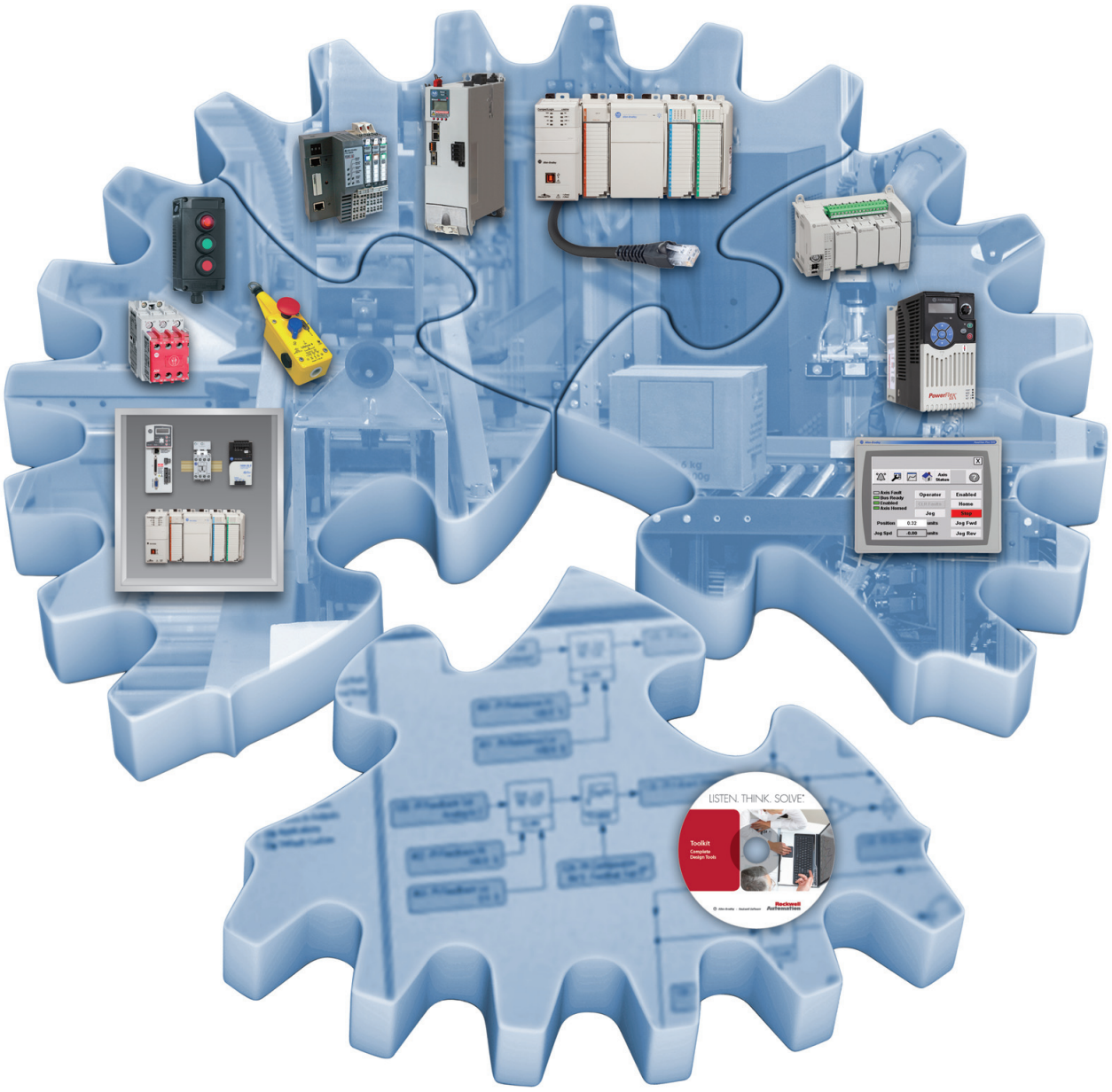
Quick Start

Original Instructions

Allen-Bradley

RAPID Equipment Interface

Installation of the RAPID Equipment Interface Creates a RAPID-ready Machine



Important User Information

Read this document and the documents listed in the additional resources section about installation, configuration, and operation of this equipment before you install, configure, operate, or maintain this product. Users are required to familiarize themselves with installation and wiring instructions in addition to requirements of all applicable codes, laws, and standards.

Activities including installation, adjustments, putting into service, use, assembly, disassembly, and maintenance are required to be carried out by suitably trained personnel in accordance with applicable code of practice.

If this equipment is used in a manner not specified by the manufacturer, the protection provided by the equipment may be impaired.

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.

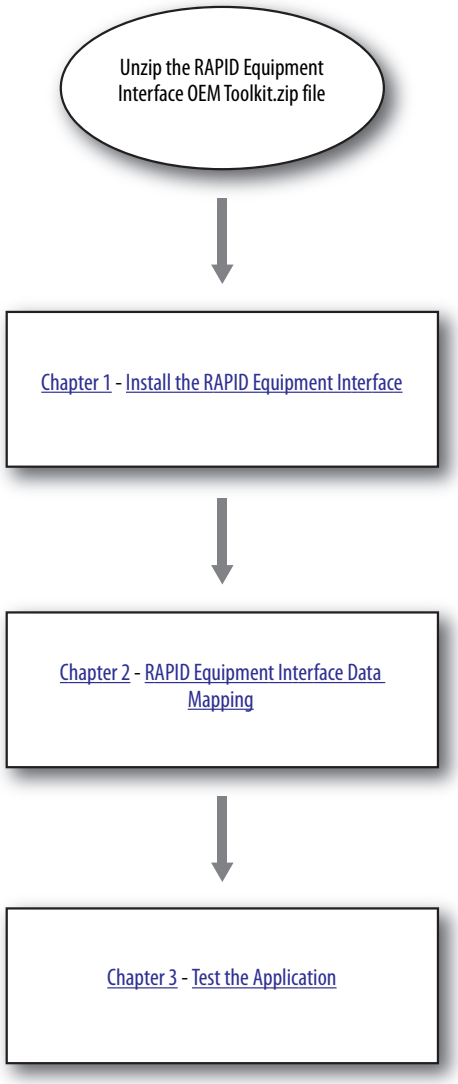
Throughout this manual, when necessary, we use notes to make you aware of safety considerations.

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

Labels may also be on or inside the equipment to provide specific precautions.

	**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.
	**ARC FLASH HAZARD:** Labels may be on or inside the equipment, for example, a motor control center, to alert people to potential Arc Flash. Arc Flash will cause severe injury or death. Wear proper Personal Protective Equipment (PPE). Follow ALL Regulatory requirements for safe work practices and for Personal Protective Equipment (PPE).

You must complete the steps that are described in the following graphic to implement the RAPID Equipment Interface Add-On Instruction. The tasks that are required to complete each step are listed in the respective chapters.



Notes:

This manual contains new and updated information as indicated in the following table.

Topic	Page
Added information how to Download the Toolkit.	11
Updated the RAPID Equipment Interface Tag List.	39
Added Use of the Verification Utility.	49
Added OMAC PackML State Model.	81

Notes:

About This Publication

The beginning of each chapter contains the following information. Read these sections carefully before you begin work in each chapter:

- **Before You Begin** - The chapters in this quick start do not have to be completed in the order in which they appear. However, this section defines the minimum amount of preparation that is required before completing the current chapter.
- **What You Need** - This section lists the tools that are required to complete the steps in the current chapter, including, but not limited to, hardware and software.
- **Follow These Steps** - This section illustrates the steps in the current chapter and identifies the steps that are required to complete the examples.

RAPID Equipment Interface Toolkit

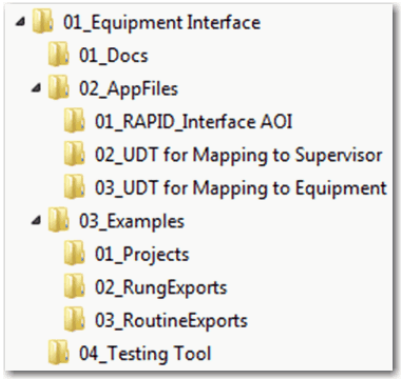
This document is used with the RAPID Equipment Interface OEM Toolkit. You can access the toolkit at the Rockwell Automation Product Compatibility and Download Center (PCDC) that is available at:

http://compatibility.rockwellautomation.com/Pages/home.aspx

The toolkit includes the following:

- Implementation instructions, that is, this document
- Import files that contain the Equipment Interface Add-On Instruction and UDTs
- Example data mappings
- Interface Testing Tool, that is, a FactoryTalk® View Studio application
- Instructional videos
- Add-On Instruction/UDT documentation

The file structure for the toolkit is shown in the follow graphic.



IMPORTANT Add-On Instruction and UDT definitions are already included in the 02_RungExports and 03_RoutineExports files, found in the 03_Examples folder. If you are using these example files to create your interface, the Add-On Instruction and UDT definitions found in the 02_AppFiles folder is not needed.

RAPID Equipment Interface Recommendations

The most efficient way to complete the Add-On Instruction installation is to import the Equipment Interface Add-On Instruction (AOI) and Data Mapping rungs provided with the toolkit. When the rung or routine import method is used, the following are automatically created:

- AOI definitions
- UDT definitions
- An instance of the Add-On Instruction
- Controller-scope tags

Alternatively, you can import the Add-On Instruction and UDT definitions independently, and construct your own EI routines, but only if you are familiar with the RAPID interface.

IMPORTANT If the installation process results in a conflict, due to duplicate UDTs, Add-On Instructions, or associated tags, you must mediate the conflict without changing the RAPID_EI data structure. Changing this structure impacts your machine's ability to communicate with the RAPID System supervisory controller when your equipment is integrated into a production line.

The RAPID_Interface_AOI is protected so modifications are not possible. **Do not modify** the RAPID_Interface_AOI. Changing this structure can result in unexpected equipment behavior, and make it difficult for the system integrator to provide support during line integration and start-up.

Use of the RAPID Interface Equipment Mapping (RAPID_EM) data structure, included in the examples that are provided, is optional. However, following these examples makes sure that you avoid data type mismatches with the RAPID system.

If your existing equipment data tags are in the correct format, these tags can be mapped directly into the RAPID_Interface_AOI instead of using the RAPID_EM tags. Such mapping is useful if MES data structures are in place, or if the equipment was programmed by using Rockwell Automation Power Programming (V4.x) or other ISA-TR88.00.02 based templates.

IMPORTANT Machine programs that are constructed by using Power Programming, version 4.2, or earlier, contain a UDT_Event, which serves as the structure for the FirstOutFault tag. The RAPID_EI and RAPID_EM tag structures contain a UDT with the same name. Overwriting the existing UDT_Event structure with the RAPID UDT_Event structure is typically okay as the new definition adds an additional tag element and does not modify or remove any existing elements. It must be verified that the addition of the category element to the UDT_Event data structure does not adversely affect the existing program or systems that communicate with it.

Disclaimer

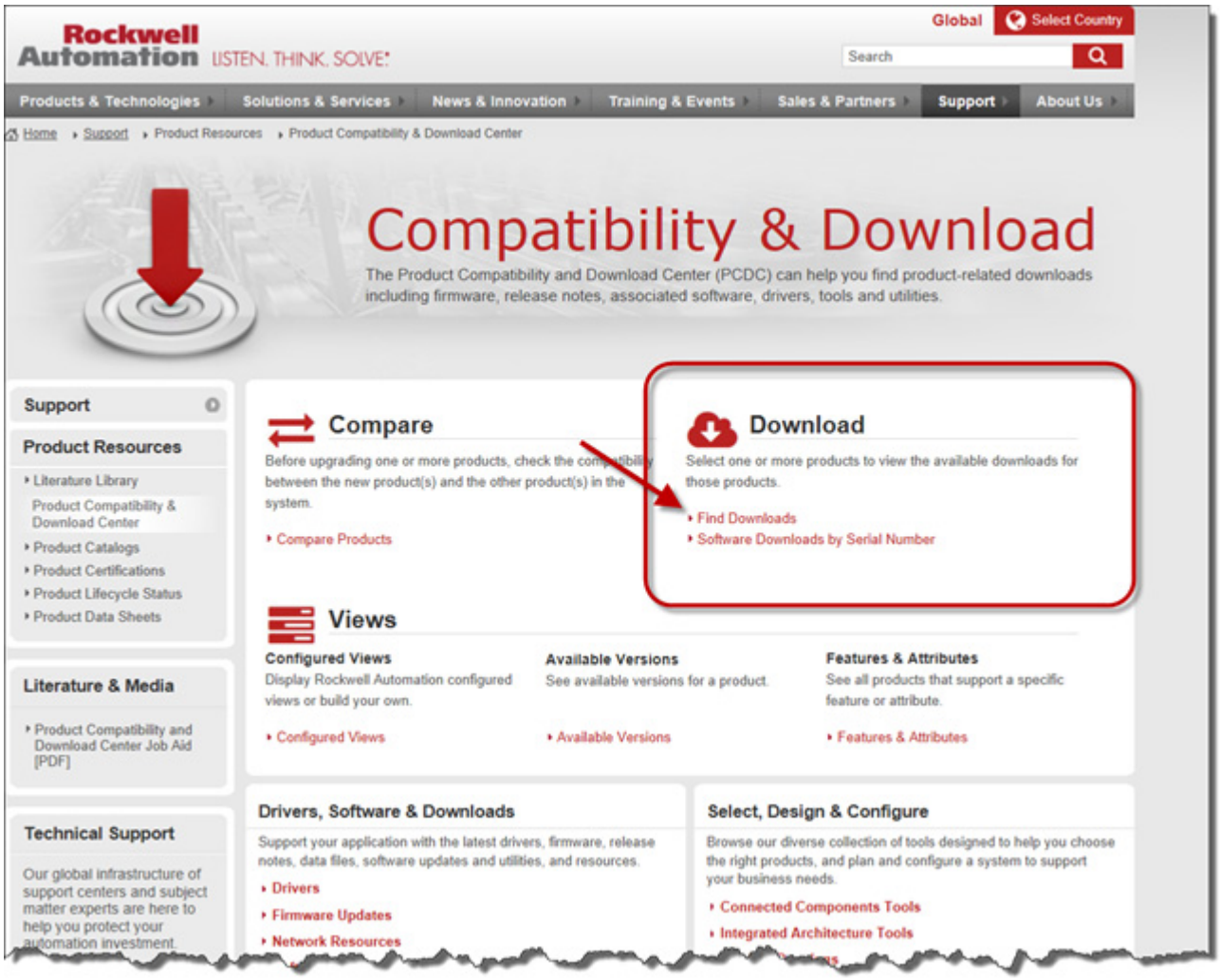
Individuals using this information are responsible for determining that the Rapid Interface Program is acceptable for use in their application. Rockwell Automation, Inc. is not responsible for damages resulting from the use of this information. The illustrations, charts, and examples that are shown in this document are intended solely to clarify the functions of Rockwell Automation products and the RAPID application. We offer options for interfacing with the RAPID application. The requirements that are associated with a specific installation vary, and Rockwell Automation, Inc. cannot assume responsibility or liability for any given application.

