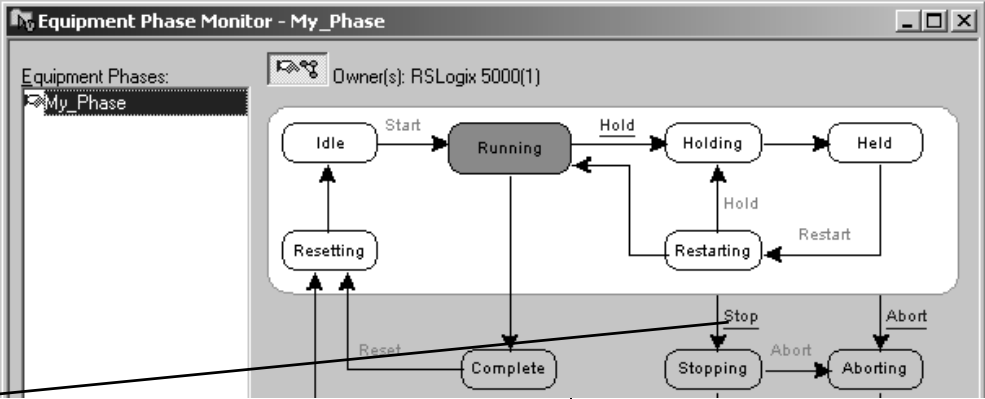


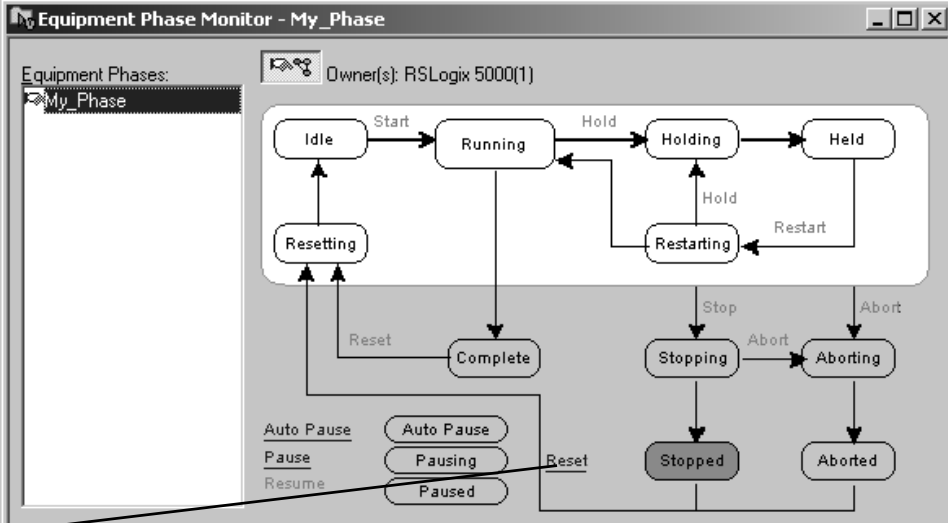
2. Click the ownership button and then Yes—take ownership. This action lets you use this window to step through the states.



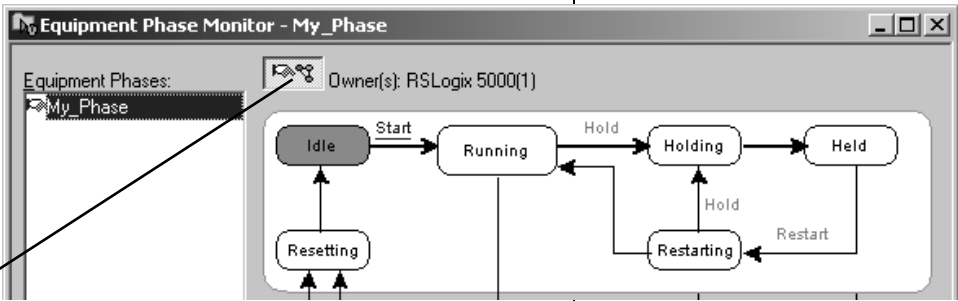
3. Click Start.

- The equipment phase goes to the Running state.
- Any code in the Running state routine starts running. This routine is where you put the code for the normal production sequence of your equipment.

Step	Notes
<p>4. Click Stop.</p>	 <ul style="list-style-type: none"> The equipment phase goes to the Stopped state. The Running state routine stops running. The Stopping state routine is optional. Without it, the equipment phase goes directly to the Stopped state.

<p>5. Click Reset.</p>	 <ul style="list-style-type: none"> The equipment phase goes to the Idle state. The Resetting state routine is optional. Without it, the equipment phase goes directly to the Idle state.
------------------------	---

Step	Notes
------	-------

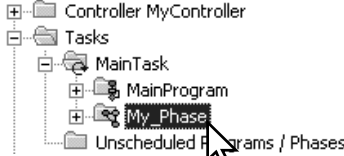


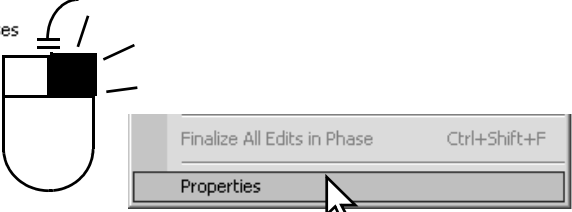
6. Click the ownership button.

This action releases the equipment phase from control by this window.

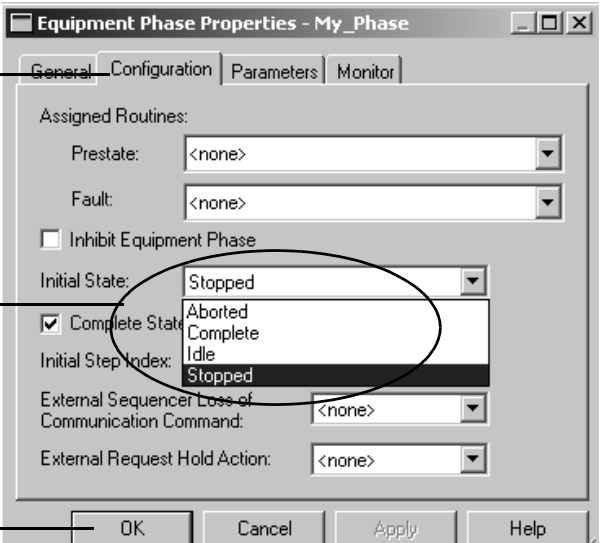
Configure the Initial State for an Equipment Phase

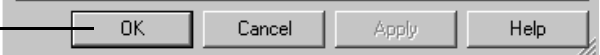
The initial state is the first state to which the equipment phase goes after power-up.

- 

1. In the project tree, select the **My_Phase** folder under **MainTask**.
- 

2. Right-click on the selected **My_Phase** folder and select **Properties**.
3. Choose your initial state.



The **Equipment Phase Properties - My_Phase** dialog box is shown with the **General** tab selected. The **Assigned Routines** section includes **Prestate** and **Fault** dropdown menus, both set to **<none>**. The **Inhibit Equipment Phase** checkbox is unchecked. The **Initial State** dropdown menu is open, showing a list of states: **Stopped**, **Aborted**, **Complete**, **Idle**, and **Stopped**. The **Complete State** checkbox is checked. The **Initial Step Index** dropdown menu is also open, showing the same list of states. The **External Sequencer Loss of Communication Command** and **External Request Hold Action** dropdown menus are both set to **<none>**.
- 

4. Click **OK** to apply the changes.

Guidelines

Purpose of This Chapter

This chapter guides your development and programming of a Logix5000™ project that uses equipment phases. Use the procedures for the following:

- Before you lay out the equipment phases for your Logix5000 project.
- As a reference while you program the project.

Review the following guidelines before you lay out your project. Refer to these guidelines as needed.

Topic	Page
Equipment Model Guidelines	24
State Model Guidelines	26
Equipment Code Guidelines	31
Execution Guidelines	34
Transition Guidelines	40
State Completion Guidelines	47
Equipment Interface Tag Guidelines	50
Alias Tag Guidelines	55

Equipment Model Guidelines

Each equipment phase is a specific activity that your equipment does. An equipment phase tells the equipment what to do and when to do it.

Follow these guidelines to decide how many equipment phases to use.

Guideline	Details	
Make sure that each equipment phase does an independent activity.	Make sure that each equipment phase does an activity that is independent (relatively independent) from other equipment. The equipment phase commands all equipment that works together to do the specific activity.	
	Example	
	This activity is probably an equipment phase	This activity is probably NOT an equipment phase
	<ul style="list-style-type: none"> • Fill bottles with product. • Put bottles in carton. • Add water to a tank. • Mix ingredients in tank 	<ul style="list-style-type: none"> • Accelerate filler axis (too small) • Run bottling line (too large) • Open water valve (too small) • Brew ingredients (too large)
Keep the number of equipment phases and programs within the following limits.	If you have this controller	You can have up to
	ControlLogix®	100 programs and equipment phases per task
	SoftLogix™	100 programs and equipment phases per task
	FlexLogix™	32 programs and equipment phases per task
	CompactLogix™	32 programs and equipment phases per task
List the equipment that goes along with each equipment phase.	Example	
	This equipment phase	Relates this equipment
	Add_Water	Water pump Water valve Limit switch
Smart_Belt	Coarse belt axis Fine belt axis Exit belt axis	

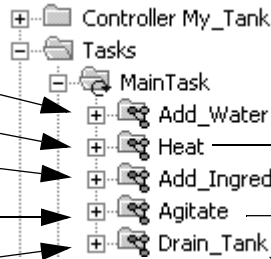
Example 1: Tank

This example shows the equipment phases for a tank that cooks ingredients.

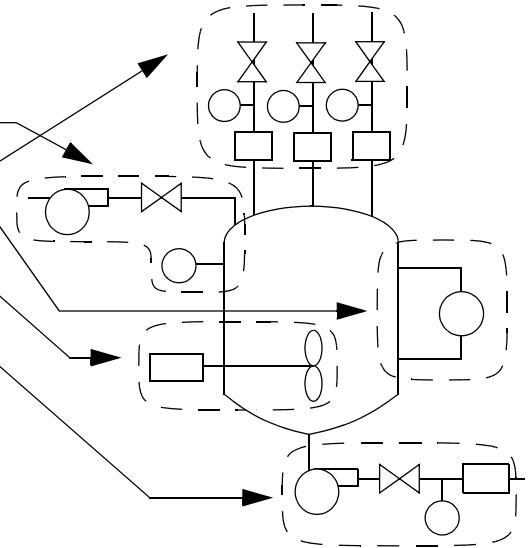
To cook the ingredients, the tank completes these steps.

1. Adds water.
2. Heats the water.
3. Adds other ingredients.
4. Mixes all ingredients.
5. Dispenses the finished product.

Which become these phases

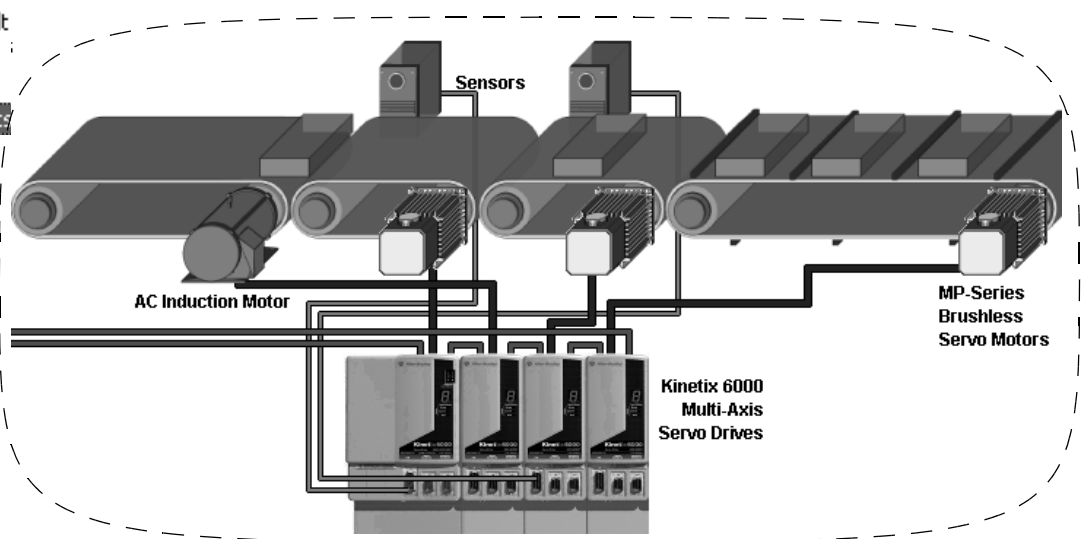
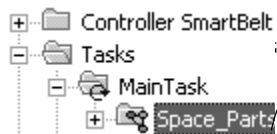


Which commands this equipment



Example 2: Smart Belt

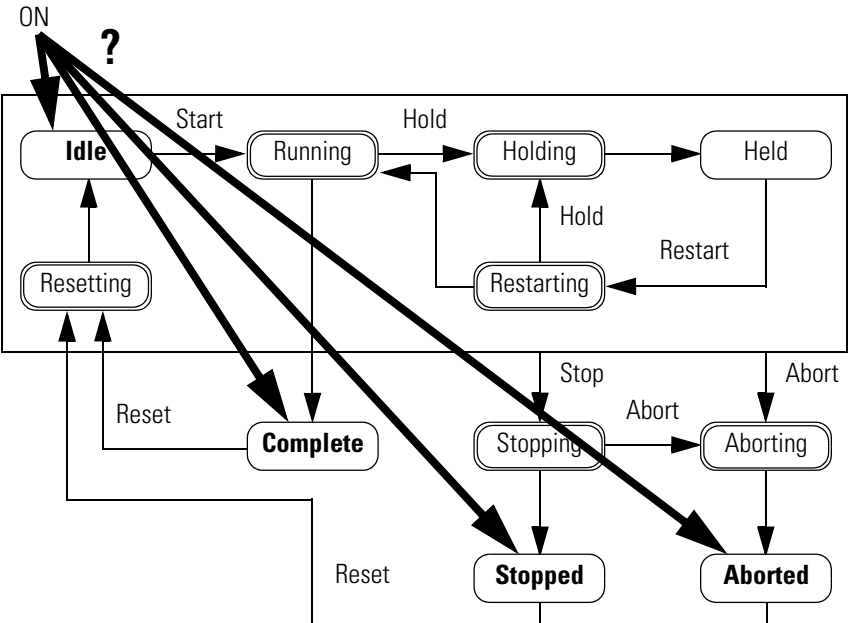
This example shows a smart belt. The smart belt does only one activity. It spaces product evenly on an exit belt. Because it does only one activity, it needs only one equipment phase.



State Model Guidelines

A state model divides the operating cycle of your equipment into a series of states. Each state is an instant in the operation of the equipment. It's the actions or conditions of the equipment at a given time.