No patent liability is assumed by Rockwell Automation, Inc. with respect to use of this information.

Download the Toolkit

Go to the Rockwell Automation Product Compatibility and Download Center (http://www.rockwellautomation.com/rockwellautomation/support/pcdc.page) and download the latest version of the RAPID Equipment Interface OEM Toolkit.

Figure 1 - The Rockwell Automation Product Compatibility and Download Center



1. Click Find Downloads.
2. Type “RAPID” into the Product Search field and select the RAPID Equipment Interface Toolkit in the latest version.
3. In the Downloads overview, click Select Files to choose the files for your Download Cart.

Figure 2 - Product Downloads Page

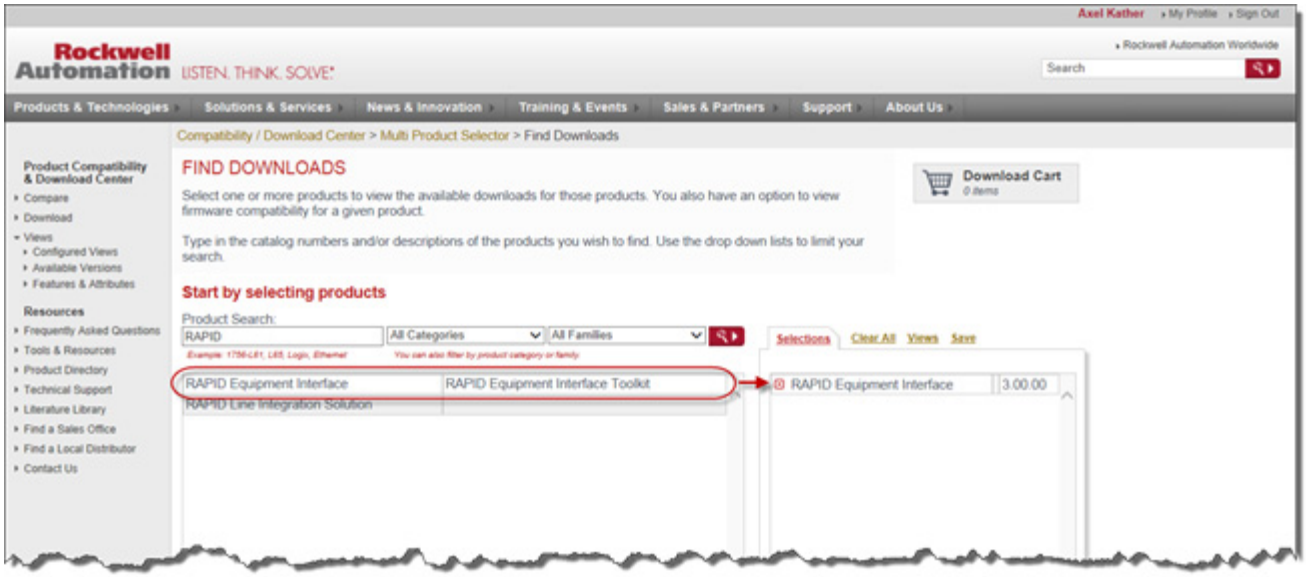
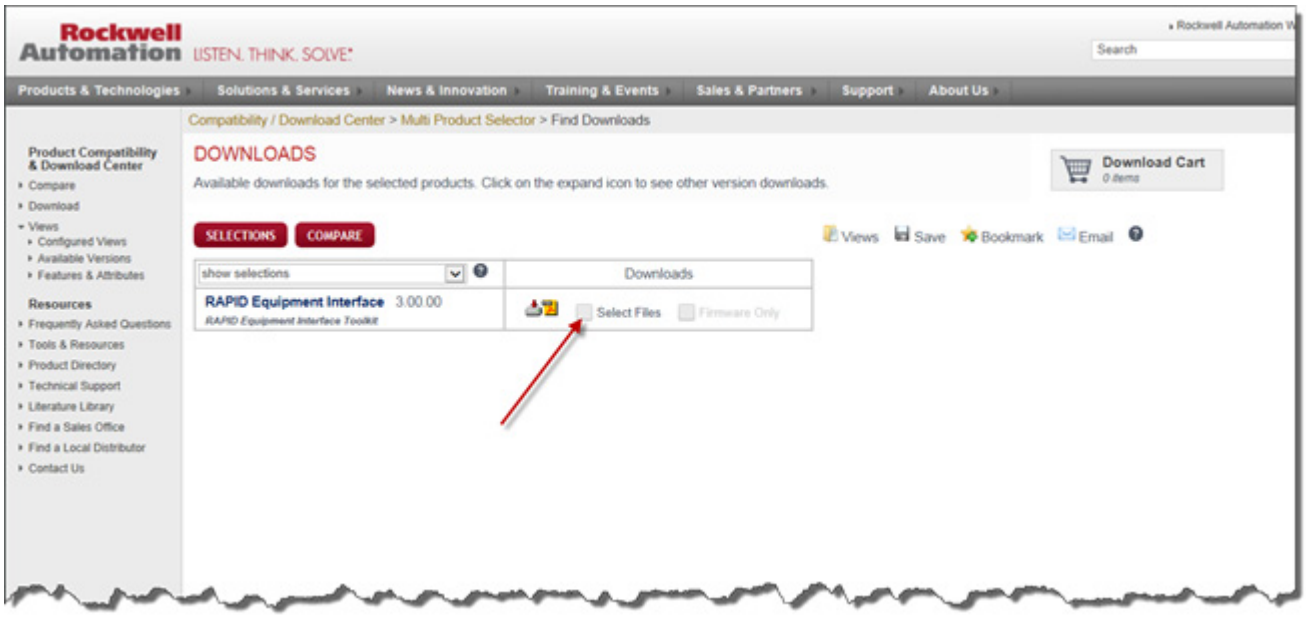
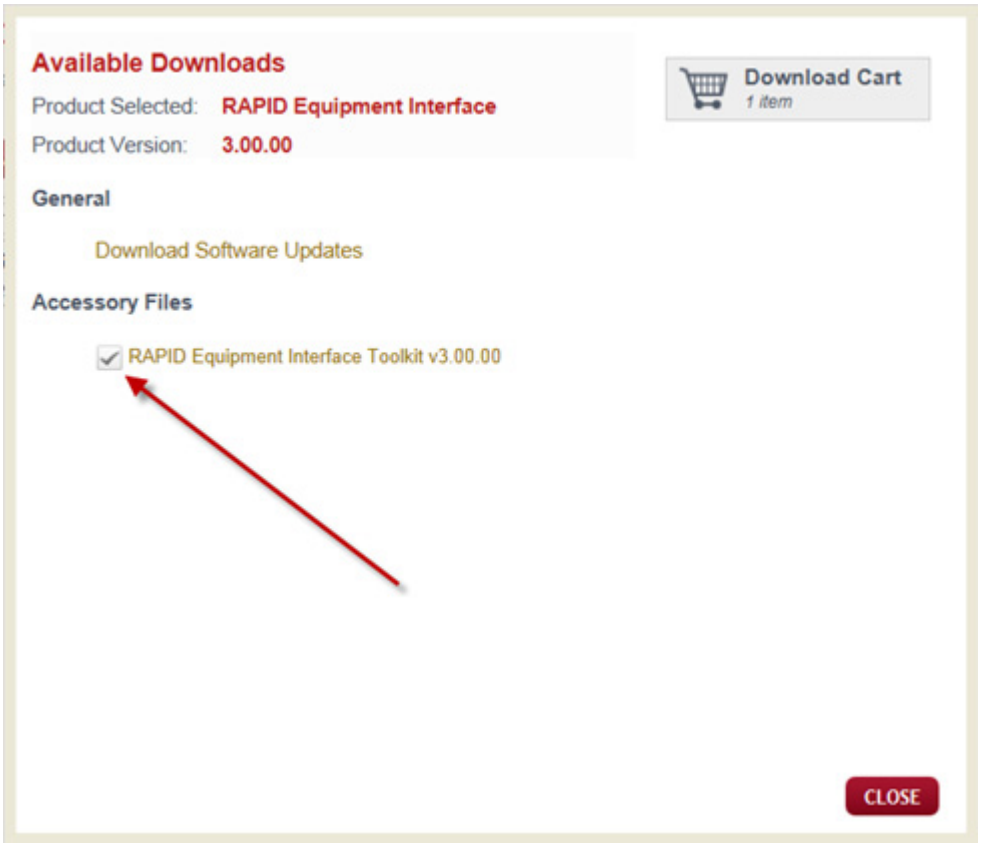


Figure 3 - File Selection Dialogue



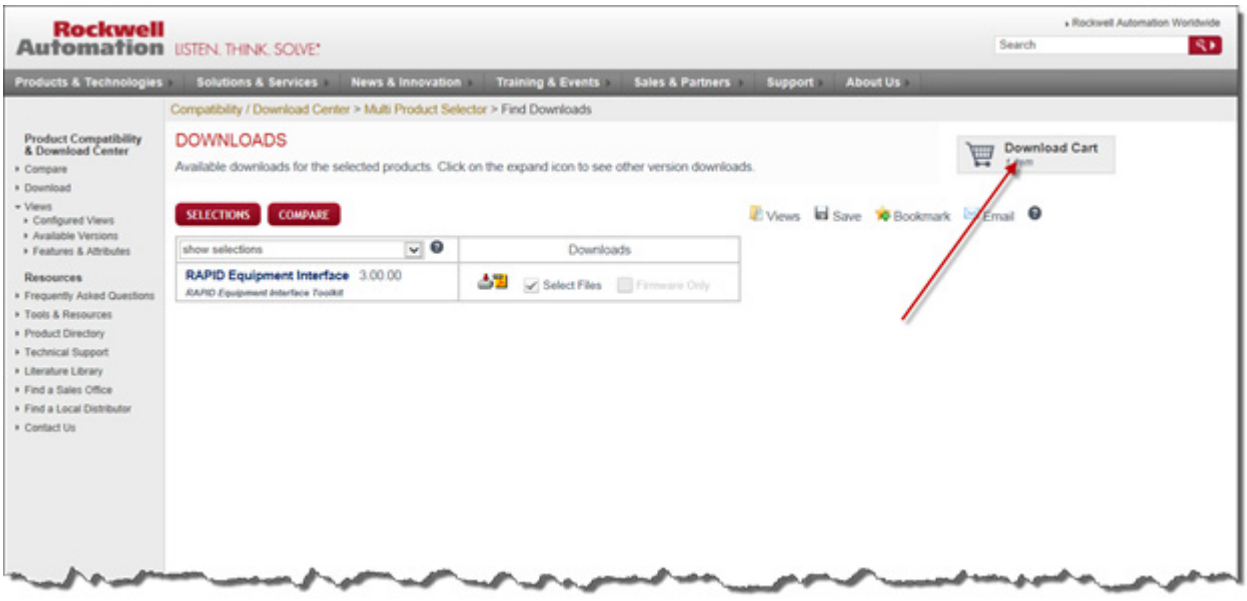
4. Select the RAPID Equipment Interface Toolkit, and then click close. The files are added to your Download Cart.

Figure 4 - Add Files from Available Downloads



5. To access your selected files, click Download Cart.

Figure 5 - Download Cart



6. To start downloading, click Download Now.

Additional Resources

These documents contain additional information concerning related products from Rockwell Automation.

Resource	Description
Industrial Automation Wiring and Grounding Guidelines, publication 1770-4.1	Provides general guidelines for installing a Rockwell Automation industrial system.
Product Certifications website, http://www.rockwellautomation.com/global/certification/overview.page	Provides declarations of conformity, certificates, and other certification details.

You can view or download publications at http://www.rockwellautomation.com/global/literature-library/overview.page. To order paper copies of technical documentation, contact your local Allen-Bradley distributor or Rockwell Automation sales representative.

This document is intended for use by Original Equipment Manufacturers (OEMs) who are preparing their equipment for integration into a production line employing the RAPID Line Integration™ System, version 3.*x*. It explains how to incorporate the RAPID_Interface_AOI into an existing equipment control program by using RSLogix 5000® software.

This document can also be used by system integrators to add RAPID Equipment Interfaces to existing equipment in a manufacturing facility in which they plan to deploy a RAPID system.

To effectively use this document, user must be proficient at programming with RSLogix 5000 software, including the development and use of User Defined Data Types (UDTs) and Add-On Instructions (AOIs). They must also have a thorough understanding of the control program and associated equipment functions, for the target equipment.

Install the RAPID Equipment Interface

This chapter describes how to install the RAPID EI, and associated UDT data structures, into your control program.

By using this chapter, you complete the following tasks:

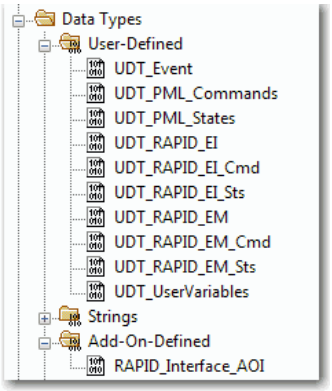
- Import the Add-On Instruction definition.
- Import the associated UDT definitions (RAPID_EI UDT and RAPID_EM UDT).
- Map tags to the interface Add-On Instruction for communication between the RAPID supervisor and the RAPID Equipment Interface.
- Map tags to the interface Add-On Instruction for communication between the RAPID Equipment Interface and the RAPID Equipment Mapping.

You can complete the tasks that are described in this chapter manually. We recommend that you import the files that are provided in folder 02_AppFiles. If you use the base Add-On Instruction and UDT definition L5X files, additional work is required to enter the tag references on the Add-On Instruction. The example files include the mapping UDTs.

Before You Begin

Before you import the RAPID_Interface_AOI, verify that the target program does not have UDT or Add-On Instruction definitions that conflict with definitions being imported.

The following graphic shows the UDT and Add-On Instruction definitions that are imported.



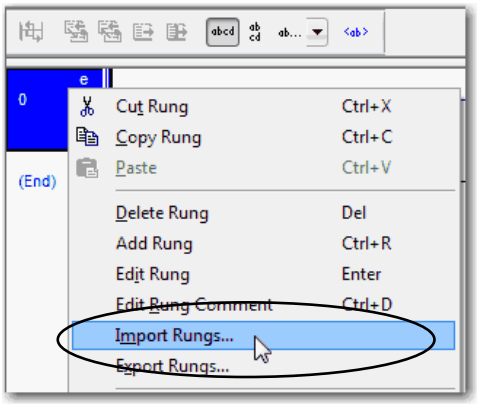
IMPORTANT All instructions included in the import files are compatible with ControlLogix® applications using RSLogix 5000® software version, 16 or later. However, if you are using a ControlLogix application with RSLogix 5000 software, version 20 or later, with Add-On Instructions that were created with RSLogix 5000 software, version 19 or earlier, the Add-On Instructions must be unsigned, imported, and then re-signed.