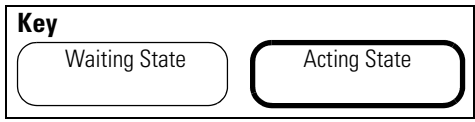
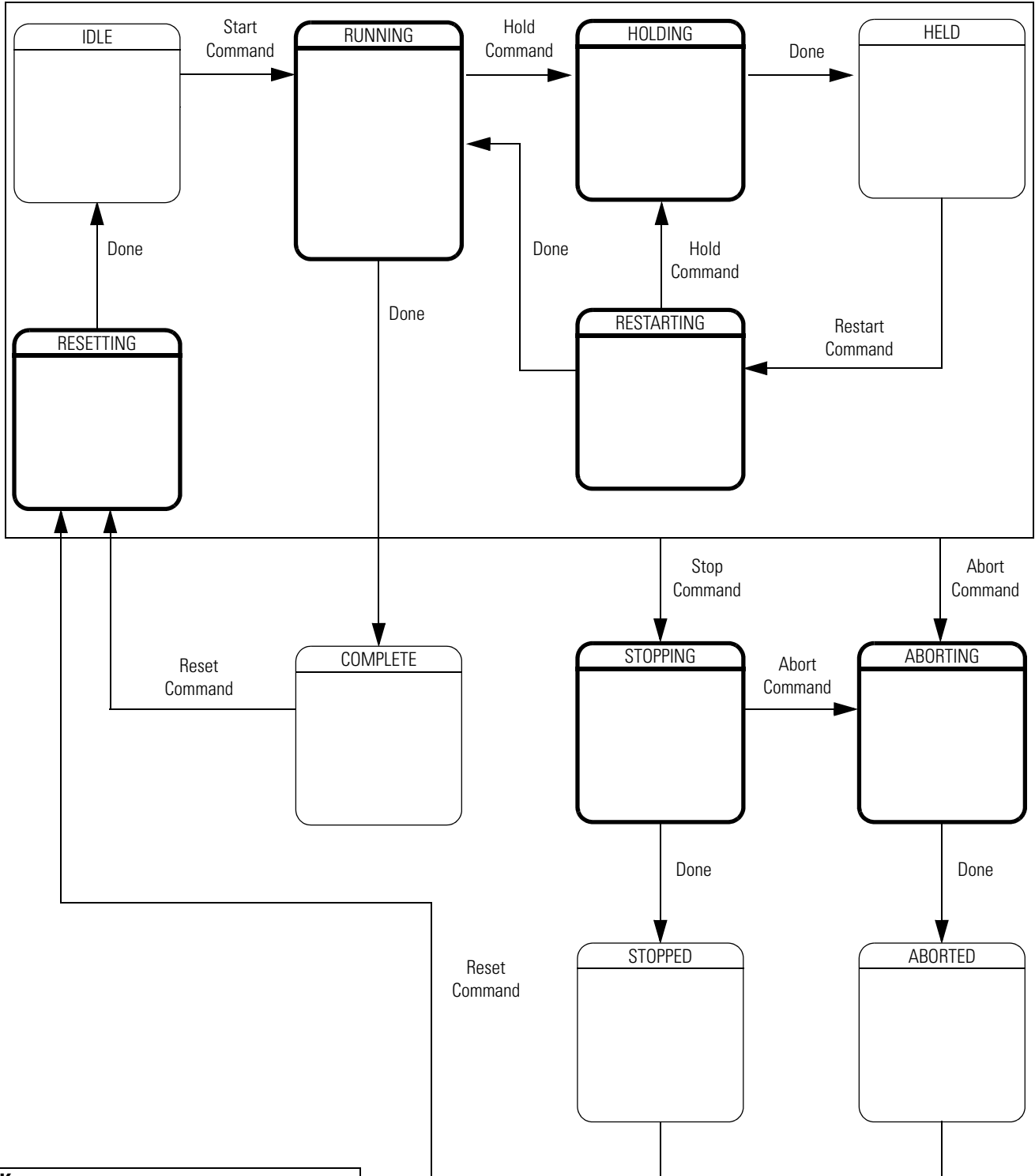
Follow these guidelines as you fill out the state model for an equipment phase.

Guideline	Details
Fill out one state model for each phase.	Each phase runs its own set of states. Fill out one state model worksheet for each phase.
Decide which state you want as your initial state after powerup.	<p>Which state do you want the equipment phase to go to when you turn on power?</p>  <pre> graph TD ON((ON ?)) --> Idle Idle -- Start --> Running Running -- Hold --> Holding Holding -- Hold --> Held Held -- Restart --> Restarting Restarting -- Hold --> Holding Running -- Stop --> Stopping Stopping -- Abort --> Aborting Aborting -- Abort --> Aborted Stopping --> Stopped Aborted --> Aborted Complete -- Reset --> Resetting Stopped -- Reset --> Resetting Aborted -- Reset --> Resetting Resetting --> Idle </pre> <p>An equipment phase goes to its initial state when you turn on power. We recommend that you use one of these states as the initial state:</p> <ul style="list-style-type: none"> • Idle (default) • Complete • Stopped <p>Choose the initial state that shows what your equipment is waiting to do after powerup (reset, run, and so forth).</p>

Guideline	Details	
<p>Start with the initial state and work through the model.</p>	<p>Start with the initial state. Then work forward from that point. Use the following questions to help you.</p>	
	For this State	Ask
	Stopped	What happens when you turn on power?
	Resetting	How does the equipment get ready to run?
	Idle	How do you tell that the equipment is ready to run?
	Running	What does the equipment do to make product?
	Holding	How does the equipment pause without making scrap?
	Held	How do you tell if the equipment is safely paused?
	Restarting	How does the equipment resume production after a pause?
	Complete	How do you tell when the equipment is done with what it had to do?
	Stopping	What happens during a normal shutdown?
	Aborting	How does the equipment shutdown if a fault or failure happens?
Aborted	How do you tell if the equipment is safely shut down?	
<p>Use only the states that you want.</p>	<p>Define only the states that are appropriate for your equipment. You are not required to use all states. The equipment phase just skips any states that you don't add.</p>	
<p>For the producing and standby states, use subroutines.</p>	<p>If you want to define producing and standby states for your equipment, use subroutines.</p> <ul style="list-style-type: none"> A. Create a routine for the producing state and another routine for the standby state. B. In the running state, check for the produce ersus standby conditions. Set either the Producing bit or the Standby bit of the equipment phase tag. C. To call the corresponding routine, use the Producing and Standby bits as conditions. <p>See Appendix A.</p>	

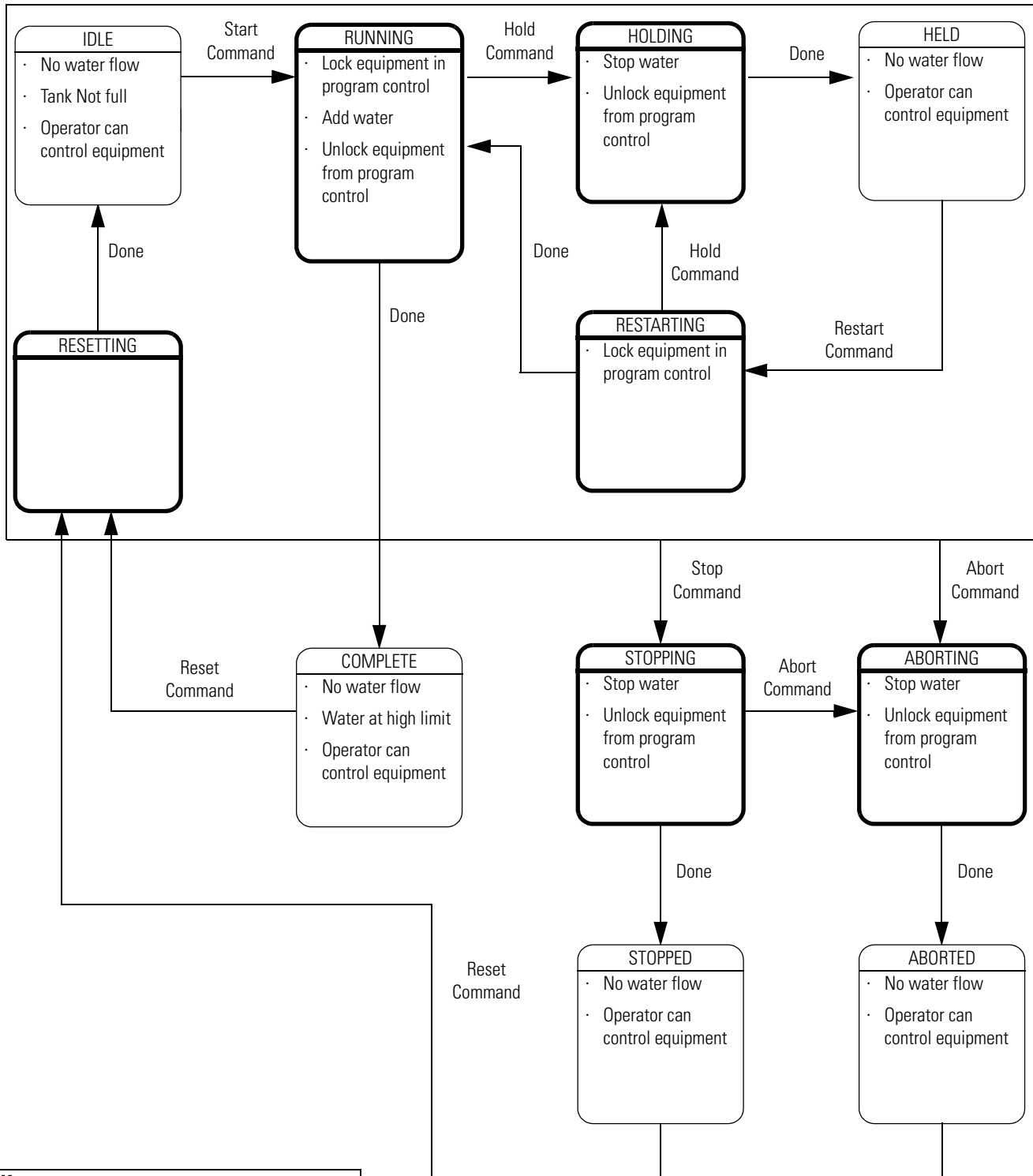
State Model Worksheet

Equipment Phase:



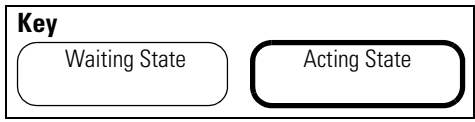
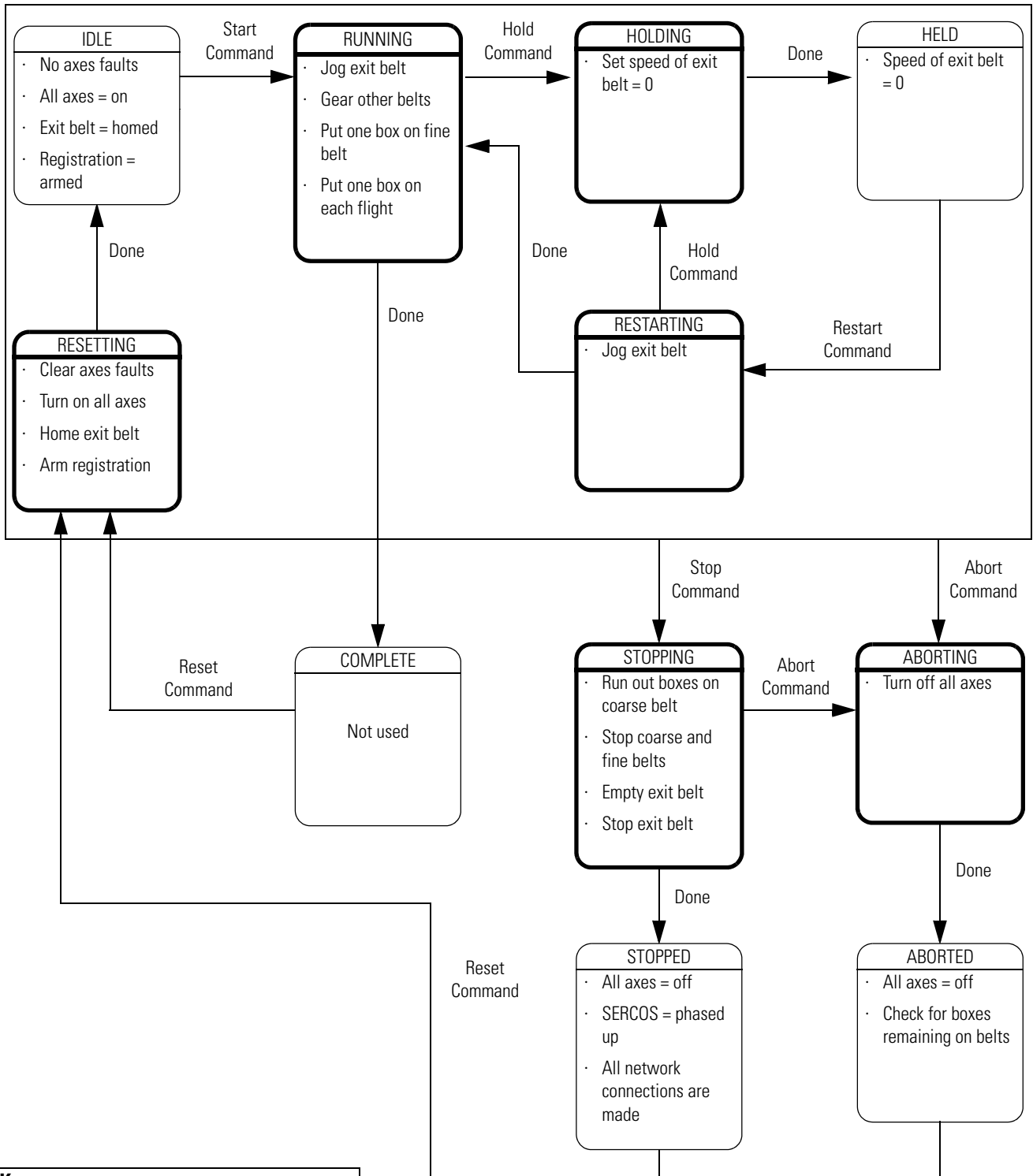
Example 1: Add Water