You can complete the tasks in this chapter manually. However, we recommend that you import the files in the **02_AppFiles** folder of the toolkit.

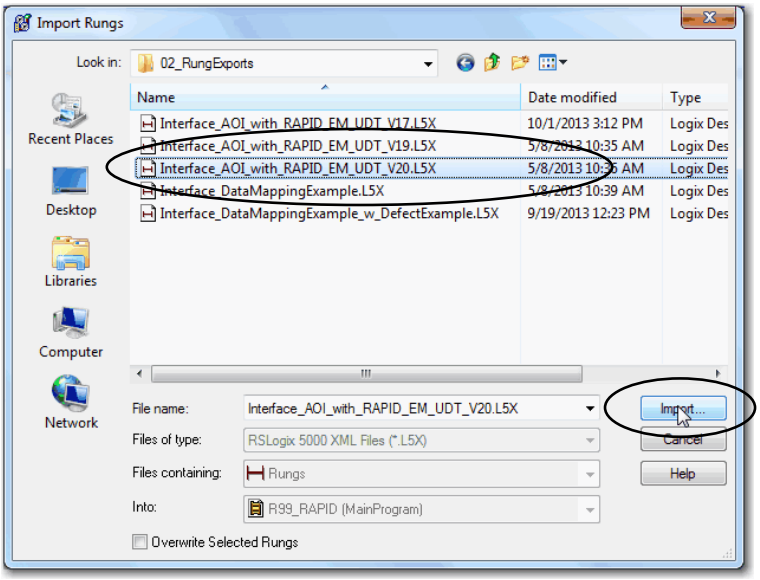
Import the RAPID Equipment Interface Add-On Instruction

Complete the following steps.

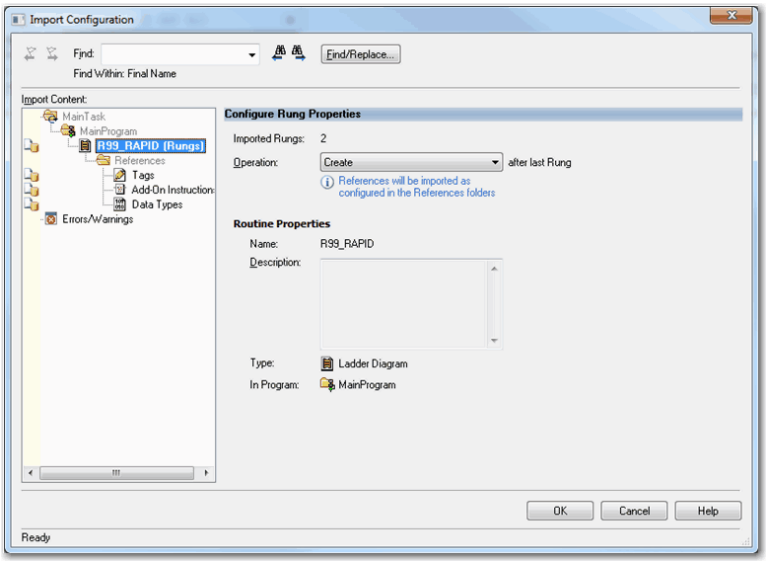
1. In your program logic, right-click a rung and choose Import Rungs.



2. Navigate to the desired L5X file and click Import.



The Import Configuration dialog box appears.



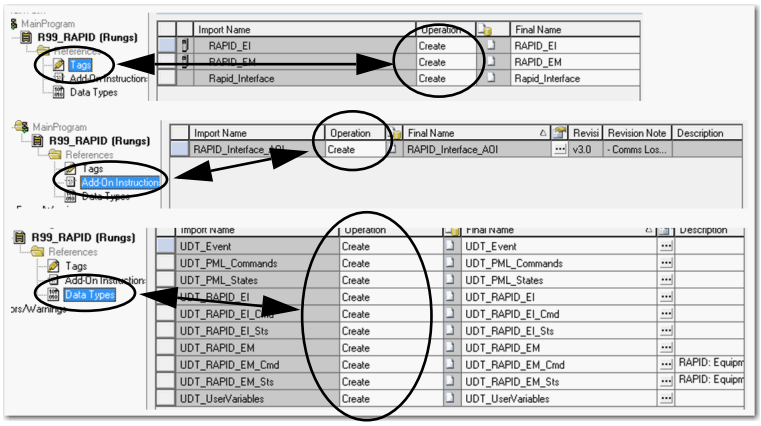
3. Complete the steps that are described in the Verify There Are No Conflicts section.

Verify There Are No Conflicts

Before you click OK to complete the import, verify that there are no conflicting tags, data types, or Add-On Instructions, and consider the following:

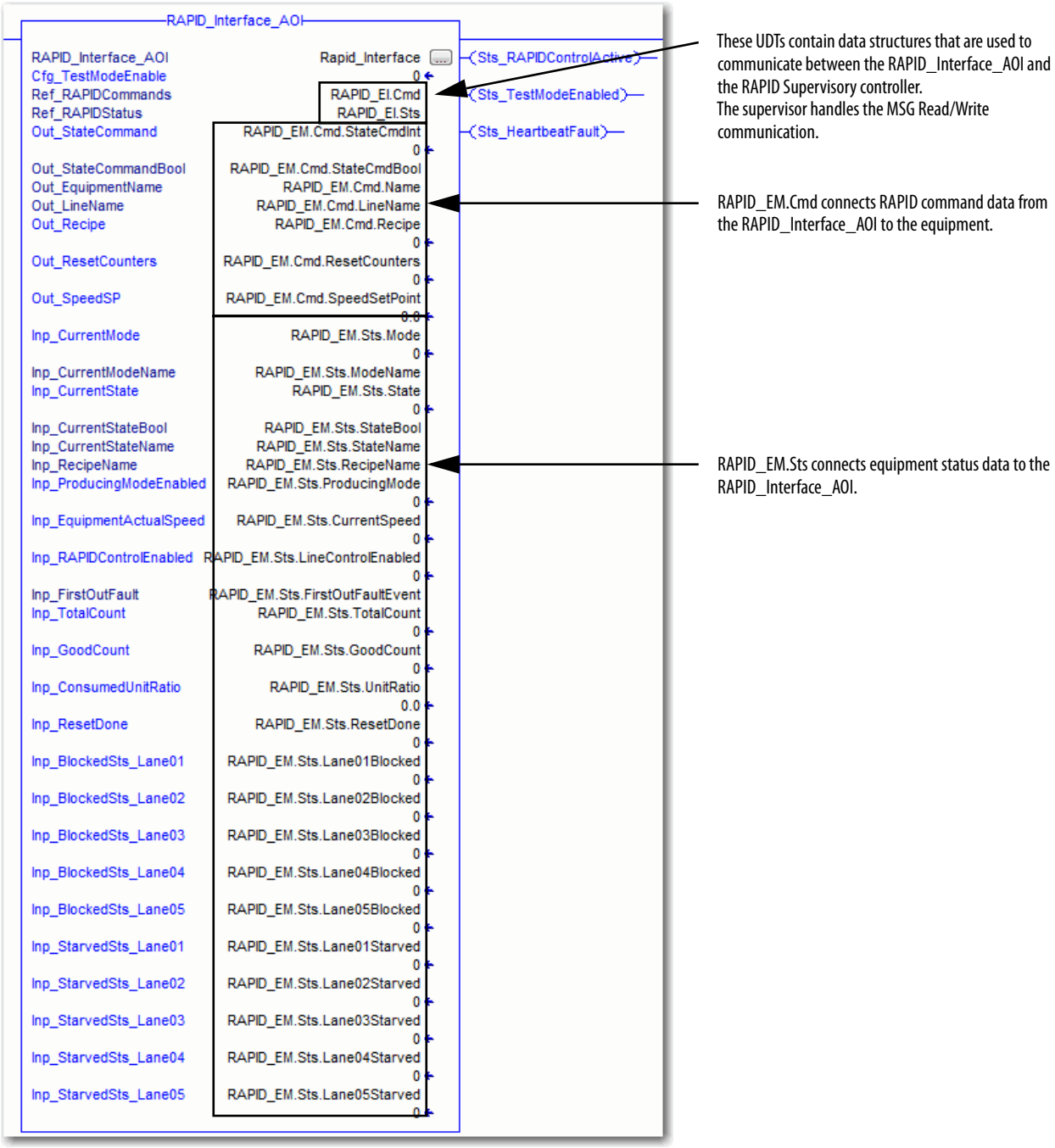
- If all of these sections have an operation of **Create**, there are no conflicts.
- If there are any items that have an operation of **Use Existing**, you must verify that there is not a conflict or that setting the operation to **Overwrite** does not cause a problem with any existing tags or programming.

  After the import operation is complete, the UDTs in the target program must match the RAPID definition of all UDTs or communications are impacted.



Click OK to complete the import.

When the import process is complete, the rung shown is added to your program. The tags are mapped to the Add-On Instruction as shown.



The RAPID_Interface_AOI is installed. The data structures are created and mapped to the Add-On Instruction.

RAPID Equipment Interface Data Mapping

This chapter describes how to map equipment data to the RAPID_EM data structure.

By using this chapter, you complete the following tasks:

- Find or establish the required data elements in your equipment control program.
- Map these data elements to the RAPID_EM data structure.

You can complete data mapping by using the sample code import files that are provided in folder 02_AppFiles or by creating your own custom code. Regardless of whether you start with our sample code or create your own code, you must follow the RAPID data mapping instructions that are described in this chapter.

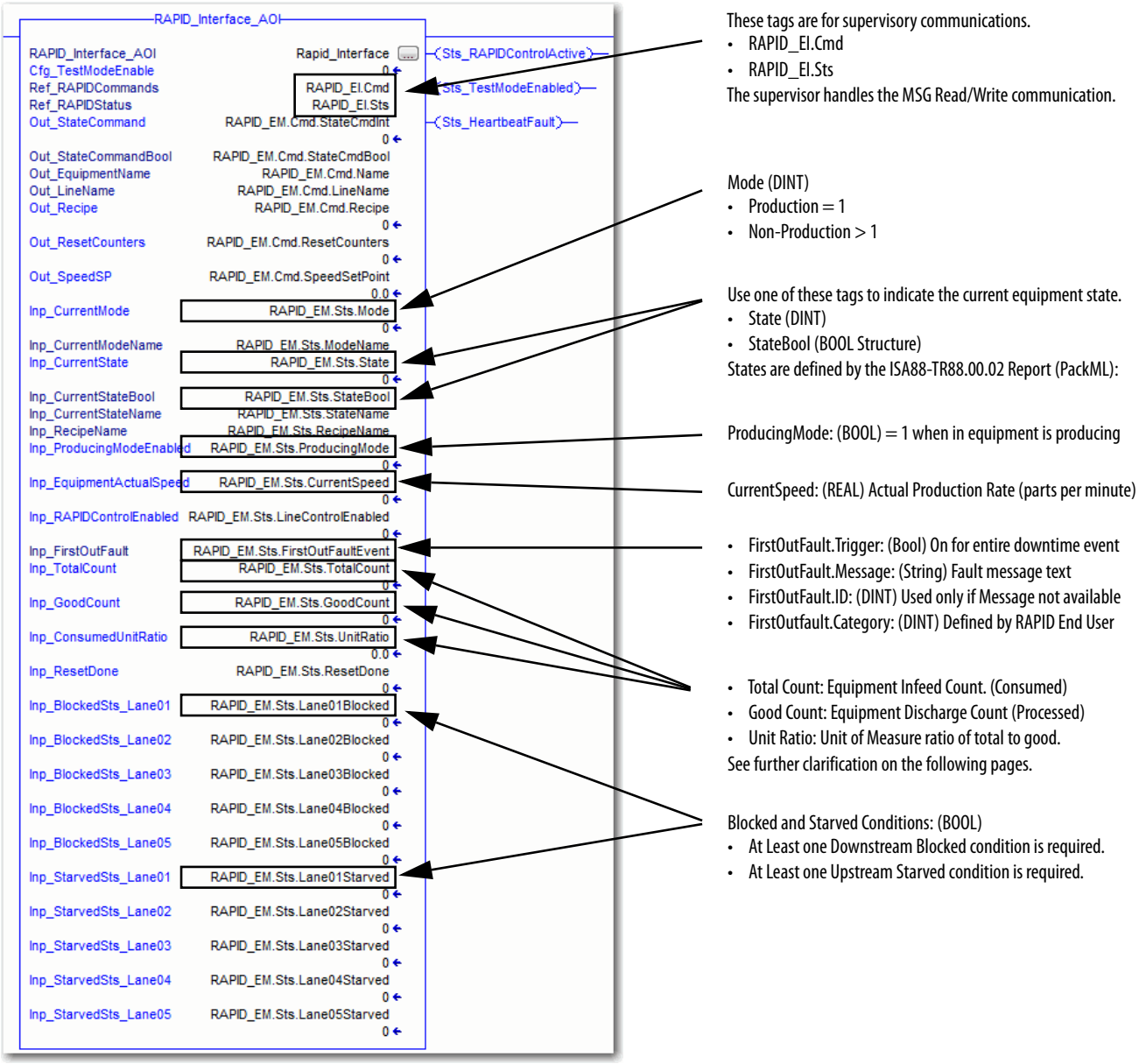
If the target equipment program was programmed per ISA-TR88.00.02 / PackML, note the information that is provided on page 33 regarding an alternative method to implementing the equipment interface for PackML based programs.

For more information on the data structure, see Appendix A, RAPID_EM Data Structure on page 39.

Data Mapping for RAPID Line Performance

AOI Tag Mapping Requirements for RAPID Line Performance Functions

The following graphic shows the minimum tag mapping requirements for equipment that is installed in a production line with a RAPID **Line Performance** system.



These tags are for supervisory communications.
- RAPID_EI.Cmd
- RAPID_EI.Sts

The supervisor handles the MSG Read/Write communication.

Mode (DINT)
- Production = 1
- Non-Production > 1

Use one of these tags to indicate the current equipment state.
- State (DINT)
- StateBool (BOOL Structure)

States are defined by the ISA88-TR88.00.02 Report (PackML):

ProducingMode: (BOOL) = 1 when in equipment is producing

CurrentSpeed: (REAL) Actual Production Rate (parts per minute)

- FirstOutFault.Trigger: (Bool) On for entire downtime event
- FirstOutFault.Message: (String) Fault message text
- FirstOutFault.ID: (DINT) Used only if Message not available
- FirstOutfault.Category: (DINT) Defined by RAPID End User

- Total Count: Equipment Infeed Count. (Consumed)
- Good Count: Equipment Discharge Count (Processed)
- Unit Ratio: Unit of Measure ratio of total to good.

See further clarification on the following pages.

Blocked and Starved Conditions: (BOOL)
- At Least one Downstream Blocked condition is required.
- At Least one Upstream Starved condition is required.

Data Mapping Logic Requirements for RAPID Line Performance Functions

The following sections contain examples of how data, including part counts, can be mapped from the existing equipment control system tags into the RAPID_EM structure. The RAPID_EM data mapping is highlighted in green in the following examples. These code examples are provided as rung and routine .L5X files that can be imported directly into your existing equipment control program:

- Part Counting - Total and Good Parts Counts Provided
- Part Counting - Total Parts Are Calculated
- Part Counting - Unit Ratio (Inp_ConsumedUnitRatio)
- Blocked and Starved Conditions (Inp_BlockedStatus_Lanexx, Inp_StarvedStatus_Lanexx
- Equipment Mode Status (Inp_CurrentMode, Inp_CurrentModeName)
- Equipment State Status (Inp_CurrentState, Inp_CurrentStateBool)
- First Out Fault Status (Inp_FirstOutFault)
- Current Speed Status (Inp_EquipmentActualSpeed)

In these examples, part counts are described by using the terms that are commonly associated with OEE and Machine Performance applications, and ISA-TR88.00.02 / PackML standards. The RAPID Equipment Interface requires that **Total Counts** and **Good Counts** be mapped to the EM data structure. If they are provided by using the

PackML (ISA-TR88.00.02) data mapping method, it is important to make sure that the part counts adhere to the rules associated with the use of the inp_ConsumedUnitRatio value detailed below, which is NOT included in PackML.

A Unit Ratio value also must be provided that indicates the ratio of Total Parts to Good Parts. Bottles per Case as an example, which can vary by each SKU produced on a machine. It is understood that on some equipment, Total or Good may have to be calculated by adding or subtracting a bad part count to the part count that is provided. The calculation could be **Total - Bad = Good** or **Good + Bad = Total**; however the unit ratio must be factored into this math.

RAPID/Overall Effectiveness (OEE) Terms	ISA-TR88.00.02 / PackML Terms	Measured At	Example: Cartoner Making Twelve Packs
Total Parts	Consumed Parts	Infeed	Cans
Good Parts	Processed Parts	Discharge	Cartons
Unit Ration	N/A	SKU-based	Value =12

When Total and Good parts are natively provided by the equipment sensors, the counter values can be free running and rollover at any value. RAPID only considers positive incremental count values as valid counts as a means of letting a rollover to occur without disrupting part count accumulation over the course of long production runs.

When a good or total count must be calculated, give this need special consideration. The calculation becomes invalid if a part count that is used in the calculation rolls over at another time than the other part count used in the calculation. The part counting location (infeed/discharge) also has to be considered closely so that the Good and Total parts are reported with the proper Unit Ratio as shown in the previous table.

The Unit Ratio, that is, tag Inp_ConsumedUnitRatio, must accurately represent the following:

- The Ratio of Infeed parts to Discharge parts
- The Ratio of the Inp_TotalParts unit of measure to the Inp_GoodParts unit of measure

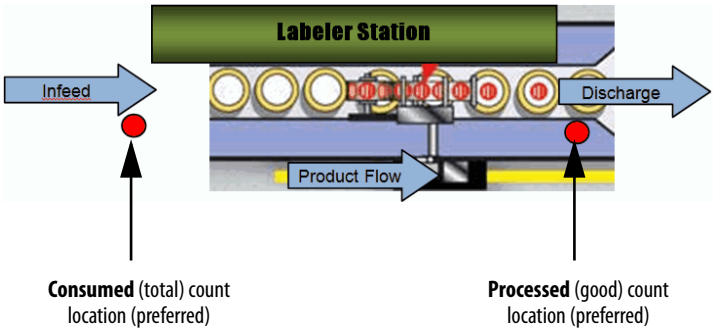
You must follow both rules regardless of where parts counts are taken.

Part Counting - Total and Good Parts Counts Provided

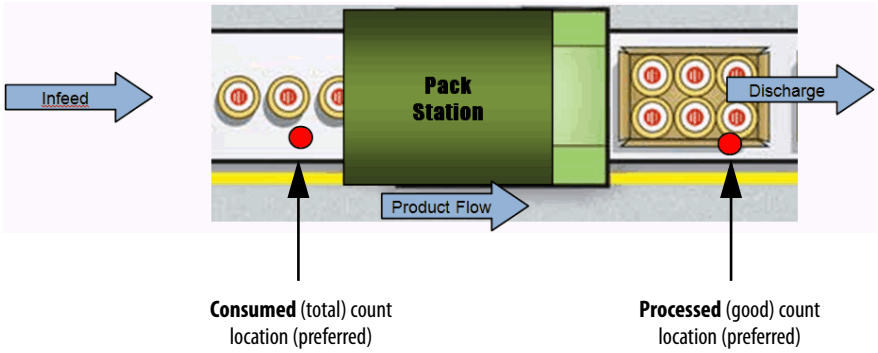
IMPORTANT This example shows the preferred native parts counting method.

The graphics below show the locations, that is, the labeler station and pack stations, where parts are typically counted on most equipment.

Using these counting locations is the best way to make sure that **all** losses within the confines of the equipment unit are accounted for. This method is also the simplest to implement. The Unit Ratio is provided so the Bad (Defective) part count can be properly calculated by the supervisory system and so part quantities can be tracked properly through the entire line.

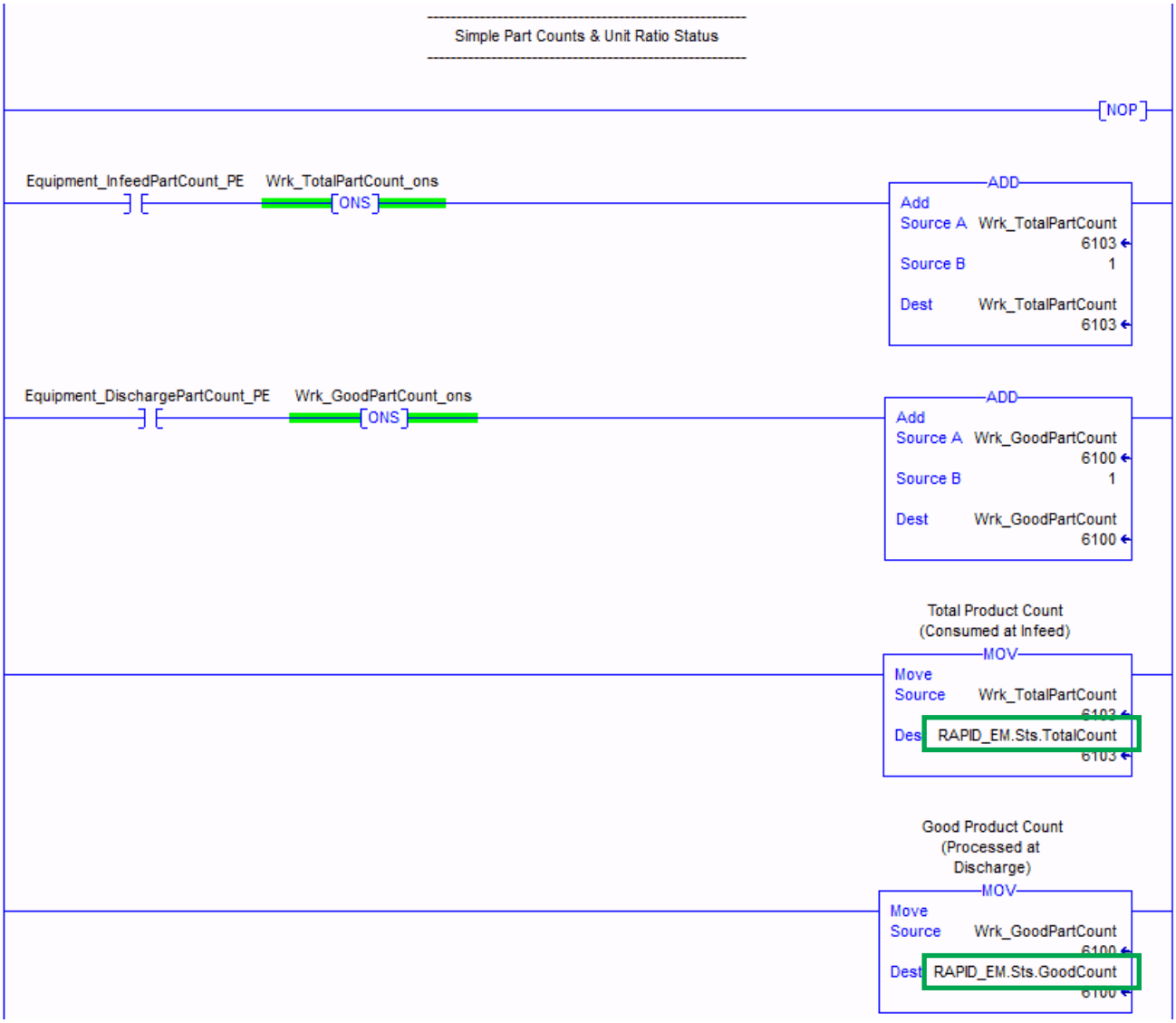


Unit Ratio = (1) Quantity of consumed parts that are needed to make one processed part.



Unit Ratio = (6) Quantity of consumed parts that are needed to make one processed part.

In the following code example, a part is counted by using an ADD instruction each time the photo-eye transitions from off to on. Additional logic can be required if all photo-eye off to on transitions cannot be trusted to be a new part. For example, de-bounce logic can be required. In general, the part counting logic is relatively simple when using the preferred counting method.



Part Counting - Total Parts Are Calculated

IMPORTANT This example shows an optional native parts counting method.

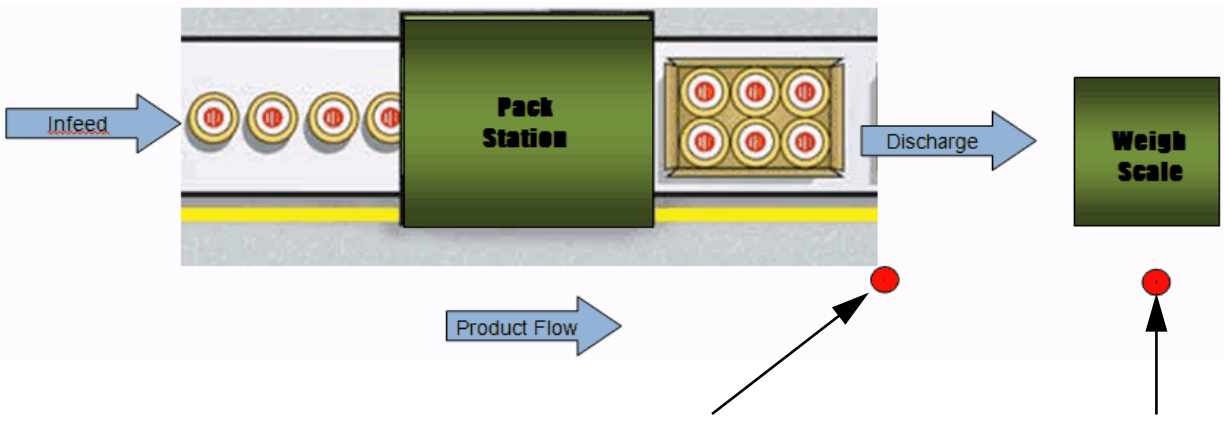
When a machine has only one counter at its discharge and bad parts are counted at a downstream inspection/reject device, Total parts count must be calculated based on the discharge count and unit ratio value. The Good parts count must be calculated by using the following formula:

Total Parts = (Discharge sensor counts * Unit Ratio)
– Unit of Measure is Bottles (Reported as if the count was measured at the infeed.)

Bad Parts = (Discharge sensor counts * Unit Ratio)
– Unit of Measure is Bottles

Good Parts = (Total Parts - Bad Parts) / Unit Ratio)
– Unit of Measure is Cases

IMPORTANT It can appear in this calculation that the unit ratio is not necessary. If it is not used, the result is a unit ratio that is not consistent with the first unit ratio rule that is described in section Part Counting - Total and Good Parts Counts Provided.

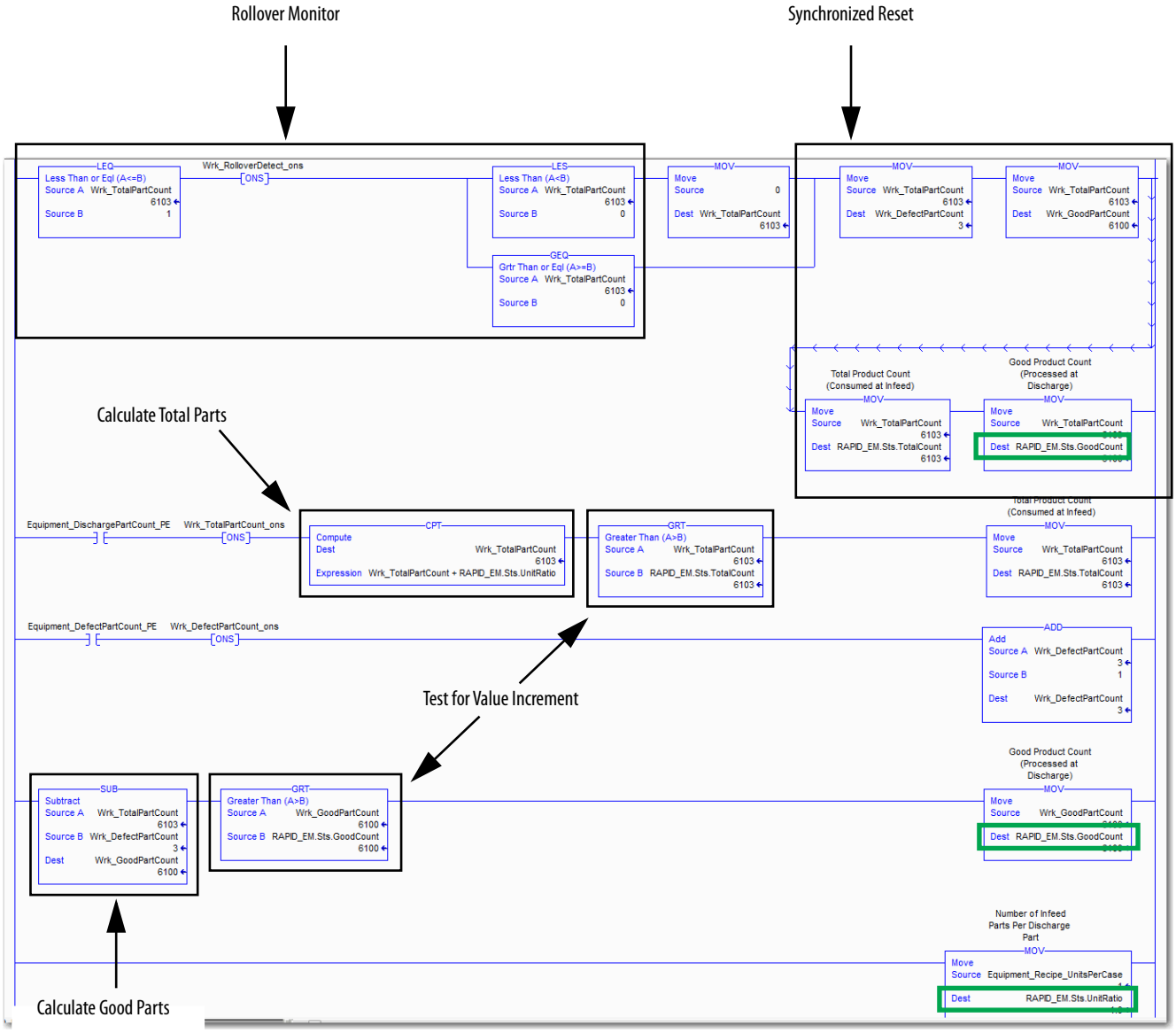


IMPORTANT: Rollover of all counter values that are used in a calculation have to be synchronized so the mathematical difference between their absolute values remains valid.