Equipment Phase: *Add Water*



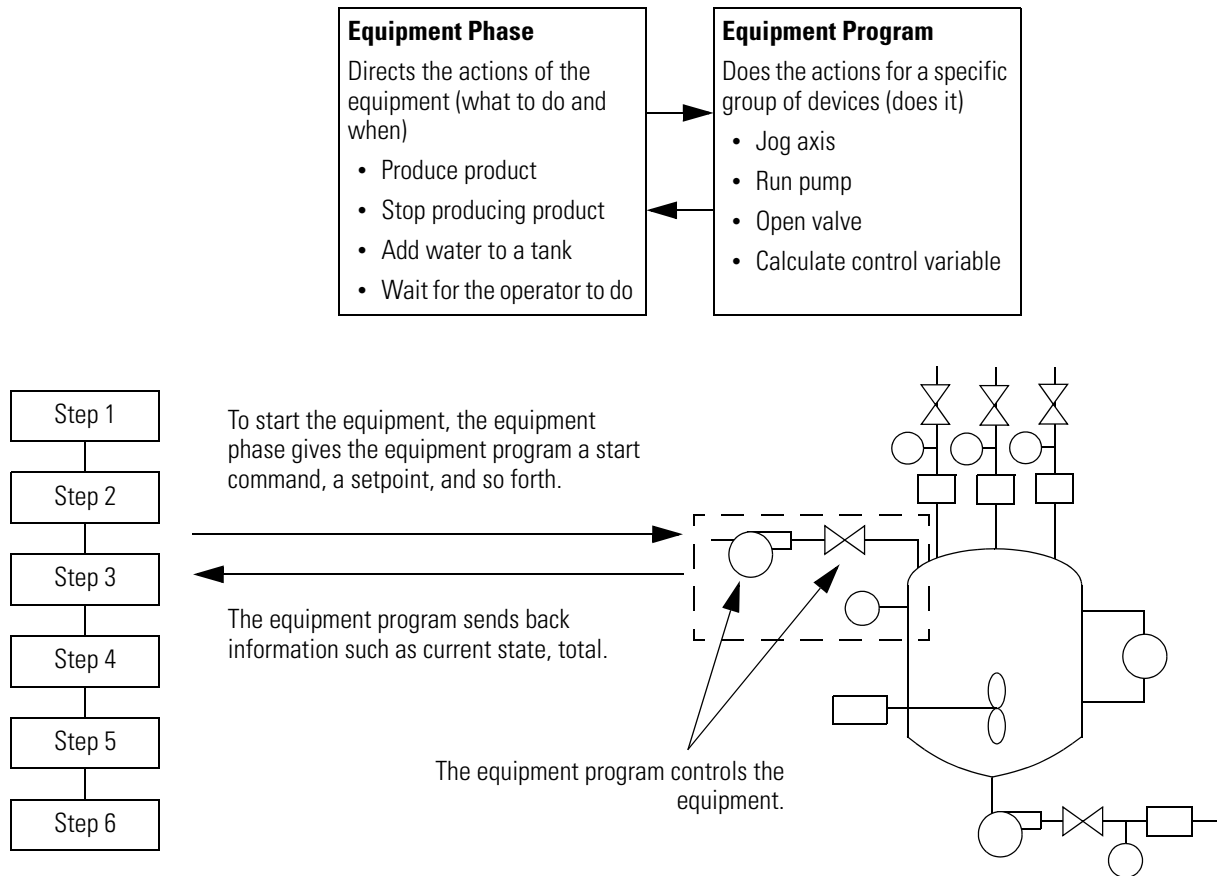
Example 2: Space Parts

Equipment Phase: *Space Parts*

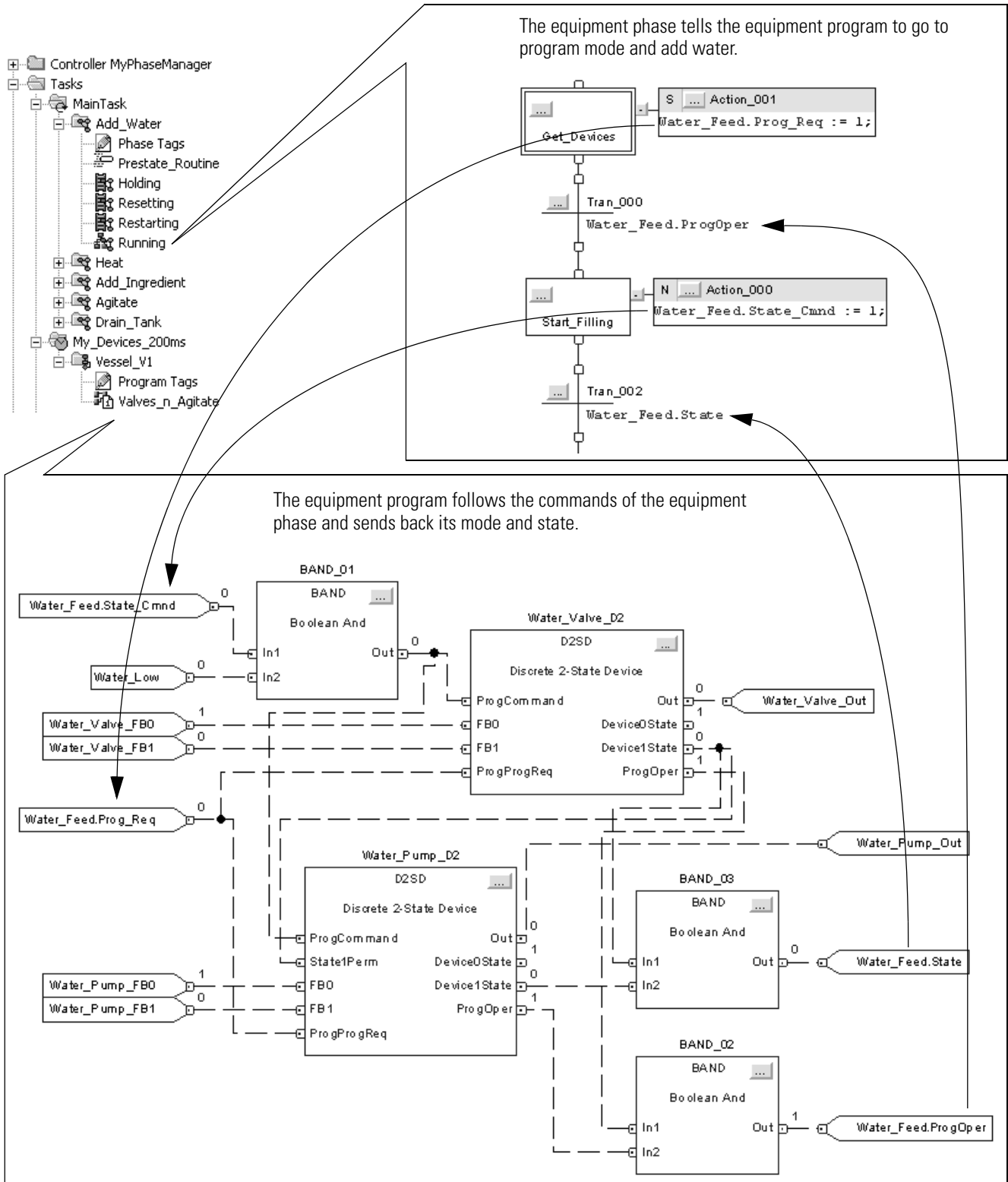


Equipment Code Guidelines

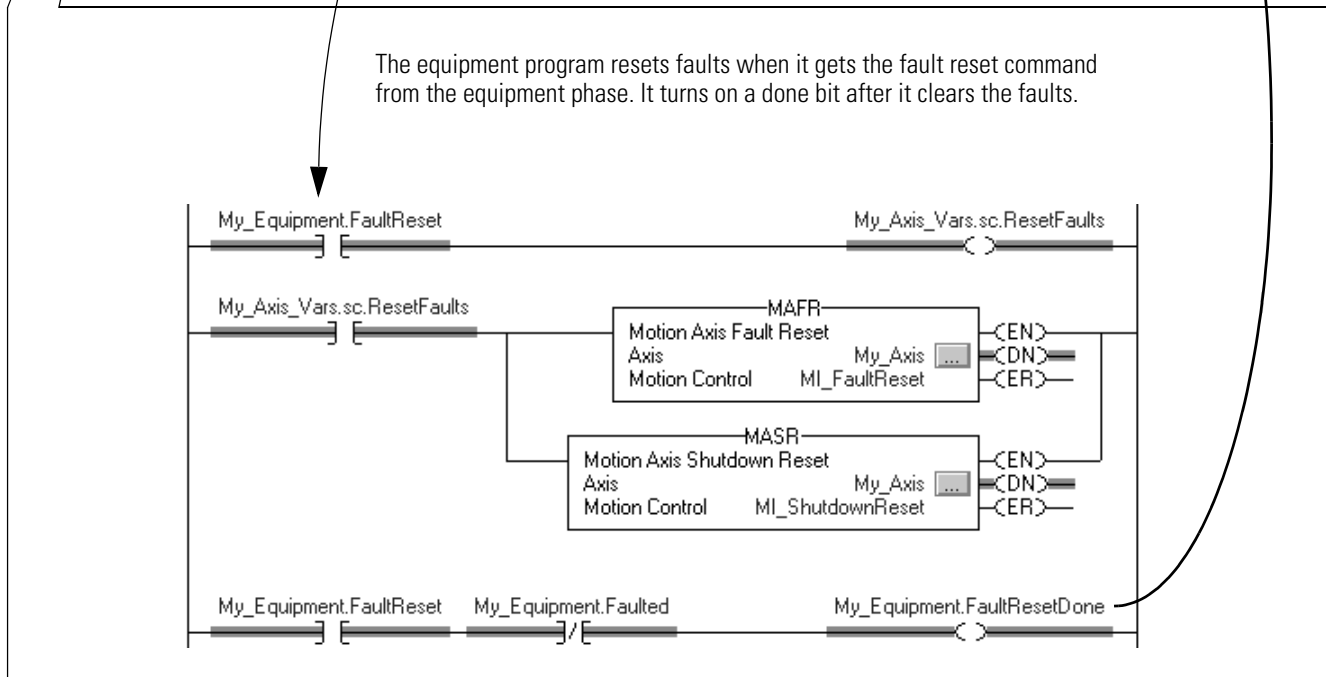
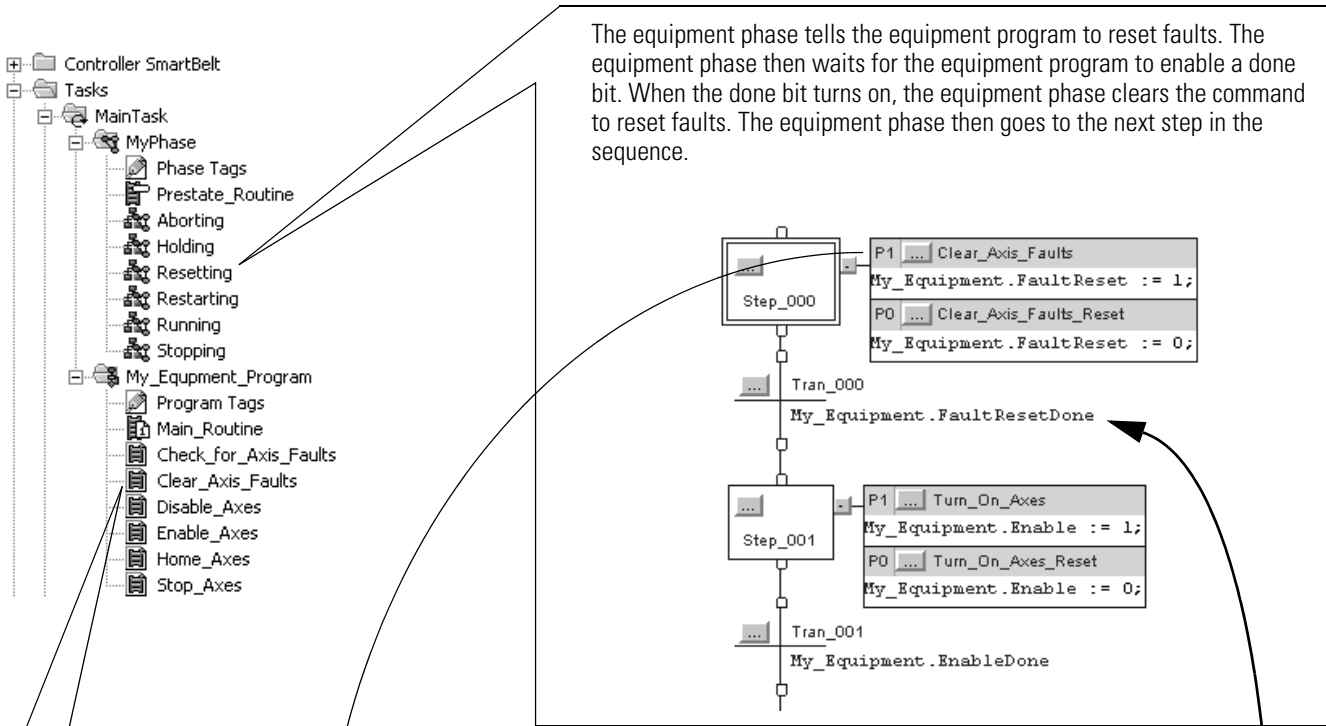
An equipment phase lets you separate the procedures (recipes) for how to make the product from the control of the equipment that makes the product. This advantage makes it much easier to execute different procedures for different products using the same equipment.



Example 1: Add Water to a Tank



Example 2: Smart Belt

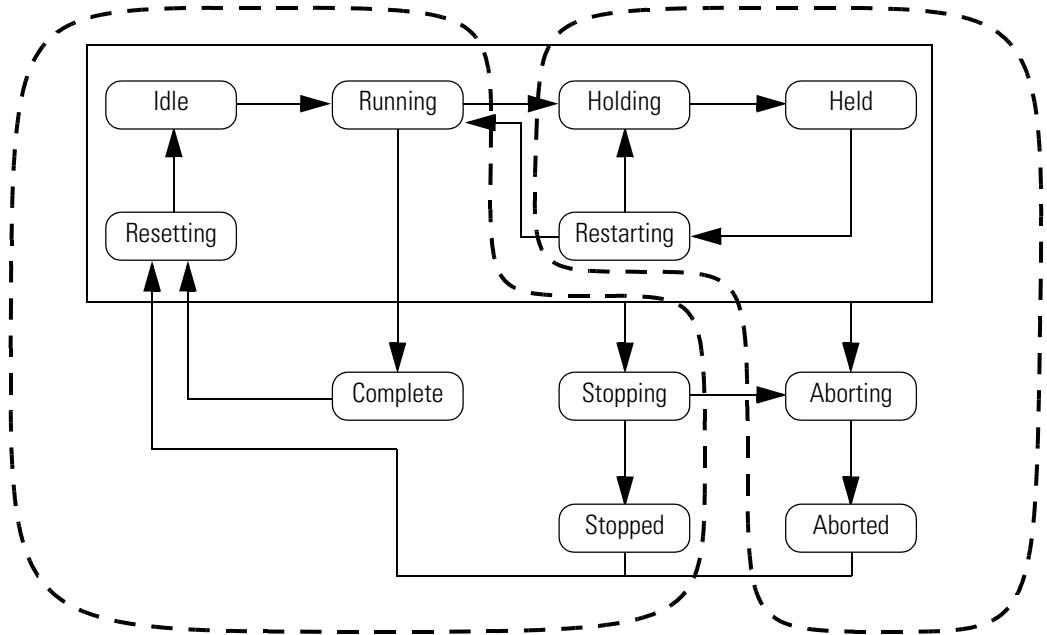


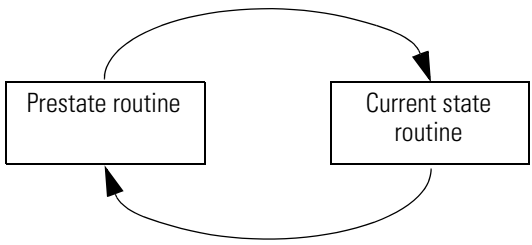
Execution Guidelines

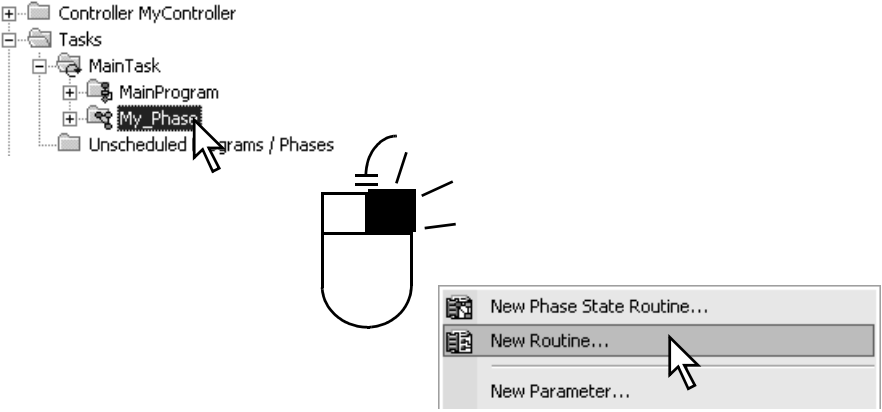
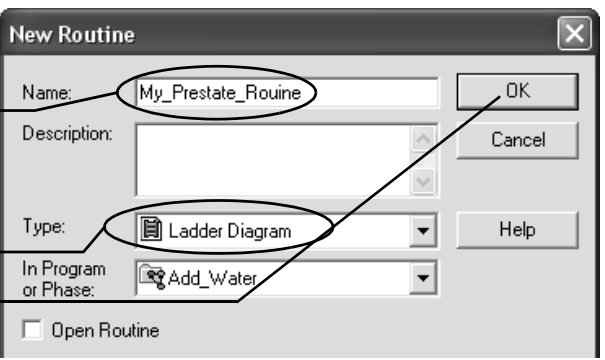
A state model makes it much easier to separate the normal execution of your equipment from any exceptions (faults, failures, off-normal conditions).

Use the resetting, running, and stopping states for the normal execution of the equipment.

Use the holding, restarting, and aborting states to handle exceptions (faults, failures, off-normal conditions).



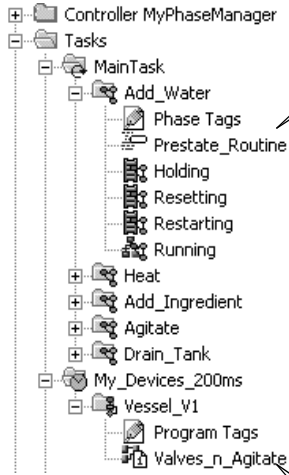
Guideline	Details
Use the prestate routine to watch for faults.	<div style="text-align: center;">  </div> <p>Use the prestate routine for conditions that you want to watch all time such as fault bits. The prestate routine:</p> <ul style="list-style-type: none"> • Runs constantly. • Runs before each scan of a state. • Runs even in the waiting states (idle, held, complete, stopped, or aborted).

Guideline	Details
<p>Create a prestate routine just like the routine for a program. It's not a phase state routine.</p> <p>1.</p>	 <p>The screenshot shows a project tree on the left with the following structure:</p> <ul style="list-style-type: none"> Controller MyController <ul style="list-style-type: none"> Tasks <ul style="list-style-type: none"> MainTask <ul style="list-style-type: none"> MainProgram <ul style="list-style-type: none"> My_Phase (highlighted) Unscheduled Programs / Phases <p>A context menu is open over 'My_Phase' with the following options:</p> <ul style="list-style-type: none"> New Phase State Routine... New Routine... (highlighted) New Parameter...
<p>2.</p>	 <p>The 'New Routine' dialog box is shown with the following fields:</p> <ul style="list-style-type: none"> Name: My_Prestate_Routine (circled) Description: (empty) Type: Ladder Diagram (circled) In Program or Phase: Add_Water (circled) <input type="checkbox"/> Open Routine <p>Buttons: OK, Cancel, Help.</p>
<p>3. Choose any language.</p> <p>4.</p>	<p>Lines 3 and 4 are connected to the 'Type' and 'In Program or Phase' fields in the 'New Routine' dialog box, respectively, indicating that any language can be chosen.</p>

Guideline	Details
<p>Assign a prestate routine.</p>	<p>1. Controller MyController Tasks MainTask MainProgram My_Phase Unscheduled Programs / Phases</p> <p>2. Finalize All Edits in Phase Ctrl+Shift+F Properties</p> <p>3. Equipment Phase Properties - Add_Ingredient General Configuration Parameters Monitor Assigned Routines: Prestate: My_Prestate_Routine Fault: <none> <input type="checkbox"/> Inhibit Equipment Phase Initial State: Idle <input checked="" type="checkbox"/> Complete State Immediately If not Implemented Initial Step Index: 0 External Sequencer Loss of Communication Command: <none> External Request Hold Action: <none></p> <p>4. OK Cancel Apply Help</p>
<p>Use a state bit to limit code to a specific state.</p>	<p>RSLogix 5000® software automatically makes a tag for each equipment phase. The tag has bits that tell you the state of the equipment phase.</p> <ul style="list-style-type: none"> • The tag is at the controller scope. • The tag uses the PHASE data type. • Use bits of the tag for code that you want to limit to certain states. <p>Example</p> <p>Suppose that the name of your equipment phase is <i>My_Phase</i>. And you have some code that you want to run only when the equipment phase is in the running state. In that case, check the <i>My_Phase.Running</i> bit for on (1):</p> <p>If <i>My_Phase.Running</i> then...</p> <p>See Appendix A for more information.</p>

Guideline	Details
Use the empty phase state routine to complete phase execution.	<p>Unlike normal program routines, phase state routines are called by the batch manager (not other program routines), so they always have the potential of being called.</p> <p>In the configuration for a phase state routine, if the Complete State Immediately if not Implemented option is checked in RSLogix 5000 programming software, version 18 or later, an implemented, but empty (no logic), phase state routine behaves the same as an implemented 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 programming software, version 16 and earlier, if a phase enters a state for which a state routine exists, but contains no logic, execution of the phase stops. The routine does complete, but there is no logic to execute.</p> <p>Choose from the following if you import a new state routine and in the Online Options dialog box.</p> <ul style="list-style-type: none"> • Import Logic Edits as Pending, an empty routine is created in the controller and the pending edits exist in the offline project. • 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. • Finalize All Edits in Program, the routine is created with the new logic and does not appear empty. <p>In the first two cases, if the Complete State Immediately if not Implemented option is checked, the empty routine completes immediately and allows phase execution to continue.</p>
Use the PFL instruction to signal a fault.	<p>The Equipment Phase Failure (PFL) instruction sets a failure code for an equipment phase. Use the code to signal a specific failure such as the fault of a specific device.</p> <ul style="list-style-type: none"> • The PFL instruction writes a code to the failure member for the equipment phase. • To see the failure code of an equipment phase, review the <i>phase_name.Failure</i> tag. • The failure code stays until any of the following happens: <ul style="list-style-type: none"> • A PFL instruction sets the failure code to a larger number. • The equipment phase transitions from the resetting state ? idle state. • A PCLF instruction clears the failure code. • FactoryTalk® Batch software clears the failure code. <p>See publication 1756-RM006 for more information.</p>
Use a PCLF instruction to clear a failure code.	<p>The Equipment Phase Clear Failure (PCLF) instruction clears the failure code for an equipment phase.</p> <ul style="list-style-type: none"> • A CLR instruction, MOV instruction, or assignment (:=) d doesn't change the failure code of an equipment phase. • If you are testing a PCLF instruction, make sure RSLogix 5000 software doesn't own the equipment phase. The PCLF instruction doesn't work if RSLogix 5000 software owns the equipment phase. <p>See publication 1756-RM006 for more information.</p>

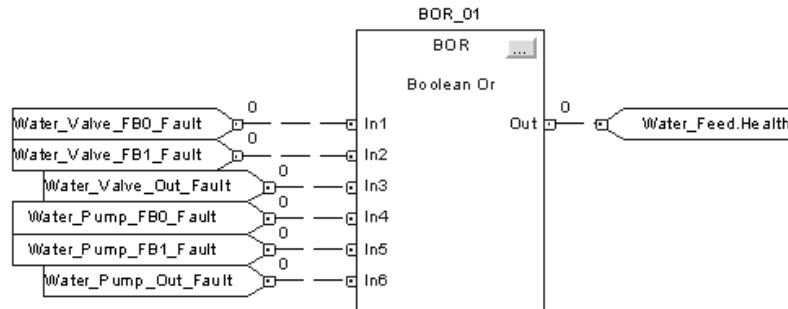
Example 1: Add Water to a Tank



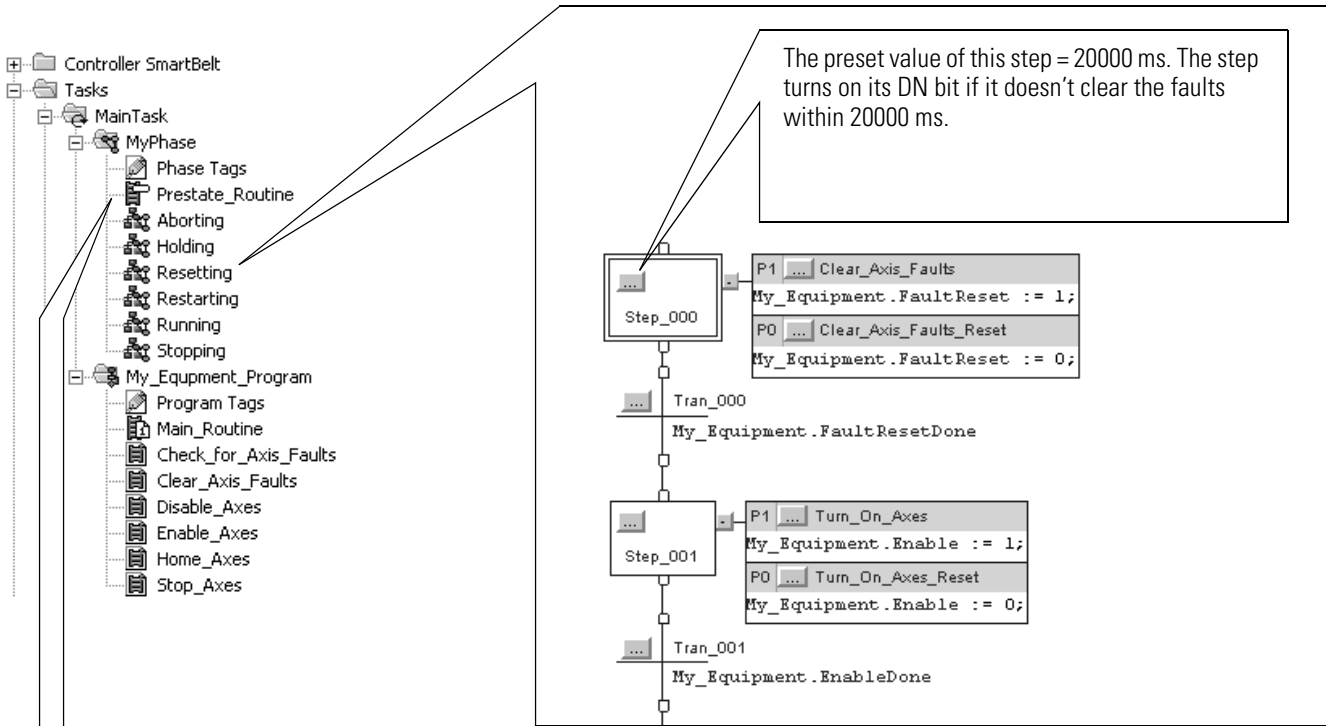
The prestate routine watches for equipment faults while the equipment phase is in the running state (*Add_Water.Running* = 1). If *Water_Feed.Health* = 1, then a fault happened. If a fault happens, the equipment phase sets a failure code of 202.

```
If Add_Water.Running And Water_Feed.Health Then
    PFL(202);
End_If;
```

The equipment program watches the fault bits of the valve, pump, and their feedback devices. If any of that equipment faults, the equipment program turns on the *Water_Feed.Health* bit.



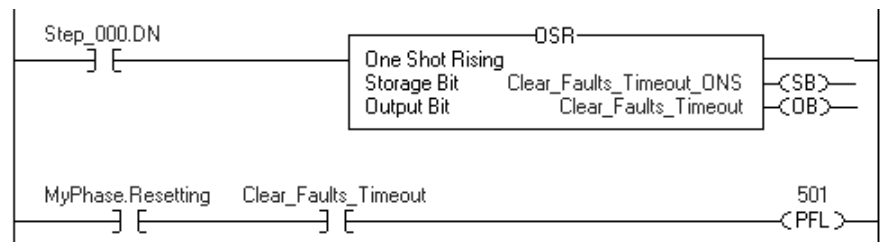
Example 2: Smart Belt



The preset value of this step = 20000 ms. The step turns on its DN bit if it doesn't clear the faults within 20000 ms.

If Step_000.DN = on, a timeout happened. When a timeout happens, the OSR instruction turns on the Clear_Faults_Timeout bit for one scan.

If MyPhase is in the resetting state and Clear_Faults_Timeout is on, then the PFL instruction signals a failure. The PFL instruction sets the failure code = 501.

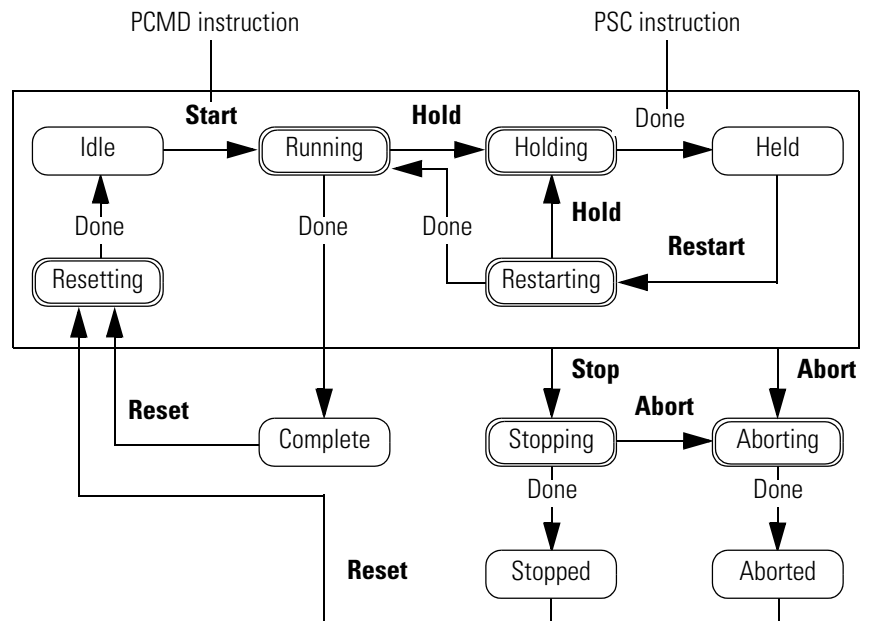


Transition Guidelines

To start an acting state, you usually have to give the equipment phase a command. The command tells the equipment phase and its equipment to start doing something or do something different. Use the Equipment Phase Command (PCMD) instruction to give a command to an equipment phase.