In the following code example, the Total part count increments by the amount of the Unit Ratio by using the compute (CPT) instruction. This is because the case packer in this example is discharging 6 bottles for every case that is counted.

This example also shows how to make sure that a rollover of a part counter does not cause the calculation to become invalid. This logic confirms that an invalid count value does not get through by checking that it is greater than the previous value, unless it is a valid rollover as detected in the first rung.

The part counting logic becomes slightly more complicated when having to use calculations to determine the count values, however if logic such as what is shown in these examples is used, reliable part counts can be achieved.

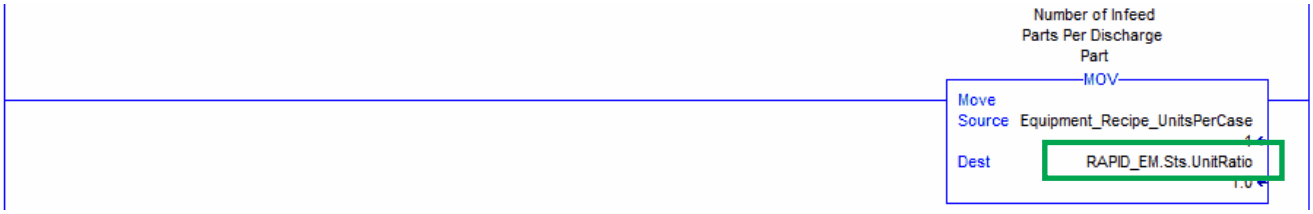


Part Counting - Unit Ratio (Inp_ConsumedUnitRatio)

The Unit Ratio is simply the quantity of Total parts (as measured or calculated at the equipment infeed) is required to make one Good part (as measured or calculated at the equipment discharge). For example, the number of bottles needed per case, or the number of cases per pallet.

Most equipment control systems that group products in some manner have a tag that tells you the number of infeed products that are required to create one discharged product. If this information is not available in the machine, it likely is always one or it can be derived from the recipe number that is being sent to the equipment from the RAPID supervisor or provided from another system.

For equipment such as a capper, labeler, or filler, the unit ratio is typically a value of 1, because the quantity of infeed parts (bottles/caps) is the same as the quantity of discharged parts.



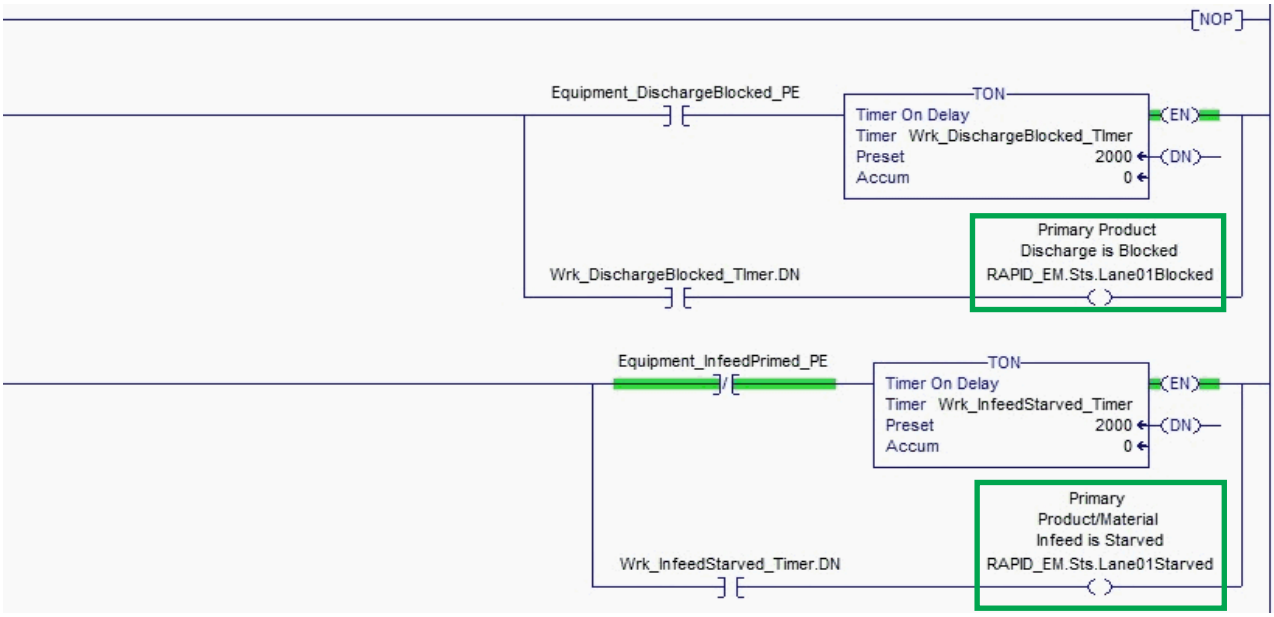
The Unit Ratio, that is, tag Inp_ConsumedUnitRatio, must accurately represent the following:

- The Ratio of Infeed parts to Discharge parts
- The Ratio of the Inp_TotalParts unit of measure to the Inp_GoodParts unit of measure

You must follow both rules regardless of where parts counts are taken.

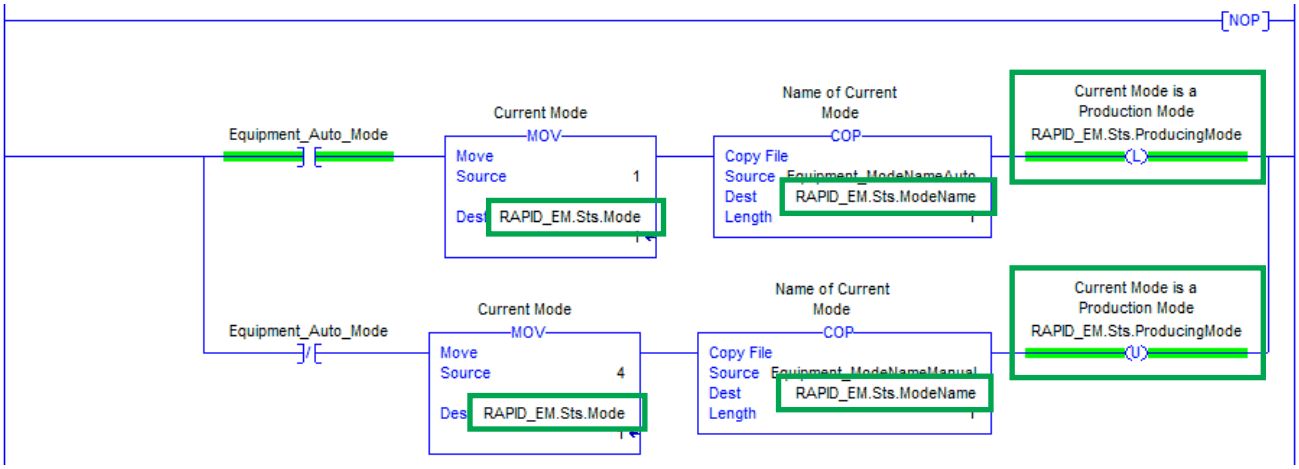
Blocked and Starved Conditions (Inp_BlockedStatus_Lanexx, Inp_StarvedStatus_Lanexx

This example, shows simple logic to detect **in-feed starved** and **discharge blocked** conditions. Additional logic can be required if the machine uses multiple in-feed or discharge lanes.



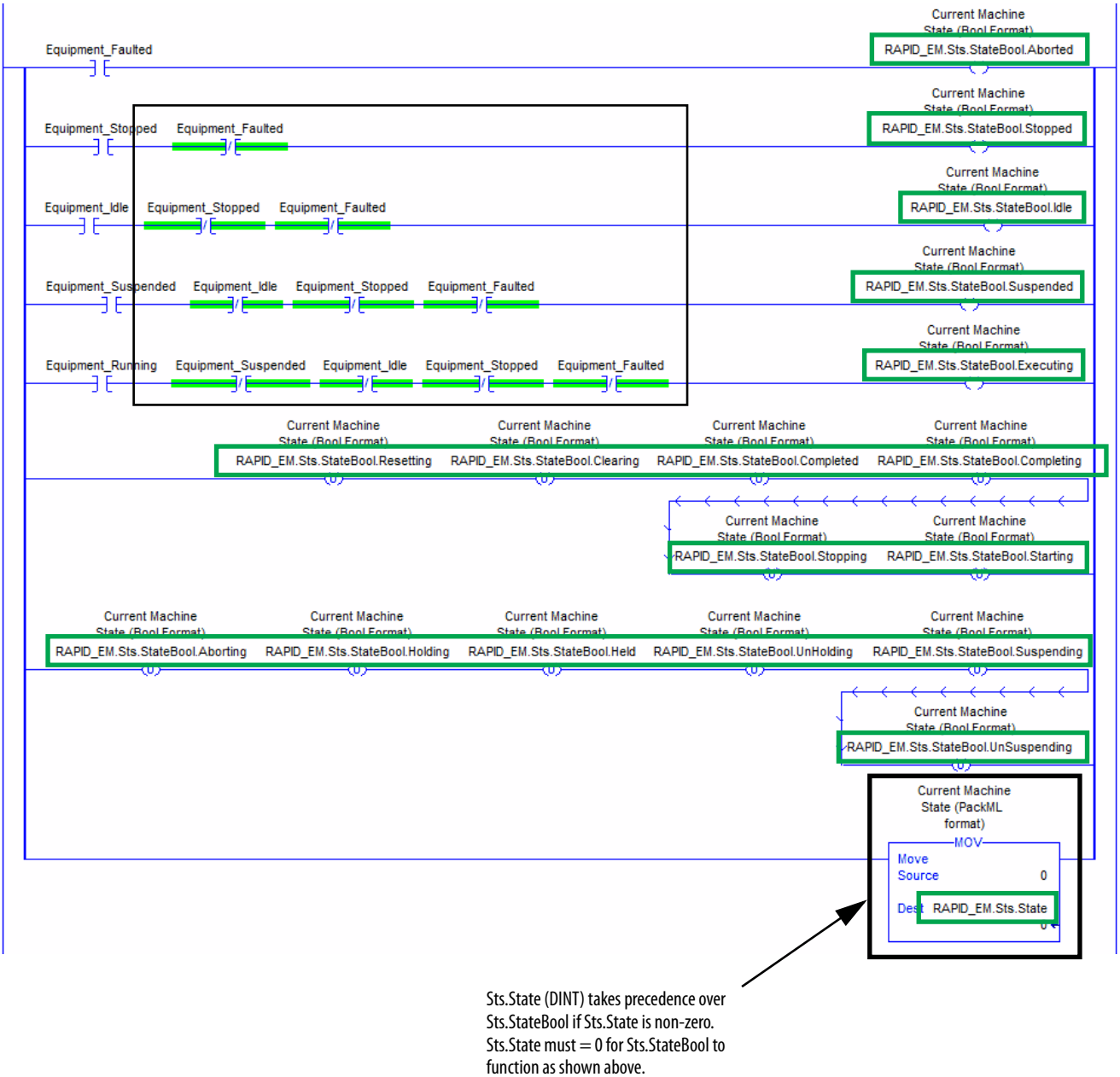
Equipment Mode Status (Inp_CurrentMode, Inp_CurrentModeName)

As a minimum, RAPID Ready machines must report when they are in production mode (value=1), and when they are not. The example below uses the **Auto_Mode** tag from the existing program to initiate the value for Mode, ModeName, and ProducingMode. If more modes are available in the existing equipment control program, these values can also be reported to the RAPID System by using the values that are shown in Appendix A of this document.



Equipment State Status (Inp_CurrentState, Inp_CurrentStateBool)

Equipment state can be reported by using either StateBool or State (DINT) tags. In this example, the Boolean structure is active, so the DINT value is set to zero. All unused Boolean tags must be set to zero. Also included in this program are permissives to make sure that only one machine state is reported at a time. These permissives prioritize the faulted state as the highest and the running state as the lowest, to make sure reporting accuracy.



Sts.State (DINT) takes precedence over Sts.StateBool if Sts.State is non-zero. Sts.State must = 0 for Sts.StateBool to function as shown above.

First Out Fault Status (Inp_FirstOutFault)

In this example, we show how to manipulate the FirstOutFaultEvent Message, ID, Category, and Trigger. To avoid an **unknown fault** in the RAPID Performance Management system, the Message and/or ID value must be present before the Trigger bit is turned on. Adding a permissive that requires one of these values to be present, before turning on the trigger bit, is a good option to consider.

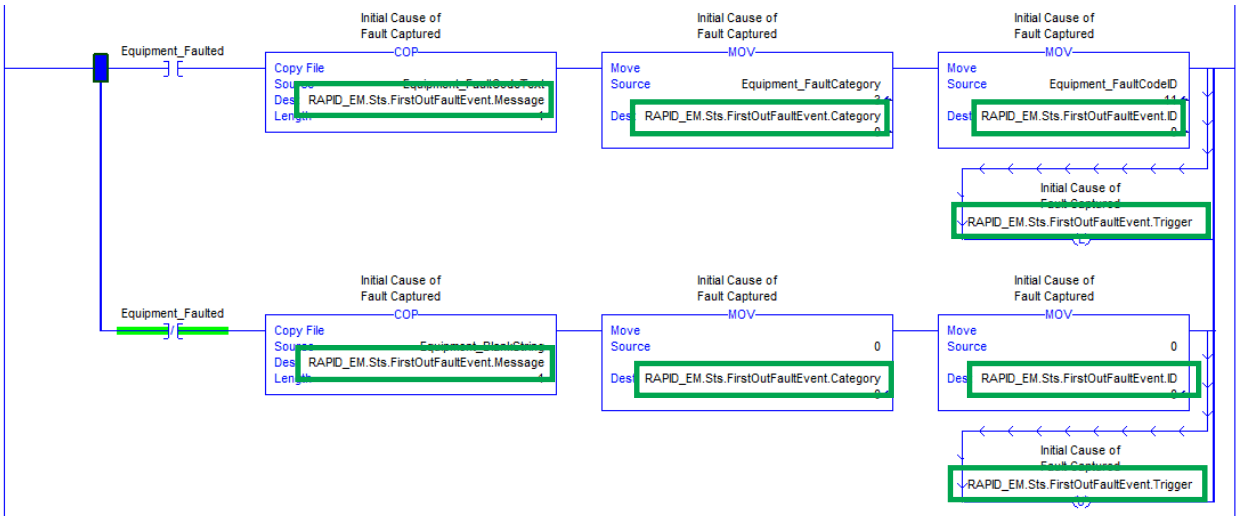
IMPORTANT Do not use dashes in fault message strings. The dash character is viewed as a delimiter by RAPID, and can cause the event strings to be truncated.