Optional: You can also use FactoryTalk Batch software in place of a PCMD instruction to trigger transitions

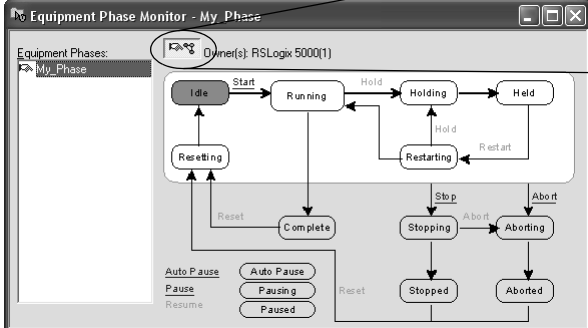
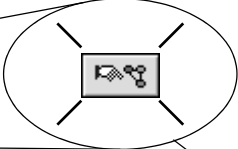
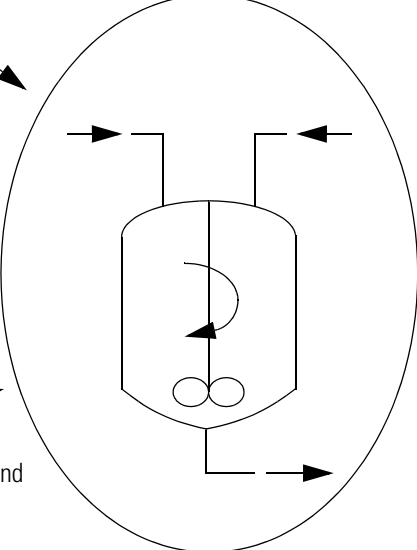
Use the state model to see which transitions need a PCMD instruction.



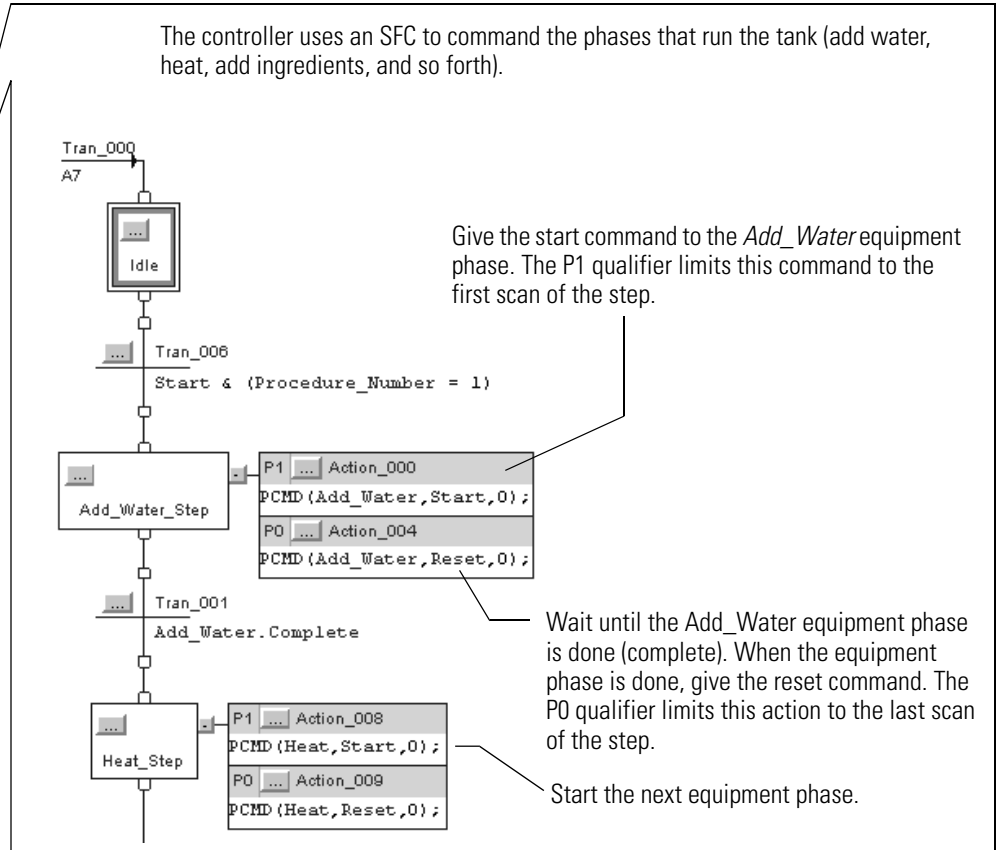
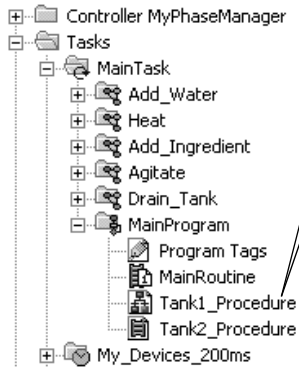
Type of Transition	Description	Instruction						
Command	<p>A command tells the equipment to start doing something or do something different. For example, the operator pushes the start button to start production and the stop button to shut down.</p> <p>PhaseManager software uses these commands:</p> <table border="1" data-bbox="402 516 1157 743"> <tr> <td data-bbox="402 516 651 554">Reset</td> <td data-bbox="659 516 907 554">Stop</td> <td data-bbox="915 516 1157 554">Restart</td> </tr> <tr> <td data-bbox="402 564 651 602">Start</td> <td data-bbox="659 564 907 602">Hold</td> <td data-bbox="915 564 1157 602">Abort</td> </tr> </table>	Reset	Stop	Restart	Start	Hold	Abort	<p>PCMD</p> <p>Use an Equipment Phase Command (PCMD) instruction to give a command. Or use RSLogix 5000 software.</p> <p>See the Logix5000 Controllers Advanced Process Control and Drives Instructions Reference Manual, publication 1756-RM006 for more information.</p>
Reset	Stop	Restart						
Start	Hold	Abort						
Done	<p>Equipment goes to a waiting state when it's done with what it's doing. You configure your code to signal when the equipment is done. The waiting state shows that the equipment is done.</p> <p>Exception: The restarting state goes to the running state when it's done.</p>	<p>PSC</p> <p>Use the Phase State Complete (PSC) instruction to signal when a state is done. See the Logix5000 Controllers Advanced Process Control and Drives Instructions Reference Manual, publication 1756-RM006 for more information.</p>						

Guideline	Details
A PCMD instruction causes a transition right away.	A PCMD instruction makes an equipment phase go to the commanded state. The equipment phase changes states as soon as it finishes its current scan. This state change happens even if the current state isn't done.

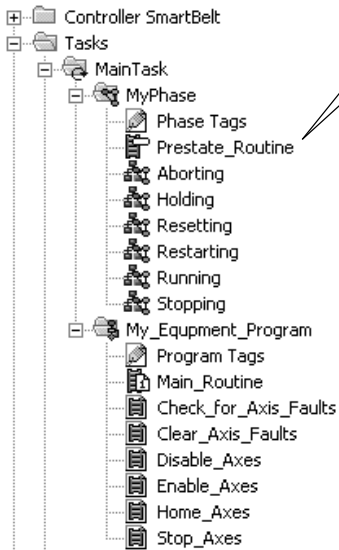
Guideline	Details								
<p>See if you must reset the state that you've left.</p>	<p>Are you leaving an acting state (for example, running, holding)?</p> <ul style="list-style-type: none"> • YES — Consider resetting the code of the state that you've left. • NO — You are probably not required to reset anything. <p>The equipment phase stops running the code of the current state when it goes to another state. Outputs remain at their last values unless the new state takes control of them. The stop also leaves an SFC at the step it was at when the equipment phase changed states.</p> <p>Example 1: No reset required</p> <p>Suppose that your equipment phase is in the idle state. In that case, it isn't running any state code. So you are probably not required to reset any state when you go to another state like running or stopping.</p> <p>Example 2: No reset required</p> <p>Suppose that your equipment phase is in the running state and you go to the holding state. When you return to the running state, you probably want to pick up where you left off. In that case, you are probably not required to reset the code in the running state.</p> <p>Example 3: Reset required</p> <p>Suppose that your equipment phase is half way through the resetting state and you give the stop command. And suppose that you want to run the entire resetting sequence when you return to it. In that case, you probably must reset the code of the resetting state. If the resetting state uses an SFC, then use the SFR instruction to reset it to the first step.</p>								
<p>Use an SFR instruction to reset the SFC of a state routine.</p>	<p>An SFC Reset (SFR) instruction is one way to reset an SFC. In some cases, reset an SFC from several other state routines.</p> <table border="1" data-bbox="565 1129 1482 1444"> <thead> <tr> <th data-bbox="565 1129 873 1203">To reset the SFC of this state</th> <th data-bbox="873 1129 1482 1203">Place an SFR instruction in this state routine</th> </tr> </thead> <tbody> <tr> <td data-bbox="565 1203 873 1255">Running</td> <td data-bbox="873 1203 1482 1255">Resetting</td> </tr> <tr> <td data-bbox="565 1255 873 1297">Holding</td> <td data-bbox="873 1255 1482 1297">Holding—Let the SFC reset itself at the last step.</td> </tr> <tr> <td data-bbox="565 1297 873 1444">Restarting</td> <td data-bbox="873 1297 1482 1444"> Reset the restarting routine in both these routines: <ul style="list-style-type: none"> • Holding—In case you return to holding before you finish restarting. • Restarting—Let the SFC reset itself at the last step. </td> </tr> </tbody> </table>	To reset the SFC of this state	Place an SFR instruction in this state routine	Running	Resetting	Holding	Holding—Let the SFC reset itself at the last step.	Restarting	Reset the restarting routine in both these routines: <ul style="list-style-type: none"> • Holding—In case you return to holding before you finish restarting. • Restarting—Let the SFC reset itself at the last step.
To reset the SFC of this state	Place an SFR instruction in this state routine								
Running	Resetting								
Holding	Holding—Let the SFC reset itself at the last step.								
Restarting	Reset the restarting routine in both these routines: <ul style="list-style-type: none"> • Holding—In case you return to holding before you finish restarting. • Restarting—Let the SFC reset itself at the last step. 								
<p>Use the PCMD instruction to go to an allowed next state.</p>	<p>PhaseManager software makes sure that an equipment phase follows the state model. So the equipment phase goes only to certain states from the state that it is in right now.</p> <p>Example 1: A transition is allowed</p> <p>Suppose that your equipment phase is in the running state and you give it the hold command. In that case, the equipment phase goes to holding because that transition is allowed.</p> <p>Example 2: A transition isn't allowed</p> <p>Suppose that your equipment phase is in the running state and you give it the reset command. In that case, the equipment phase stays in the running state. To go to the resetting state, you first have to stop or abort the equipment phase.</p>								

Guideline	Details
<p>See if you must use a POVR instruction instead of a PCMD instruction.</p>	<div style="display: flex; flex-direction: column; align-items: center;">  <div style="display: flex; justify-content: space-around; width: 100%;"> <div style="text-align: center;">  <p>Takes ownership</p> </div> <div style="text-align: center;">  <p>Can't command</p> </div> </div> <div style="margin-top: 20px;"> <div style="border: 1px solid black; padding: 5px; display: inline-block;"> <p>My Code</p> <div style="border: 1px solid black; width: 100px; height: 100px; margin: 5px; display: flex; flex-direction: column; align-items: center; justify-content: center;"> <div style="width: 80%; height: 20px; margin-bottom: 5px;"></div> <div style="width: 80%; height: 20px; margin-bottom: 5px;"></div> <div style="width: 80%; height: 20px;"></div> </div> </div> <p style="margin-left: 100px;">✗</p> </div> </div> <p>A. Are you giving the hold, stop, or abort command?</p> <ul style="list-style-type: none"> · NO — Use the PCMD instruction. · YES — Go to step B. <p>B. Must the command work even if you have manual control of the equipment phase via RSLogix 5000 software?</p> <ul style="list-style-type: none"> · YES — Use the POVR instruction instead. See the Logix5000 Controllers Advanced Process Control and Drives Instructions Reference Manual, publication 1756-RM006. · NO — Go to step C. <p>C. Must the command work even if FactoryTalk Batch software or another program owns the equipment phase?</p> <ul style="list-style-type: none"> · YES — Use the POVR instruction instead. See the Logix5000 Controllers Advanced Process Control and Drives Instructions Reference Manual, publication 1756-RM006. · NO — Use the PCMD instruction.

Example 1: Tank



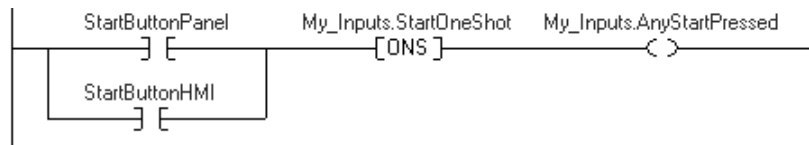
Example 2: Smart Belt



If the operator presses the start button on the machine or HMI, then

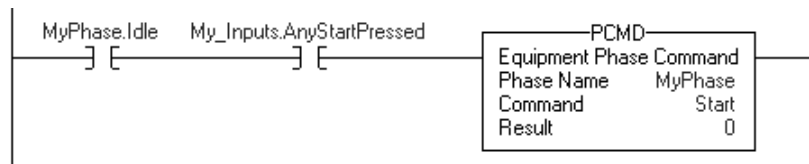
My_Inputs.AnyStartPressed = on for one scan.

The ONS instruction makes sure that *My_Inputs.AnyStartPressed* turns on only when a start button goes from off ? on.

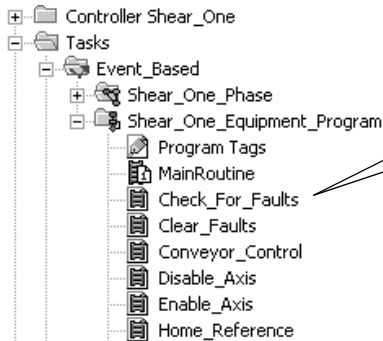


If the equipment phase is in the idle state and *My_Inputs.AnyStartPressed* = on, then

The PCMD instruction gives *MyPhase* the start command.