Messages are preferable to IDs in RAPID. Fault ID and Message values are defined by the equipment/machine programmer. If the programmer uses an ID instead of a message, he must provide a cross-reference list of IDs and corresponding messages to the RAPID system integrator, before RAPID installation. Create this cross-reference list in electronic format.

Fault categories 1...10 are assigned during RAPID installation. In addition to the example code, it can be necessary to map values from the equipment program into the Fault Category, FaultCodeText, and/or FaultCodeID. This is expected when these values are not already provided within the existing equipment control program. If no category number is provided with the fault event, then the event is categorized as **NC** or **General** Fault. Detailed and accurate fault reporting makes it easier to determine the root cause of machine and line level downtime, by using RAPID reports and data.

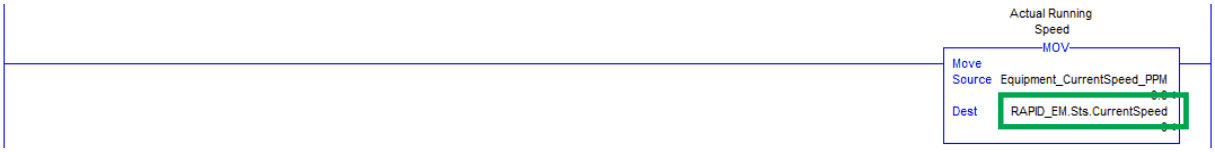
IMPORTANT Categories are assigned by a RAPID system integrator or end user.

The following graphic shows programming code that is used with this example.



Current Speed Status (Inp_EquipmentActualSpeed)

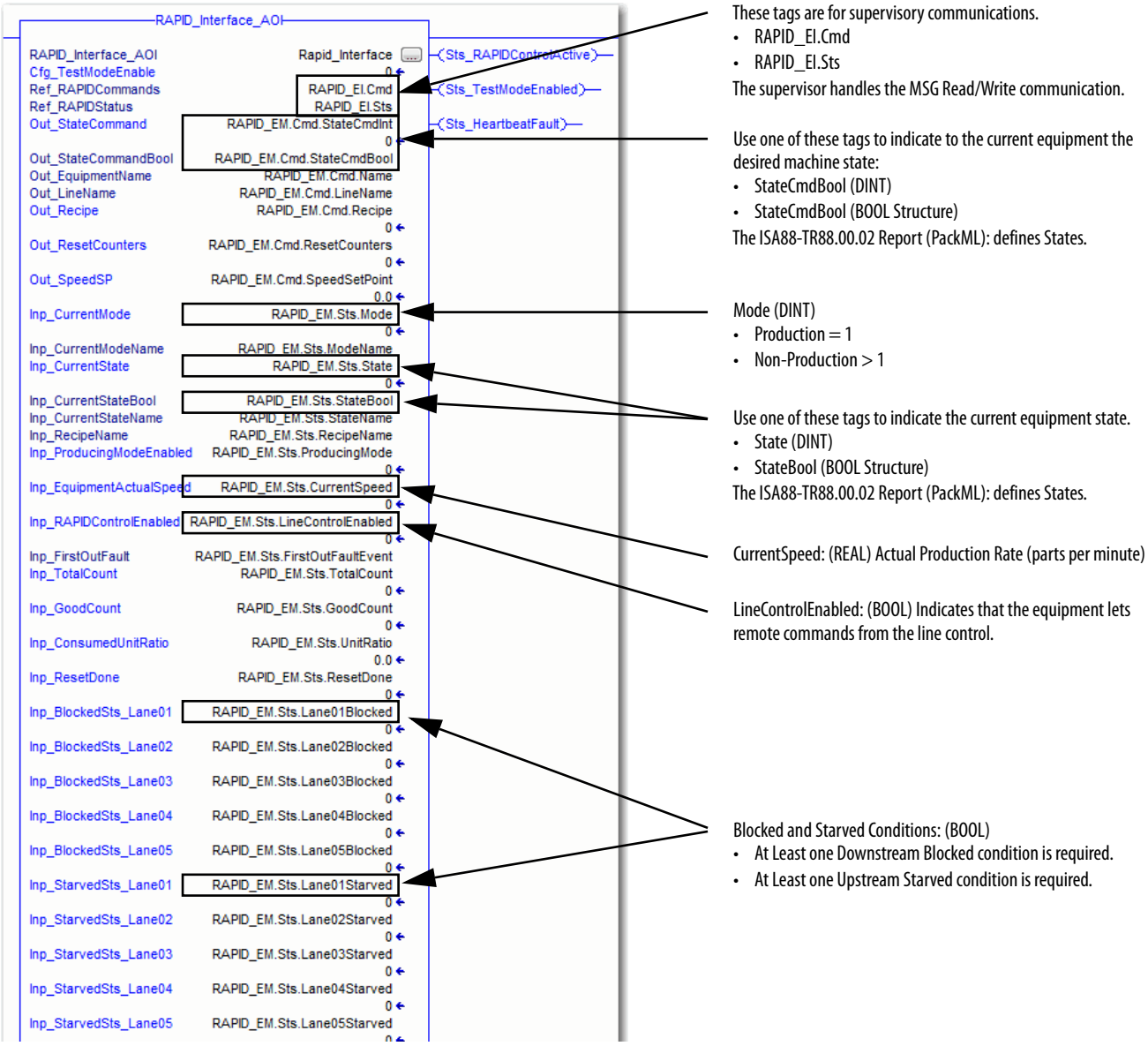
The equipment speed status is typically derived from a main drive actual speed parameter, or a virtual master axis velocity. Always report this value in parts or units per minute at the discharge of the equipment. The following graphic shows programming code that is used with this example.



Data Mapping for RAPID Line Control

AOI Tag Mapping Requirements for RAPID Line Control Functions

The following graphic shows the minimum required tag mapping for **Line Control** functionality.



These tags are for supervisory communications.
- RAPID_EI.Cmd
- RAPID_EI.Sts

The supervisor handles the MSG Read/Write communication.

Use one of these tags to indicate to the current equipment the desired machine state:
- StateCmdBool (DINT)
- StateCmdBool (BOOL Structure)

The ISA88-TR88.00.02 Report (PackML): defines States.

Mode (DINT)
- Production = 1
- Non-Production > 1

Use one of these tags to indicate the current equipment state.
- State (DINT)
- StateBool (BOOL Structure)

The ISA88-TR88.00.02 Report (PackML): defines States.

CurrentSpeed: (REAL) Actual Production Rate (parts per minute)

LineControlEnabled: (BOOL) Indicates that the equipment lets remote commands from the line control.

Blocked and Starved Conditions: (BOOL)
- At Least one Downstream Blocked condition is required.
- At Least one Upstream Starved condition is required.

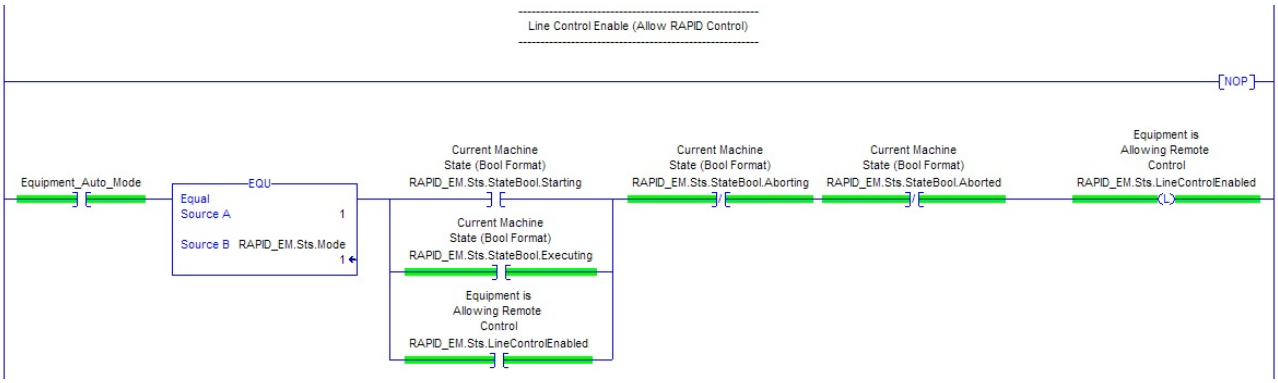
Data Mapping Logic for Line Control Functions

The following examples show how data, including part counts, can be mapped from the existing equipment control system tags into the RAPID_EM structure. The RAPID_EM data mapping is highlighted in green in the following examples. These code examples are provided as rung and routine .L5X files that can be imported directly into your existing equipment control program.

IMPORTANT This code is only an example. Your machine can require different interlocking to function properly. Power Programming, of which samples from are shown in the section, is NOT required to implement the RAPID Equipment Interface.

Line Control Enable (Inp_RAPIDControlEnabled)

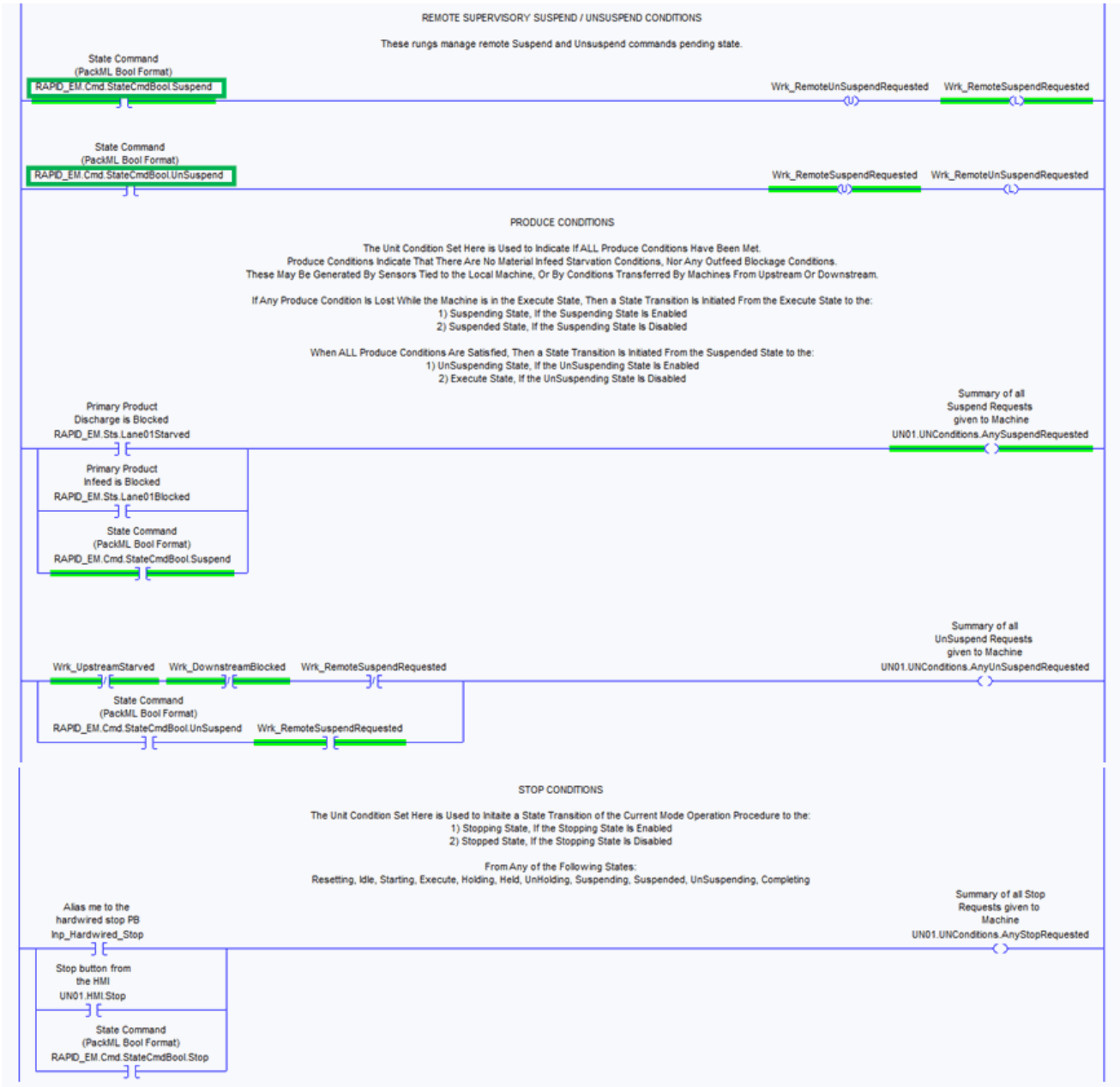
The state commands are activated by the RAPID Equipment Interface only if the RAPID_EM.Sts.LineControlEnabled (Inp_RAPIDControlEabled) bit is turned on by the equipment program. This bit tells the RAPID Line Control system that the machine is able to respond to Start, Stop, Suspend, or Unsuspend commands.



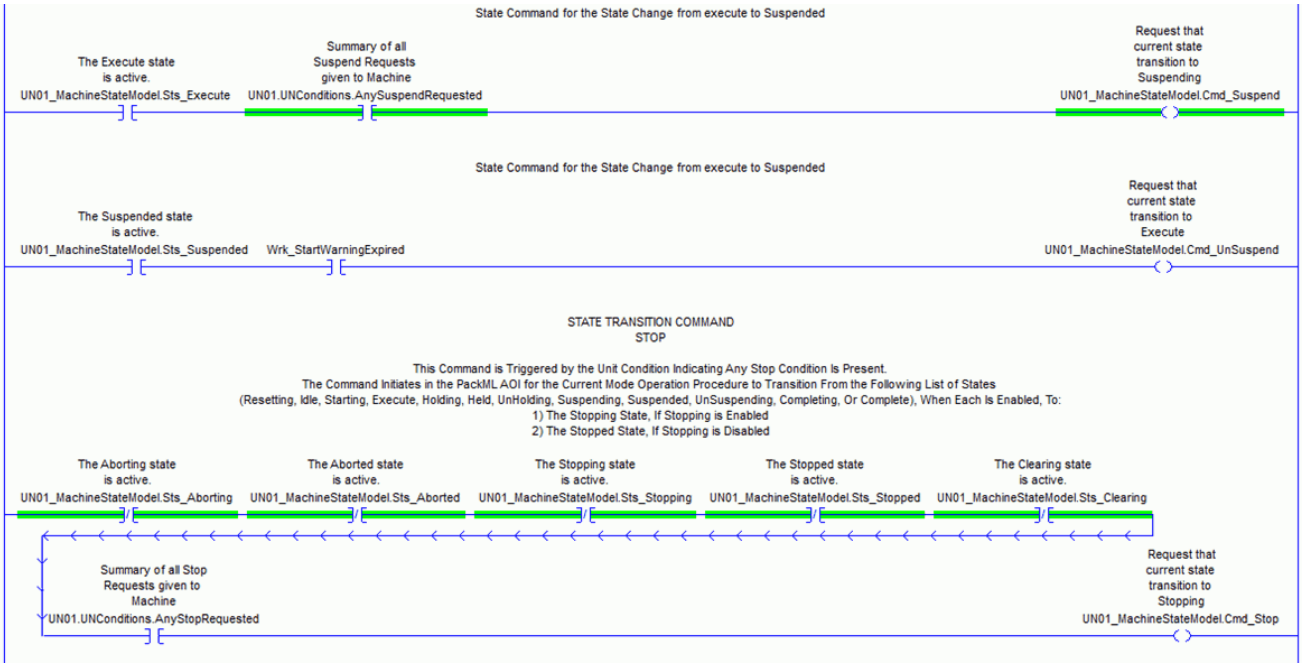
State Commands (Out_StateCommandBool - Boolean Format - or - Out_StateCommand - INT Format)

The equipment state command is used by the RAPID Line Control system to tell a machine to go from a running state to a suspended state, from a suspended state to a running state, or to stop from any state.