Example 3: Jam Detection

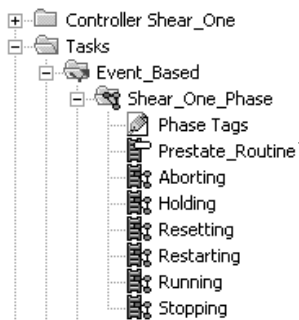


The equipment program watches for the following faults:

- Faulted axis
- Jammed material

If there is a fault, then

Local_Interface.Equipment_Faults_Cleared = 0. This tag is an alias for the controller-scoped tag *Shear_1*.



The prestate routine of the equipment phase watches for the equipment program to signal a fault.

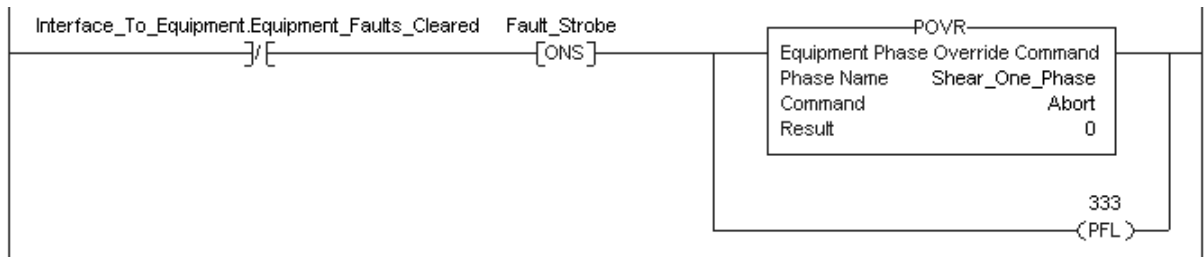
- If *Interface_To_Equipment.Equipment_Faults_Cleared* = 0 then there is a fault.
- Both *Interface_To_Equipment* and *Local_Interface* are aliases for *Shear_1*, so they have the same values.

If there is a fault Then

Give the *Shear_One_Phase* equipment phase the abort command. The POVR instruction makes sure that the command works even if someone has manual control of the equipment phase through RSLogix 5000 software.

The PFL instruction sets the failure code for *Shear_One_Phase* = 333.

The *Fault_Strobe* keeps these actions to one scan.



State Completion Guidelines

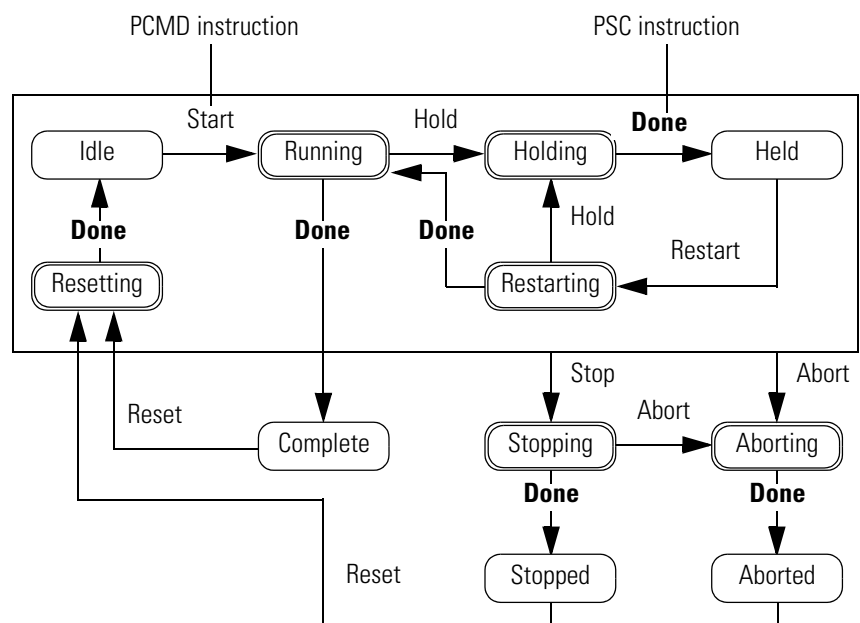
To leave an acting state, you usually signal that the state is done doing what it had to do. Use the Phase State Complete (PSC) instruction to signal when a state is done.

IMPORTANT

The PSC instruction *doesn't* stop the current scan of a routine.

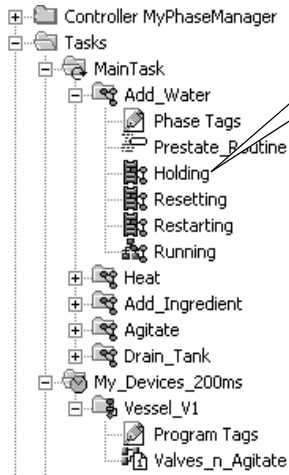
When the PSC instruction executes, the controller scans the rest of the routine and then transitions the equipment phase to the next state. The PSC instruction *does not* terminate the execution of the routine.

Use the state model to see which transitions need a PSC instruction.



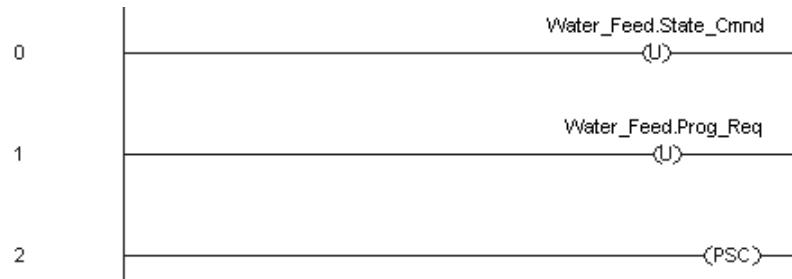
Type of Transition	Description	Instruction						
Command	<p>A command tells the equipment to start doing something or do something different. For example, the operator pushes the start button to start production and the stop button to shut down.</p> <p>PhaseManager software Software uses these commands:</p> <table border="1" data-bbox="402 443 911 533"> <tr> <td data-bbox="402 443 651 485">Reset</td> <td data-bbox="651 443 911 485">Stop</td> <td data-bbox="911 443 1162 485">Restart</td> </tr> <tr> <td data-bbox="402 485 651 533">Start</td> <td data-bbox="651 485 911 533">Hold</td> <td data-bbox="911 485 1162 533">Abort</td> </tr> </table>	Reset	Stop	Restart	Start	Hold	Abort	<p>PCMD</p> <p>Use an Equipment Phase Command (PCMD) instruction to give a command. Or use RSLogix 5000 software.</p>
Reset	Stop	Restart						
Start	Hold	Abort						
Done	<p>Equipment goes to a waiting state when it's done with what it's doing. You configure your code to signal when the equipment is done. The waiting state shows that the equipment is done.</p> <p>Exception: The restarting state goes to the running state when it's done.</p>	<p>PSC</p> <p>Use the Phase State Complete (PSC) instruction to signal when a state is done. See the Logix5000 Controllers Advanced Process Control and Drives Instructions Reference Manual, publication 1756-RM006 for more information.</p>						

Example 1: Add Water to a Tank

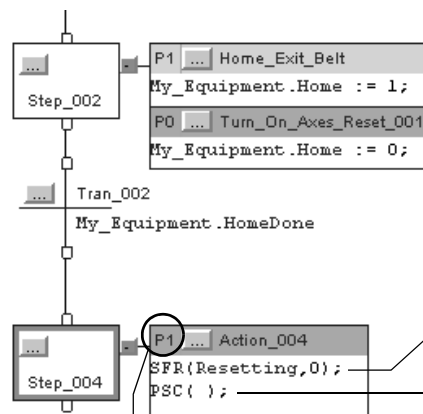
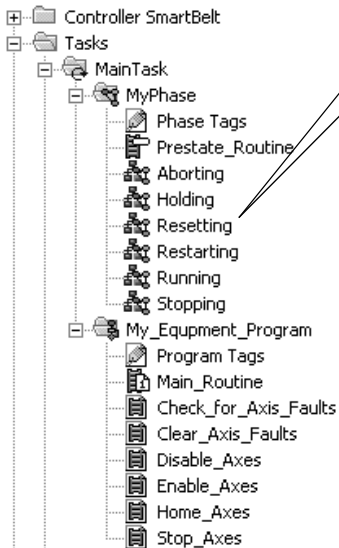


The holding state does three things.

1. Rung 0 — stop the water.
2. Rung 1 — unlock the devices from program control.
3. Rung 2 — signal that the state is done.



Example 2: Smart Belt



At the last step of the resetting state:

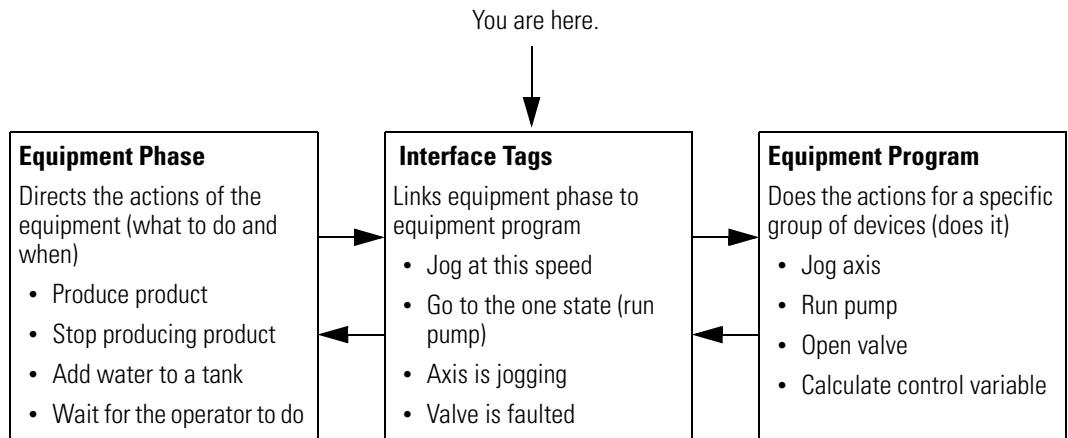
- The SFR instruction resets the SFC so it is ready for the next time you need it.
- The PSC instruction signals that the state is done.

Note: The P1 qualifier runs the actions only one time.

Equipment Interface Tag Guidelines

An equipment interface tag links an equipment phase to an equipment program.

- The equipment phase uses the tag to configure and command the equipment program.
- The equipment program uses the tag to report its status or condition.



Guideline	Details	
List the values that your equipment phase must give to the equipment program or get back from it.	Think of these values as a faceplate to the equipment program. It is the values that your equipment phase uses to control and monitor the equipment program. Exclude I/O data.	
	Inputs to the equipment program	Outputs from the equipment program
	<ul style="list-style-type: none"> • Mode requests • Setpoints • Commands such as on, off, start, stop, reset • Permissives • Overrides 	<ul style="list-style-type: none"> • Mode status • Control values • Done or completion • Alarms • Faults • Health indication • Totals or accumulated values
Create a user-defined data type.	<p>A user-defined data type lets you make a template for your data. It lets you group related data into one data type. You then use the data type to make tags with the same data lay-out.</p> <p>If you have multiple equipment phases, lay out the data type so that it's easy to use with multiple equipment phases. Consider the following:</p> <ul style="list-style-type: none"> • Include a range of data that makes the data type more versatile. • Use names that are as general as possible. <p>Example: The name <i>State_Cmnd</i> lets you use it for any equipment that runs in two states like on/off, running/not running, pumping/not pumping. It is easier to reuse than names such as <i>Open</i> or <i>Close</i>. Those names apply to valves but not pumps or motors.</p>	

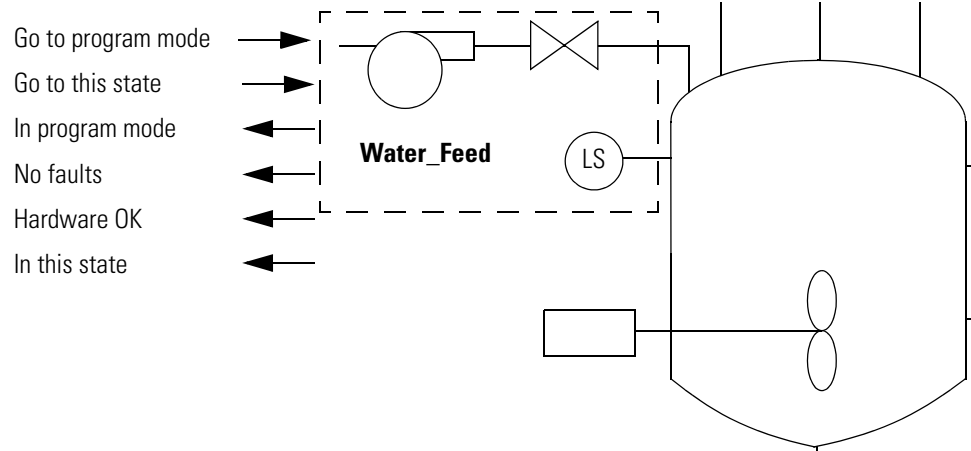
Guideline	Details
Create a tag for each equipment phase	<p>Create tag for the interface data of each equipment phase.</p> <ul style="list-style-type: none"> • Make a tag for each equipment phase. • Use the data type from guideline. • Make the tag at the controller scope. Both the equipment phase and the equipment program must get to the tag. • Consider using alias tags. See Alias Tag Guidelines on page 56.

Additional Resources

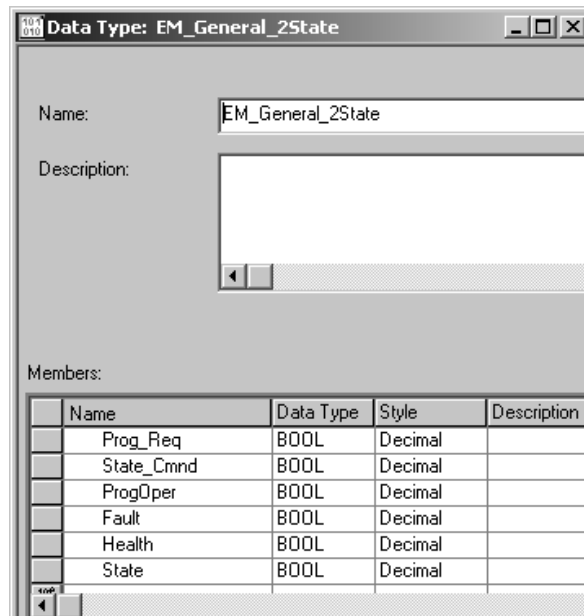
For this information	See this publication
Guidelines and considerations regarding: <ul style="list-style-type: none"> • User-defined data types • Alias tags 	Logix5000 Controllers Design Considerations, publication 1756-RM094
Step-by-step procedures on how to: <ul style="list-style-type: none"> • Create user-defined data types • Assign alias tags 	Logix5000 Controllers Common Procedures Programming Manual, publication 1756-PM001
Provides a description of each instruction in the Logix5000 format	Logix5000 Controllers Advanced Process Control and Drives Instructions Reference Manual, publication 1756-RM006

Example 1: Add Water to a Tank

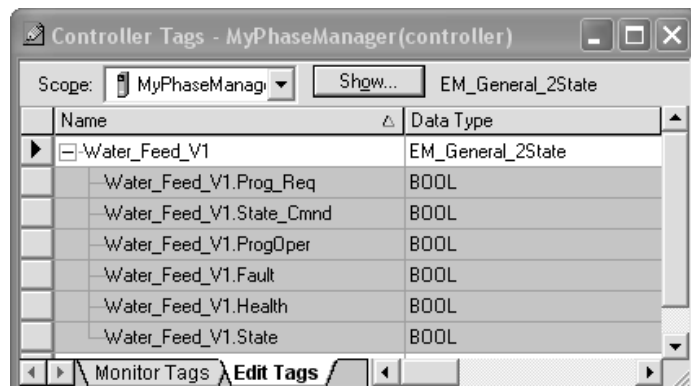
1. The equipment phase and equipment program share this data.