Below is example State Command Logic from programs that were constructed by using Power Programming V4.2.



The following example rungs show how the logic in the example State Command Logic on page 32 ties to the state model command interface of the machine.



Remaining Tags Required for RAPID Line Control Functions

All other tags that are required for line control functionality have been explained in previous sections.

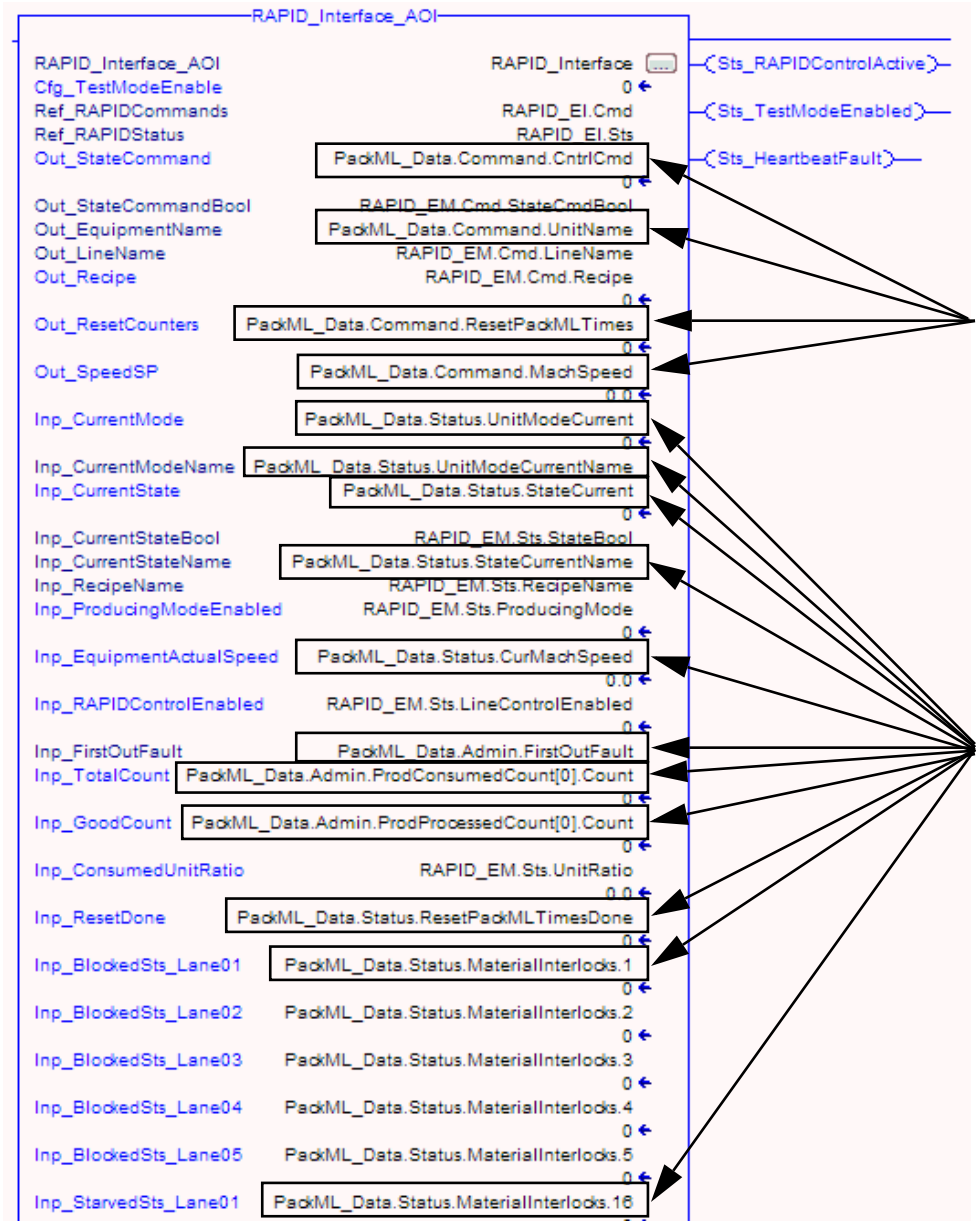
RAPID Equipment Interface with ISA-TR88.00.02 (PackML)

For equipment that has been programmed based on ISA-TR88.00.02 (PackML) there are two methods in which the PackML data can be mapped into the equipment interface:

- Packtags can be mapped to the RAPID_EM data structure as shown in Chapter 1 and Chapter 2.
- Packtags data elements can be mapped directly into the interface.

The RAPID application was designed around PackML terminology, state model, and data structures. Therefore, most data elements can be mapped directly into the interface without any additional logic. When using this method, the RAPID_EM data elements can be replaced directly with PackML tags. There are some data elements that RAPID requires that are not available from the PackML data structure. You must map some data as shown in Chapter 1 and Chapter 2.

The graphic below shows how PackML data elements (PackTags) can be directly mapped to the equipment interface Add-On Instruction without any additional mapping logic. Some RAPID data elements are not provided by the PackTags data structure so these tags must be mapped to other equipment data points as described in Chapter 1 and Chapter 2.



IMPORTANT: Any tag mappings on this Add-On Instruction that have a RAPID_EM tag must be mapped according to instructions in Chapter 2.

For a complete definition of all data mapping tags, see Appendix A.

Additional Resources

For more information on ISA-TR88.00.02 and PackML programming concepts, see the following:

- Industry Standards websites:
  - http://www.omac.org/
  - http://www.isa.org/
- Rockwell Automation websites (ISA-88 Modular Programming capabilities from Rockwell Automation):
  - http://www.rockwellautomation.com/rockwellautomation/solutions-services/oem/design-develop-deliver/modular-programming.page?
  - http://www.rockwellautomation.com/rockwellautomation/solutions-services/oem/power-programming.page?
  - https://rockwellautomation.custhelp.com/app/answers/detail/a_id/66060

IMPORTANT You must have a Rockwell Automation Technical Support Center account to access this link.

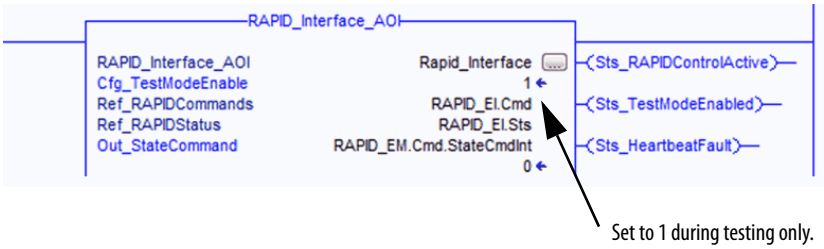
Notes:

Test the Application

The Equipment Interface testing tool is a FactoryTalk® View Studio ME application that you use to verify that the Equipment Interface Add-On Instruction is functioning properly in the equipment control program.

The FactoryTalk View Studio ME interface testing tool lets you verify that your machine data is mapped properly for use with the RAPID Line Integration™ System. Once your data is mapped, use this tool to verify that the data being sent to the RAPID system accurately reflects the actual status your equipment.

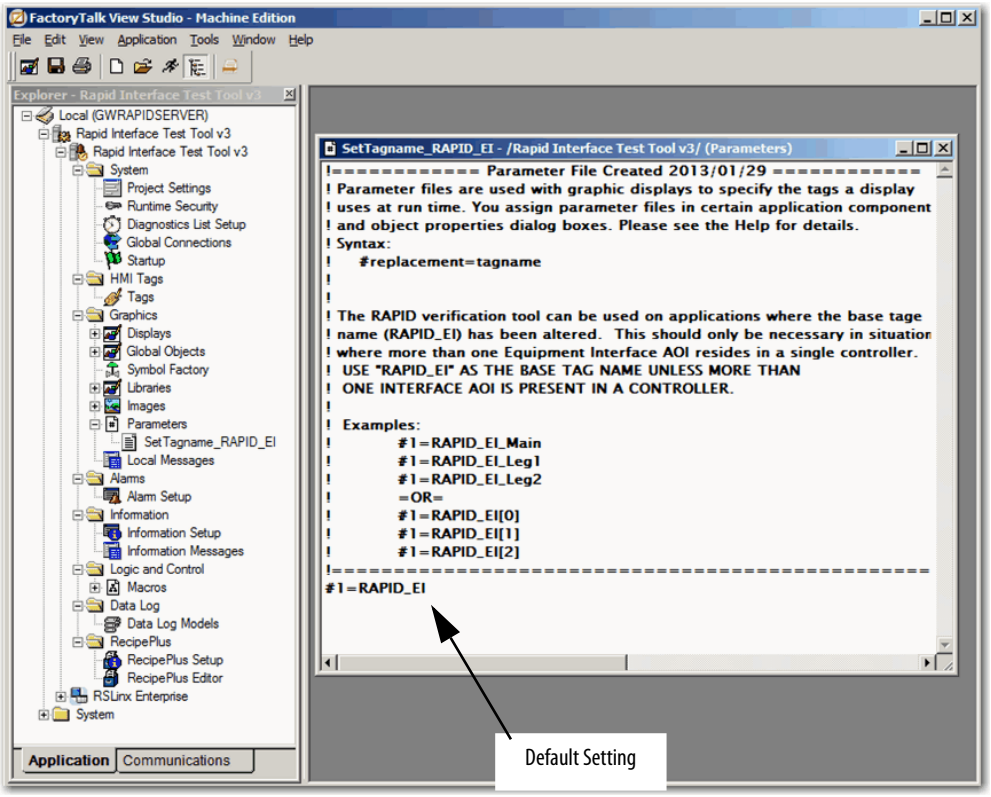
Start FactoryTalk View Studio ME and restore the RAPID_Interface_Test_Tool_v3.apa found in the Equipment Interface folder structure.



By default, the testing tool is set to connect to the following:

- Controller Topic Name: EQUIPMENT - Set in RSLinx® Enterprise DDE/OPC settings
- Controller Tag Name: RAPID_EI - Set in the View Studio project parameters

You must modify the settings that are shown in the following graphic if you are not using the default values.



When configured correctly, the application runs directly from FactoryTalk View Studio software. Click Application > Test Run Application to use the tool.

Testing Tool Instructional Videos

The RAPID Equipment Interface OEM Toolkit includes instructional videos that show how to use the contents of the toolkit, including how to test your application.

IMPORTANT The WebEx ARF player is required to play the recordings. After successful connection to the links below, you receive instructions about how to use the player.

You can access the Toolkit at the Product Compatibility and Download Center at http://rockwellautomation.com/support as an additional download with a valid serial number for FactoryTalk View SE, ME, or RSLogix 5000® software.

RAPID_EM Data Structure

This chapter provides definitions of the tags that are used with the RAPID_EM data structure.

RAPID Equipment Interface Tag List

Table 1 describes the RAPID_EM status tags.

Table 1 - Rapid Tag Descriptions

Tag Details				
Tag Name	**Description**	**Data Format**	**Requirement**	**Comment**
RAPID_EM.Sts.Mode	Current Mode of Equipment. 1 = Production Mode 2 = CIP (Cleaning) 3 = Change Over 4 = Maintenance/Manual 5 = Other	DINT	Required	Production =1, Non-Production > 1
RAPID_EM.Sts.ModeName	Current Mode of Equipment Name of Current Mode if it is not as defined in the target program.	String	Required if .Mode is not being used as standard	Optional: Name of current mode

Table 1 - Rapid Tag Descriptions

Tag Details				
Tag Name	**Description**	**Data Format**	**Requirement**	**Comment**
RAPID_EM.Sts.State	This tag presents the current equipment state that is represented by an integer value. This tag is typically used when an ISA-TR88.00.02 state model exists with values as defined. ISA-TR88.00.02 (PackML) defines integer values as: • 1= Clearing [Optional state] • 2= Stopped [Required state: Default state if not Running, Suspended, or Faulted] • 3= Starting [Optional state] • 4= Idle [Optional state] • 5= Suspended [Required state: External upstream/ downstream permissive is present] • 6= Execute [Required state: "Running"] • 7= Stopping [Optional state] • 8= Aborting [Optional state] • 9= Aborted [Required state: .FirstOutFault values must be provided when in this state] • 10= Holding [Optional state] • 11= Held [Optional state] • 12= UnHolding [Optional state] • 13= Suspending [Optional state] • 14= Unsuspending [Optional state] • 15= Resetting [Optional state] • 16= Completing [Optional state] • 17= Complete [Optional state]	DINT	Required Unless .StateBool is used	if not .StateBool
RAPID_EM.Sts.StateBool	This tag presents the current state of the equipment with a unique Boolean tag that represents each state. All states are mutually exclusive, as defined by ISA-TR88.00.02. NOTE: If .State is not zero, then this tag is ignored.	UDT_PML_States	Required if .State is not being used.	if not .State unused states must be set to 0!
RAPID_EM.Sts.StateName	Name of current state if not defined as above.	String	Optional	Optional: Name of current state if not defined as standard
RAPID_EM.Sts.RecipeName	Name of current recipe being used by equipment.	String	Optional	Optional: Name of current recipe in use of equipment
RAPID_EM.Sts.ProducingMode	If current mode must be monitored for performance, set to Boolean value=1 (True).	BOOL	Required	Producing Mode = 1 (BOOL) if you want to monitor the current mode for performance.
RAPID_EM.Sts.CurrentSpeed	Current rate of machine measured in discharged or processed parts per minute.	DINT	Required	Actual Production Rate in parts per minute