2. A user-defined data type creates a template for the data.



3. A tag stores the data that the equipment phase and equipment program share. The tag uses the user-defined data type from step 2.



Example 2: Smart Belt

The equipment phase and equipment program share this data.

Equipment program interface

Commands		Conditions or status	
Enable	Abort	FaultScroll	EnableCyclingDone
Disable	FaultReset	Faulted	DisableCyclingDone
Home	Stop	EnableDone	AbortingDone
ActivateRun	ArmRegistration	DisableDone	FaultResetDone
EnableProduct		HomeDone	StoppingDone
DisableProduct		ActivateRunDone	Selected
EnableCycling		EnableProductDone	RegistrationArmed
DisableCycling		DisableProductDone	

A separate user-defined data type holds data for each axis.

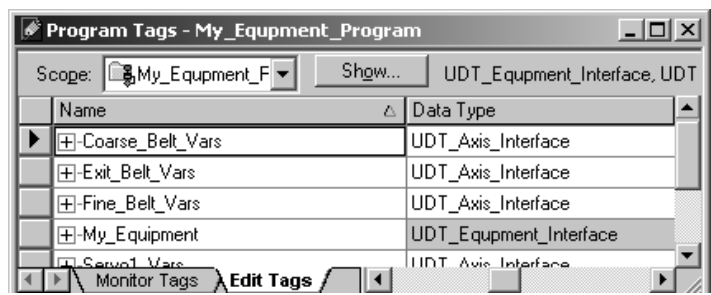
Axis interface

Commands		Conditions or status		
Enable	Abort	State	NoMotion	MoveActive
Disable	Stop	On	Homed	HomeDone
Home	ActivateRun	Ok	AxisSelected	RunDone
AutoRun		Auto	GearActive	
ResetFaults		Jogging	CamActive	

There is an interface tag for each axis and one for the entire machine.

One tag stores the data that the equipment phase and equipment program share. Other tags store the data for each individual axis.

Interface tag for each axis ←
 Interface tag for entire machine —

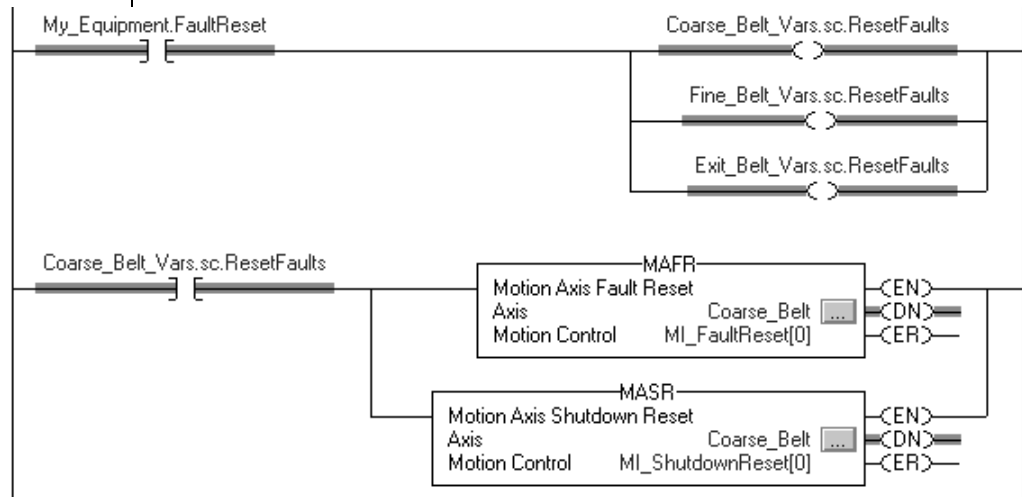


Example 2: Smart belt, Continued

The equipment program gets the command from the equipment phase and passes it to each axis.

Routine of the equipment program

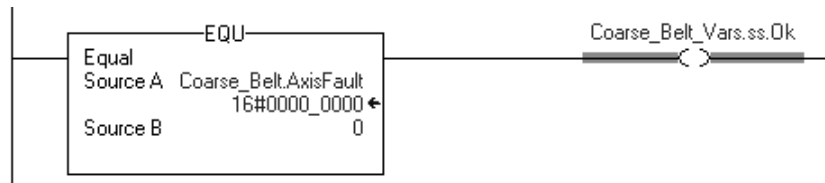
This tag	Is the interface between
My_Equipment	Equipment phase and equipment program
Coarse_Belt_Vars	Equipment program and an axis



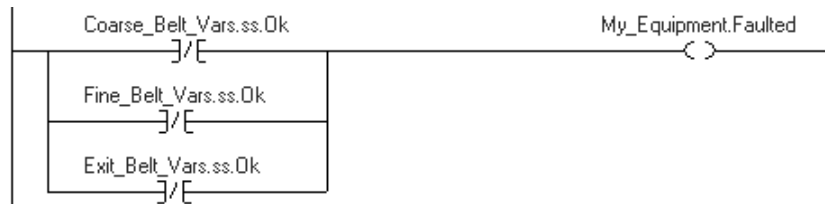
The equipment program collects the fault status of each axis and passes it back to the equipment phase.

Routine of the equipment program

The equipment program checks the fault code of each axis. If an axis isn't faulted, the OK bit for the axis turns on.



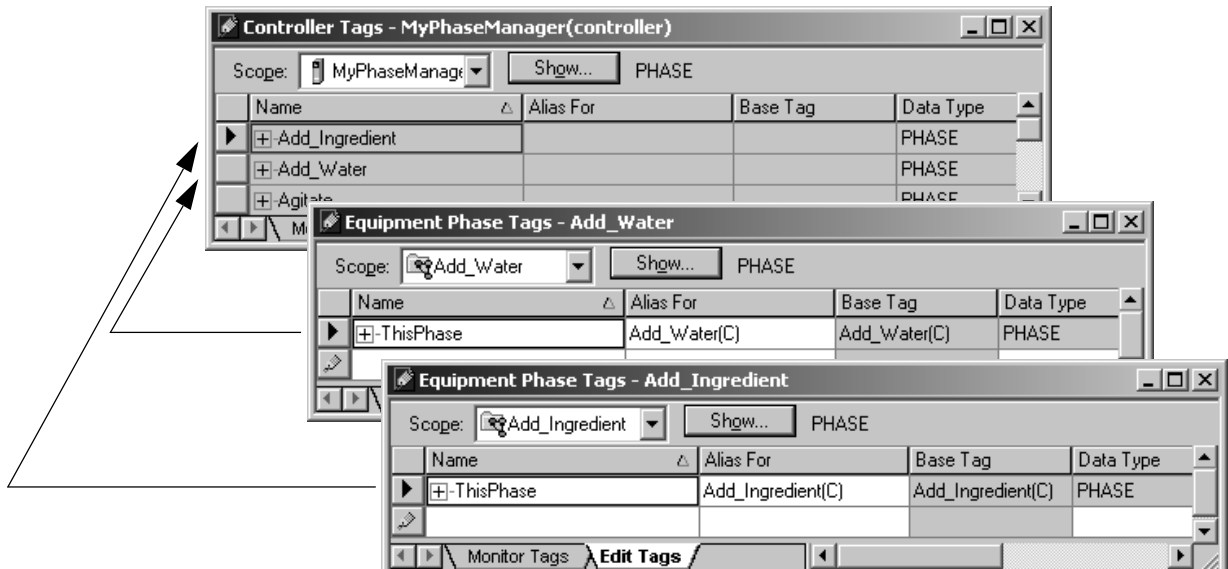
The equipment program collects the OK status of each axis. If the OK bit of each axis = on, then My_Equipment.Faulted = off (no faults).



Alias Tag Guidelines

Program-scoped tags and phase-scoped tags make your code easier to reuse. Make the tags aliases for tags at the controller scope. If you reuse the equipment phase (for example, copy/paste), simply point the phase-scoped tags to new tags at the controller scope. This practice reduces address fixes within the code.

Example



The controller automatically makes a tag for an equipment phase. The tag is at the controller scope (controller tag). Suppose that you plan to reuse an equipment phase for another part of your tank.

1. Make an alias tag for the first equipment phase. Make the tag at the phase scope and point it to the controller tag for that equipment phase.
2. Use the alias tag throughout the code of the equipment phase (This Phase).
3. Make a copy of the equipment phase.
4. Point the alias tag of the copy to its controller tag.

Additional Resources

For this information	See this publication
Guidelines and considerations for alias tags	Logix5000 Controllers Design Considerations, publication 1756-RM094
Steps to assign alias tags	Logix5000 Controllers Common Procedures Programming Manual, publication 1756-PM001

Notes:

PHASE Data Type

Introduction

The PHASE data type gives you status information about an equipment phase.

When you create an equipment phase, RSLogix 5000® software creates a tag for the status of the equipment phase.

Controller scope
Name = *phase_name*
PHASE data type



Set and Clear Equipment Phase Tag Values

For most of the members of the PHASE data type, you can only monitor its value. You can control only the following members.

Member	Control Method	
StepIndex	<p>If you program an equipment phase as a sequence of steps in ladder diagram or structured text, use the StepIndex value as the step number or bit value. (SFCs automatically sequence through steps.)</p> <ul style="list-style-type: none"> To initialize the StepIndex value, use the configuration properties for the equipment phase. <div style="text-align: right;"> </div> <ul style="list-style-type: none"> To advance to the next step, write logic to increment the StepIndex value (for example, MOV, MUL, OTL, :=) 	
Failure	To	Use this instruction
	Set the Failure value	Equipment Phase Failure (PFL)
	Clear the Failure value	Equipment Phase Clear Failure (PCLF)

Member	Control Method
NewInputParameters	To clear the NewInputParameters bit, use an Equipment Phase New Parameters (PRNP) instruction.
Producing	Use bit-level instructions or an assignment to set or clear this bit (for example, OTE, :=).
Standby	Use bit-level instructions or an assignment to set or clear this bit (for example, OTE, :=).

PHASE Data Type

If you want to	Then check this member	Data type	Notes	
			For this state:	Use this bit:
Use one member to monitor the state of an equipment phase	State	DINT	Read-only	
			Running	0
			Holding	1
			Restarting	2
			Stopping	3
			Aborting	4
			Resetting	5
			Idle	6
			Held	7
			Complete	8
			Stopped	9
Aborted	10			
See if the equipment phase is in the running state	Running	BOOL	Read-only	
See if the equipment phase is in the holding state	Holding	BOOL	Read-only	
See if the equipment phase is in the restarting state	Restarting	BOOL	Read-only	
See if the equipment phase is in the stopping state	Stopping	BOOL	Read-only	
See if the equipment phase is in the aborting state	Aborting	BOOL	Read-only	
See if the equipment phase is in the resetting state	Resetting	BOOL	Read-only	
See if the equipment phase is in the idle state	Idle	BOOL	Read-only	
See if the equipment phase is in the held state	Held	BOOL	Read-only	
See if the equipment phase is in the complete state	Complete	BOOL	Read-only	

If you want to	Then check this member	Data type	Notes	
See if the equipment phase is in the stopped state	Stopped	BOOL	Read-only	
See if the equipment phase is in the aborted state	Aborted	BOOL	Read-only	
Use one member to monitor the substate of an equipment phase	Substate	DINT	Read-only	Use this bit
			Pausing	0
			Paused	1
			AutoPause	2
See if the equipment phase is in the pausing substate	Pausing	BOOL	Read-only	
See if the equipment phase is in the paused substate	Paused	BOOL	Read-only	
See if the equipment phase is in the auto pause substate	AutoPause	BOOL	Read-only	
Use an integer value or the bits of an integer to sequence through a series of steps	StepIndex	DINT	<ul style="list-style-type: none"> To initialize the StepIndex value, use the configuration properties for the equipment phase. To advance to the next step, use logic such as an MOV, MUL, or := to increment the StepIndex value. 	
Flag a specific exception for an equipment phase (fault, failure, off-normal condition, and so forth.)	Failure	DINT	To	Use
			Set a Failure value	PFL instruction
			Clear the Failure value	PCLF instruction
Find the unit ID of an equipment phase	UnitID	DINT	FactoryTalk Batch software sets this value.	
Monitor the ownership of an equipment phase	Owner	DINT	Read-only	
See if an external request is in process via a PXRQ instruction	PendingRequest	DINT	<ul style="list-style-type: none"> Read-only Each bit = the state of a specific request, starting with bit 0. The bits are in the order shown by the request-specific members. 	
See if a Download Input Parameters request is in process via a PXRQ instruction	DownloadInputParameters	BOOL	Read-only	
See if a Download Input Parameters Subset request is in process via a PXRQ instruction	DownloadInputParameters Subset	BOOL	Read-only	
See if an Upload Output Parameters request is in process via a PXRQ instruction	UploadOutputParameters	BOOL	Read-only	
See if an Upload Output Parameters Subset request is in process via a PXRQ instruction	UploadOutputParameters Subset	BOOL	Read-only	
See if a Download Output Parameter Limits request is in process via a PXRQ instruction	DownloadOutputParameterLimits	BOOL	Read-only	
See if an Acquire Resources request is in process via a PXRQ instruction	AcquireResources	BOOL	Read-only	

If you want to	Then check this member	Data type	Notes
See if a Release Resources request is in process via a PXRQ instruction	ReleaseResources	BOOL	Read-only
See if a Send Message To Linked Phase request is in process via a PXRQ instruction	SendMessageToLinkedPhase	BOOL	Read-only
See if a Send Message To Linked Phase And Wait request is in process via a PXRQ instruction	SendMessageToLinkedPhaseAndWait	BOOL	Read-only
See if a Receive Message From Linked Phase request is in process via a PXRQ instruction	ReceiveMessageFromLinkedPhase	BOOL	Read-only
See if a Cancel Message To Linked Phase request is in process via a PXRQ instruction	CancelMessageToLinkedPhase	BOOL	Read-only
See if a Send Message To Operator request is in process via a PXRQ instruction	SendMessageToOperator	BOOL	Read-only
See if a Clear Message To Operator request is in process via a PXRQ instruction	ClearMessageToOperator	BOOL	Read-only
See if a Generate E Signature request is in process via a PXRQ instruction	GenerateESignature	BOOL	Read-only
See if a Download Batch Data request is in process via a PXRQ instruction	DownloadBatchData	BOOL	Read-only
See if a Download Material Track Data Container In Use request is in process via a PXRQ instruction	DownloadMaterialTrackDataContainerInUse	BOOL	Read-only
See if a Download Container Binding Priority request is in process via a PXRQ instruction	DownloadContainerBindingPriority	BOOL	Read-only
See if a Download Sufficient Material request is in process via a PXRQ instruction	DownloadSufficientMaterial	BOOL	Read-only
See if a Download Material Track Database Data request is in process via a PXRQ instruction	DownloadMaterialTrackDatabaseData	BOOL	Read-only
See if an Upload Material Track Data Container In Use request is in process via a PXRQ instruction	UploadMaterialTrackDataContainerInUse	BOOL	Read-only
See if an Upload Container Binding Priority request is in process via a PXRQ instruction	UploadContainerBindingPriority	BOOL	Read-only
See if an Upload Material Track Database Data request is in process via a PXRQ instruction	UploadMaterialTrackDatabaseData	BOOL	Read-only
See if your logic has aborted a PXRQ instruction	AbortingRequest	BOOL	Read-only
See if FactoryTalk Batch software has new parameters for an equipment phase	NewInputParameters	BOOL	<ul style="list-style-type: none"> • Read-only • FactoryTalk Batch software sets this bit when it has new parameters for an equipment phase. • To clear the NewInputParameters bit, use a PRNP instruction.

If you want to	Then check this member	Data type	Notes
Initiate a producing state	Producing	BOOL	Logix5000 equipment phases don't have a producing state. To create a producing state, use the Producing bit.
Initiate a standby state	Standby	BOOL	Logix5000 equipment phases don't have a standby state. To create a standby state, use the Standby bit.

Notes:

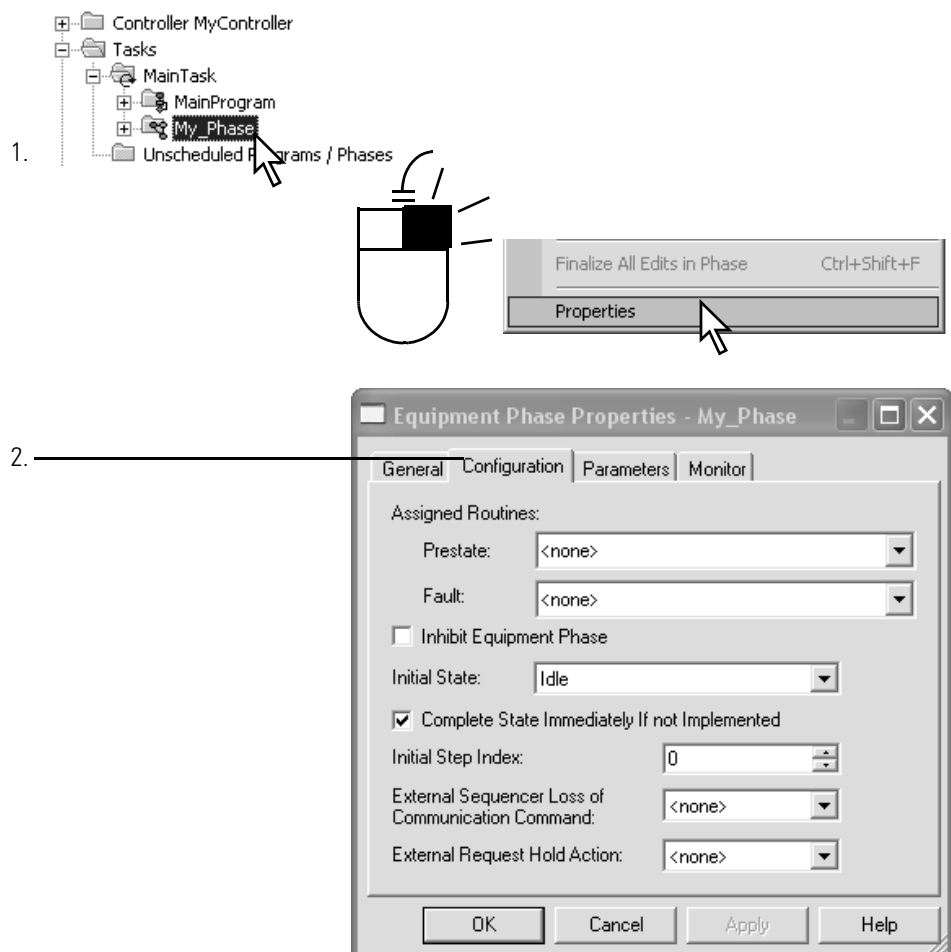
Configure an Equipment Phase

Introduction

This appendix steps you through the configuration settings for an equipment phase.

Use this appendix when you want to change the default settings of an equipment phase.

Open the Configuration for an Equipment Phase



Equipment Phase Settings

Use the following settings to configure an equipment phase.

Setting	Choices
<p>Prestate</p>	<div data-bbox="755 378 1279 619" data-label="Diagram"> </div> <p>The prestate routine runs constantly, even when the equipment phase is in the idle state. It runs before <i>each</i> scan of a state.</p> <p>Do you want to run a prestate routine?</p> <ul style="list-style-type: none"> • YES — Select the routine that you want to run. • NO — Leave this box set to none.
<p>Fault</p>	<p>A fault routine lets you clear a major fault made by an instruction.</p> <p>Do you want to create a fault routine for the instructions in this equipment phase?</p> <ul style="list-style-type: none"> • YES — Select the routine that you want as your fault routine. • NO — Leave this box set to none.
<p>Inhibit Equipment Phase</p>	<p>Do you want the controller to run this equipment phase?</p> <ul style="list-style-type: none"> • YES — Leave this box unchecked or uncheck it. • NO — Check this box.
<p>Initial State</p>	<p>Which state do you want the equipment phase to go to when you turn on the controller?</p> <ul style="list-style-type: none"> • Idle • Complete • Stopped • Aborted
<p>Complete State Immediately If not Implemented</p>	<p>Do you want the equipment phase to skip any states that you aren't using?</p> <ul style="list-style-type: none"> • YES — Leave this box checked or check it. • NO — Uncheck this box.
<p>Initial Step Index</p>	<p>A. Are any of the state routines in ladder diagram or structured text?</p> <ul style="list-style-type: none"> • NO — Skip this box. • YES — Go to step B. <p>B. Do any of those state routines use step numbers?</p> <ul style="list-style-type: none"> • YES — Type the number for the first step of each state. • NO — Skip this box. <p>The tag for the equipment phase has a StepIndex number. The controller resets the StepIndex each time the equipment phase changes states. The controller resets the StepIndex to the number you put in the Initial Step Index box.</p>

Setting	Choices
External Sequencer Loss of Communication Command	<p>A. Are you using RSBizWare™ Batch software to command this equipment phase?</p> <ul style="list-style-type: none"> • NO — Skip this box. • YES — Go to step B. <p>B. If the controller loses communication with RSBizWare Batch software, what do you want the equipment phase to do?</p> <ul style="list-style-type: none"> • Continue in its current state — Choose None. • Go to aborting — Choose Abort. • Go to holding — Choose Hold. • Go to stopping — Choose Stop. <p>The equipment phase must still follow the state model. For example, it goes to holding only if it is in running or restarting when communication fails.</p>
External Request Hold Action	<p>A. Are you using any PXRQ instructions?</p> <ul style="list-style-type: none"> • NO — Skip this box. • YES — Go to step B. <p>B. What do you want to do if an equipment phase goes to holding while a PXRQ instruction is in process?</p> <ul style="list-style-type: none"> • Nothing — Choose none. • Stop the request — Choose Clear.

Notes:

This manual uses the following terms.

Term	Definition	Example
Unit	A group of equipment that works together to produce the product or interim product. The equipment of a unit operates independent (relatively independent) from other equipment.	<ul style="list-style-type: none"> • brew kettle • mixing tank • bottle filling machine • bottle capping machine
Equipment module	<p>A group of input devices, output devices, motors, drives, and soft controls (PID loops, totalizers, and so forth.) that go together to perform a specific activity (task, function) of a unit. The devices within an equipment module:</p> <ul style="list-style-type: none"> • work as one entity. • operate independent (relatively independent) from other equipment. 	<ul style="list-style-type: none"> • fill a tank with water • mix the contents of a tank • drain a tank • fill bottles • cap bottles
Equipment module interface	Collection of data values that you supply to an equipment module or get from it to monitor and control it. An equipment module interface includes on/off/start/stop commands, mode requests, set points, and fault/health status. It acts as a faceplate for your logic to the equipment module.	<ul style="list-style-type: none"> • BOOL tag: Go_To_This_State • BOOL tag: In_This_State • BOOL tag: Go_To_This_Mode • BOOL tag: In_This_Mode
Unit procedure	<p>The sequence of processing activities that a unit performs to produce the product or interim product.</p> <ul style="list-style-type: none"> • A unit procedure directs the execution of phases. • A unit procedure could be a hierarchy of SFCs that is subdivided into specific operations. Each operation directs the execution of a group of phases. • A unit may have multiple unit procedures depending on how the sequence changes for different products. 	<p style="text-align: center;">Brew</p> <pre> graph TD Charge[Charge] --- Boil[Boil] Boil --- Drain[Drain] </pre>
Phase	A specific task that your equipment does. A phase directs the actions of your equipment. It tells the equipment what to do and when to do it.	<ul style="list-style-type: none"> • Fill bottles with product. • Put bottles in carton. • Fill tank with water • Mix ingredients in tank
State	<p>The condition of your equipment in relation to normal production. A phase can have up to 11 different states, some of which are active and other are waiting.</p> <ul style="list-style-type: none"> • Active (. . . ing) states represent the things your equipment does at a given time (running, holding, restarting, stopping, aborting, resetting). Each state contains a separate blocks of code (routine) and can call other routines. • Waiting states represent the condition of your equipment when it is in-between active states (stopped, complete, idle, held, aborted, stopped). Waiting states have <i>no</i> associated routines or logic <p>A phase transitions from one state to another only in a specific order. Your equipment can go from its current state to only certain other states.</p>	<ul style="list-style-type: none"> • In the running state, fill the tank with water. • In the holding state, temporarily stop filling the tank with water. • In the resetting state, reset the logic and clear the totals.

Notes:

A

- aborted state**
 - use 13
- aborting state**
 - use 13, 34
- add**
 - equipment phase 18
 - phase state routine 18

C

- clear**
 - PHASE tag values 57
- command**
 - example 45, 46
 - give 14
 - give with PCMD instruction 41
 - give with RSLogix 5000 software 19
- complete state**
 - use 13
- configure**
 - equipment phase 63
- create**
 - equipment phase 18
 - phase state routine 18

E

- equipment module**
 - See equipment program
- equipment phase**
 - add prestate routine 35, 36
 - compared to PackML 16
 - compared to S88 16
 - configure 63
 - create 18
 - create a phase state routine 18
 - data type 58
 - define your states 26
 - faults 34
 - handle faults 34
 - inhibit 64
 - initial state 22
 - instructions 9
 - lay out 24
 - lay out the code 31
 - monitor 15, 19
 - number 24
 - override command 44
 - overview 9
 - phase state routine 18
 - set a failure code 38
 - set initial step index 64
 - set or clear tag values 57
 - set the initial state 22
 - set the prestate routine 37
 - set up 24
 - start 45, 46
 - states 12
 - test states 19
 - use 24

equipment phase instructions

overview 9

equipment program

interface tag 51
lay out the code 31
set up the data 51
use 31

example

equipment phases for a machine 25
equipment phases for a tank 25
handle a jam 47
handle fault of a device 39
handle timeout 40
interface tag for a machine 54
interface tags for a tank 53
machine is done resetting 50
procedure for a tank 45
separate code for a machine 33
separate code for a tank 32
sequence equipment phases 45
start a machine 46
state model for a machine 30
state model for a tank 29
tank is done adding water 50

exception

handle 34

external request

hold action 65
respond to lost communication 65

F**fault**

example 39, 40, 47
handle 34
set a failure code 38

H**held state**

use 13

holding state

use 13, 34

I**idle state**

use 13

inhibit

equipment phase 64

initial state

choose 26
set 22

initial step index

set 64

M**monitor**

equipment phase 19

O**override command**

example 47

ownership

overview 15

take with RSLogix 5000 software 15

P**PFL instruction**

use 38

phase

See equipment phase

PHASE data type

members 58

set or clear values 57

use a state bit 37

phase state routine

add 18

prestate routine

add 35, 36

assign 37

example 39, 40, 46, 47

overview 34

use 34

producing state

set up 26

program

equipment phase 31

PXRQ instruction

hold action 65

lost communication 65

R**resetting state**

use 13

restarting state

use 13, 34

routine

add phase state routine 18

RSLogix 5000 software

give command 19

monitor an equipment phase 15, 19

ownership 15

running state

use 13

S**sequencer**

example 45

set

hold action for a PXRQ instruction 65

initial step index 64

PHASE tag values 57

set up

equipment interface tag 51

equipment phase 24

states 26

transitions 41

standby state

set up 26

state model

See states

state routine

See phase state routine

states

compared to PackML 16

compared to S88 16

handle exceptions 34

mark as done 48

overview 12

set the initial state 22

set up transitions 41

step through 15, 19

transition when done 48

transitions between states 14, 41

use 13, 26

use a state bit 37

stopped state

use 13

stopping state

use 13

T**test**

equipment phase 19

transition

step through 19

when done 48

Rockwell Automation Support

Rockwell Automation provides technical information on the Web to assist you in using its products. At <http://www.rockwellautomation.com/support/>, 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://www.rockwellautomation.com/support/>.

Installation Assistance

If you experience an anomaly within the first 24 hours of installation, review the information that is contained in this manual. You can contact Customer Support for initial help in getting your product up and running.

United States or Canada	1.440.646.3434
Outside United States or Canada	Use the Worldwide Locator at http://www.rockwellautomation.com/support/americas/phone_en.html , or contact your local Rockwell Automation representative.

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 to complete the return process.
Outside United States	Please contact your local Rockwell Automation representative for the return procedure.

Documentation Feedback

Your comments will help us serve your documentation needs better. If you have any suggestions on how to improve this document, complete this form, publication [RA-DU002](#), available at <http://www.rockwellautomation.com/literature/>.

Rockwell Automation maintains current product environmental information on its website at <http://www.rockwellautomation.com/rockwellautomation/about-us/sustainability-ethics/product-environmental-compliancepage>.

Rockwell Otomasyon Ticaret A.Ş., Kar Plaza İş Merkezi E Blok Kat:6 34752 İçerenköy, İstanbul, Tel: +90 (216) 56984000

www.rockwellautomation.com

Power, Control and Information Solutions Headquarters

Americas: Rockwell Automation, 1201 South Second Street, Milwaukee, WI 53204 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