Table 1 - Rapid Tag Descriptions

Tag Details				
Tag Name	**Description**	**Data Format**	**Requirement**	**Comment**
RAPID_EM.Sts.LineControlEnabled	When RAPID Line Control function is being implemented, Set to Boolean value=1 (true) when the equipment can accept remote commands. (Start, Stop, Suspend, Unsuspend)	BOOL	Required	Indicated that the equipment allows commands from line control
RAPID_EM.Sts.FirstOutFaultEvent.ID	Reason for being in Fault/ Aborted state. The .Message string and/or ID value must be populated before turning on the trigger bit. The trigger stays on until the equipment is no longer faulted or stopped. The category number is assigned by the end user or RAPID system integrator. Categories are used to report fault events as being related to items like Materials, Utilities, Mechanical, Electrical, Safety. The equipment/machine programmer defines the .ID and .Message tags.	UDT_Event	Required	Fault Message ID (if not .Message used)
.Message				Fault Message Text (if not .ID used)
.Category				Example: Categories: 1: Material; 2: Utilities; 3: Electrical; 4: Mechanical
.Trigger				On for entire downtime event (BOOL)
RAPID_EM.Sts.TotalCount	Quantity of parts that are consumed by the equipment. Typically counted at the equipment infeed.	DINT	Required	Equipment Infeed Count
RAPID_EM.Sts.GoodCount	Quantity of parts that are processed by the equipment. Typically counted at the equipment discharge.	DINT	Required	Equipment discharge count
RAPID_EM.Sts.UnitRatio	Number of consumed parts that are contained in each processed part. (example: bottles per case)	Real	Required	Ratio of total to good (must fit to .TotalCount and .Good Count of Equipment)
RAPID_EM.Sts.ResetDone	Reserved for future use	-	-	Reserved for future use
RAPID_EM.Sts.Lane01Blocked	Primary Product Infeed is Blocked	BOOL	Required	At least one downstream blocked condition is required (BOOL)
RAPID_EM.Sts.Lane02Blocked	Use only if multiple discharge lanes exist.	BOOL	Optional	Optional if more downstream conditions available
RAPID_EM.Sts.Lane03Blocked	Use only if multiple discharge lanes exist.	BOOL	Optional	
RAPID_EM.Sts.Lane04Blocked	Use only if multiple discharge lanes exist.	BOOL	Optional	
RAPID_EM.Sts.Lane05Blocked	Use only if multiple discharge lanes exist.	BOOL	Optional	
RAPID_EM.Sts.Lane01Starved	Primary Product Infeed is Starved	BOOL	Optional	At least one upstream starved condition is required (BOOL)
RAPID_EM.Sts.Lane02Starved	Use only if multiple infeed lanes or consumed materials exist.	BOOL	Optional	At least one upstream starved condition is required (BOOL)
RAPID_EM.Sts.Lane03Starved	Use only if multiple infeed lanes or consumed materials exist.	BOOL	Optional	
RAPID_EM.Sts.Lane04Starved	Use only if multiple infeed lanes or consumed materials exist.	BOOL	Optional	
RAPID_EM.Sts.Lane05Starved	Use only if multiple infeed lanes or consumed materials exist.	BOOL	Optional	

Table 1 - Rapid Tag Descriptions

Tag Details				
Tag Name	**Description**	**Data Format**	**Requirement**	**Comment**
RAPID_EM.Cmd.StateCmdInt	Optional: Can be used instead of StateCmdBool if it is desired to process the command in an integer format. Typical commands that are used by the line control function are Suspend, Unsuspend, and Stop. Only implement Start if a remote start is allowed from an idle state. Command Definitions: • 1 = Reset [not used by RAPID] • 2 = Start [optional if remote start from idle is allowed] • 3 = Hold [not used by RAPID] • 4 = UnHold [not used by RAPID] • 5 = Suspend [required if line control is in use] • 6 = UnSuspend [required if line control is in use] • 7 = Stop [required if line control is in use] • 8 = Abort [not used by RAPID] • 9 = Clear [not used by RAPID]	DINT	Optional	if not .StateCmdBool
RAPID_EM.Cmd.StateCmdBool .Suspend .Unsuspend .Stop .Start (optional)	If RAPID Line Control function is being implemented and .StateCmdInt is not being used, these values must be available. Requires machine level programming to initiate machine state changes. Typical Commands that are used by the line control function are Suspend, Unsuspend, and Stop. Only implement Start if a remote start is allowed from an idle state. NOTE: StateCmdBool is only monitored if StateCmdInt is equal to zero.	UDT_PML_Commands	Required	if not .SateCmdInt
RAPID_EM.Cmd.Name	Can be used on local HMI	String	Optional	Optional to be used on HMI
RAPID_EM.Cmd.LineName	Can be used on local HMI	String	Optional	Optional to be used on HMI
RAPID_EM.Cmd.Recipe	Can be used to set the recipe for the current machine. Requires machine-level programming to implement.	DINT	Optional	Optional to set the recipe for the machine that is running (needs respective programming on the equipment)
RAPID_EM.Cmd.SpeedSetPoint	Reserved for future use	BOOL	-	Future use reserved
RAPID_EM.Cmd.ResetCounters	Reserved for future use	Real	-	Future use reserved

The following table describes the RAPID_EM command tags.

UDT_RAPID_EM_Cmd	Description	Data Format
Requirement: If **RAPID Line Control** functionality is being implemented and .StateCmdInt is not being used, these values must be available. Requires machine level programming to initiate machine state changes. Typical commands that are used by the line control function are Suspend, Unsuspend, and Stop. Only implement Start if a remote start is allowed from an idle state. NOTE: StateCmdBool is only monitored if StateCmdInt is equal to zero.	.StateCmdBool .Suspend .Unsuspend .Stop .Start (optional)	UDT_PML_Commands
Optional: Can be used instead of StateCmdBool if it is desired to process the command in an integer format. Typical commands that are used by the line control function are Suspend, Unsuspend, and Stop. Only implement Start if a remote start is allowed from an idle state. Command Definitions: 1 = Reset [not used by RAPID] **2 = Start [optional if remote start from idle is allowed]** 3 = Hold [not used by RAPID] 4 = UnHold [not used by RAPID] **5 = Suspend [required if using line control]** **6 = UnSuspend [required if using line control]** **7 = Stop [required if using line control]** 8 = Abort [not used by RAPID] 9 = Clear [not used by RAPID]	.StateCmdInt	DINT
Optional: Can be used on local HMI	.Name	STRING
Optional: Can be used on local HMI	.LineName	STRING
Optional: Can be used to set current running machine recipe. Requires machine level programming to implement.	.Recipe	DINT
Reserved for future use	.SpeedSetPoint	REAL
Reserved for future use	.ResetCounters	BOOL

Notes:

RAPID Equipment Interface Add-On Instruction Online Help File (AOI Inputs/Outputs)

This chapter describes the online help parameters that are used with the RAPID_Interface_AOI, version 3.0. You can use the Add-On Instruction in the following programming languages:

- Ladder Logic
- Function Block Diagrams
- Structured Text

The online help parameters are required for line control programming, line performance programming, or both.

The equipment program must control these parameters for RAPID to function as designed. We recommend that all parameters that are described in Chapter 1 and Chapter 2 are programmed.

The following table describes the RAPID_EM status tags.

Required	Parameter Name	Data Type	Usage
x-both	RAPID_Interface_AOI	RAPID_Interface_AOI	InOut
x-both	EnableIn	BOOL	Input
view only	EnableOut	BOOL	Output
x-both	Cfg_TestModeEnable	BOOL	Input
x-both	Ref_RAPIDCommands	UDT_RAPID_EI_Cmd	InOut
x-both	Ref_RAPIDStatus	UDT_RAPID_EI_Sts	InOut
x-control only	Out_StateCommand	DINT	Output
x-control only	Out_StateCommandBool	UDT_PML_Commands	InOut
optional	Out_EquipmentName	STRING	InOut
optional	Out_LineName	STRING	InOut
optional	Out_Recipe	DINT	Output
optional	Out_ResetCounters	BOOL	Output
optional	Out_SpeedSP	REAL	Output
x-both	Inp_CurrentMode	DINT	Input
optional	Inp_CurrentModeName	STRING	InOut
x-or below	Inp_CurrentState	DINT	Input
x-or above	Inp_CurrentStateBool	UDT_PML_States	InOut
optional	Inp_CurrentStateName	STRING	InOut
optional	Inp_RecipeName	STRING	InOut
x-both	Inp_ProducingModeEnabled	BOOL	Input
optional	Inp_EquipmentActualSpeed	REAL	Input

Required	Parameter Name	Data Type	Usage
x-control only	Inp_RAPIDControlEnabled	BOOL	Input
x-performance only	Inp_FirstOutFault	UDT_Event	InOut
x-performance only	Inp_TotalCount	DINT	Input
x-performance only	Inp_GoodCount	DINT	Input
x-performance only	Inp_ConsumedUnitRatio	REAL	Input
optional	Inp_ResetDone	BOOL	Input
x-both	Inp_BlockedSts_Lane01	BOOL	Input
optional	Inp_BlockedSts_Lane02	BOOL	Input
optional	Inp_BlockedSts_Lane03	BOOL	Input
optional	Inp_BlockedSts_Lane04	BOOL	Input
optional	Inp_BlockedSts_Lane05	BOOL	Input
x-both	Inp_StarvedSts_Lane01	BOOL	Input
optional	Inp_StarvedSts_Lane02	BOOL	Input
optional	Inp_StarvedSts_Lane03	BOOL	Input
optional	Inp_StarvedSts_Lane04	BOOL	Input
optional	Inp_StarvedSts_Lane05	BOOL	Input
view only	Sts_RAPIDControlActive	BOOL	Output
view only	Sts_TestModeEnabled	BOOL	Output
view only	Sts_HeartbeatFault	BOOL	Output

Memory Usage in Controller Program

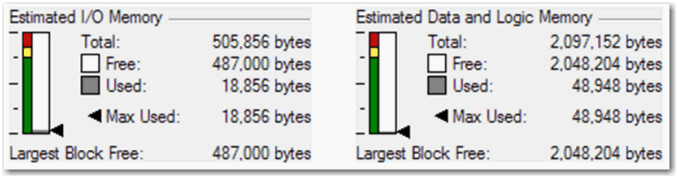
This appendix describes how importing the RAPID Equipment Interface to your Logix5000™ controller program impacts the available memory. The following table shows an example of how controller memory usage changes after the import.

Example Program: Available Free Memory Before Implementing the Equipment Interface	Impact of Importing the Equipment Interface Add-on Instruction and Associated UDTs		Impact of Adding the Equipment Interface Data Mapping Logic	
	Free Memory After Add-On Instructions and UDTs are Imported	Memory Used by Add-On Instructions and UDTs	Free Memory After Data Mapping Logic	Memory Used by Data Mapping Logic
2,048,204 bytes	2,027,256 bytes	20,948 bytes (21 KB) Fixed	2,023,492 bytes	3,764 bytes (4 KB) Can vary

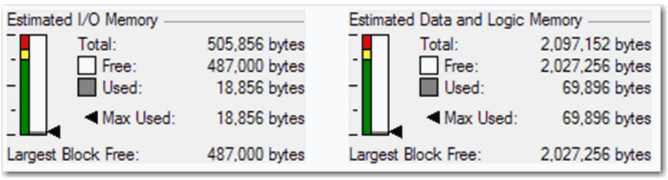
IMPORTANT As shown in the table above, the amount of memory use for the Add-On Instruction is fixed. However, the data mapping logic can vary depending if logic is needed to create the data and if control and information functions are being implemented.

The RAPID Equipment Interface memory usage affects the available Data and Logic Memory portion of controller memory. The I/O Memory portion of controller memory is unaffected by importing the RAPID Equipment Interface. The following graphic shows how the values on the Estimate tool dialog box change.

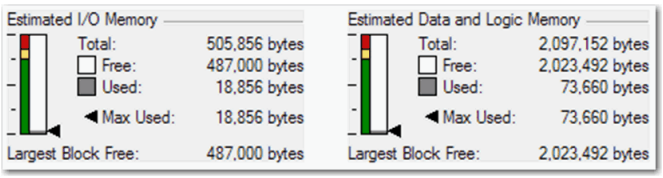
Before RAPID Equipment Interface is Added



Impact of Add-On Instructions and UDTs



Impact of Mapping Logic



Use of the Verification Utility

This section provides guidelines for how to use the RAPID Equipment Interface Verification Utility from the RAPID Equipment Interface OEM Toolkit.

Use this utility to check the following:

- That the interface Add-On Instruction and data structures are installed properly
- That the required data points have been populated to the interface
- That the interface is prepared for acceptance testing

The examples in this section refer to the Verification Utility of the Equipment Interface Toolkit version 4.x. The toolkit can be downloaded from the Rockwell Automation Compatibility and Download Center. The download is also described in Download the Toolkit on page 11.

Versions

Use RAPID Interface version 4, which is part of the RAPID Equipment Interface Toolkit v4.x. This version includes the interface Add-On Instructions, documentation, sample code, the Verification Utility, and instructional videos. The RAPID Equipment Interface Toolkit 4.x can be downloaded from the Rockwell Automation website.

Compatibility

Version 4.x of the RAPID Equipment interface is intended for use with v4.x of the RAPID Line Integration™ Solution supervisory controller.

Version 4.x of the RAPID Equipment Interface can be used with supervisory controllers v3.5 or later by performing one of the two methods:

- Program an OTE instruction to disable the interface data map diagnostics unconditionally. Tag name = EI_ErrorDetectionEnable.
- Install v3.5 Patch006, which provides support to allow the interface data map diagnostics. This patch entails updating the RAPID Root Cause Add-On Instruction, which requires a download to the supervisory controller.

Set Up the Verification Utility

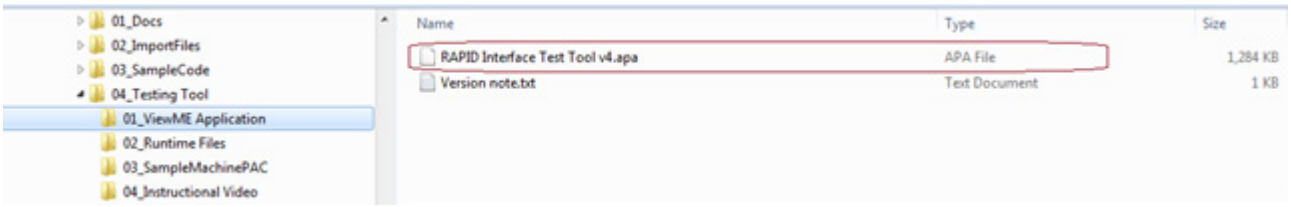
1. Copy the files onto a laptop that has FactoryTalk® View Studio (v8.1) installed.

   TIP The folder 04_Testing Tool consists of files, instructions, and a video that explains the use of the utility. View the instructional videos.

2. Open the RAPID Equipment Interface Toolkit 4.x -> 04_Testing Tool -> 01_ViewME Application folder.
3. To restore the RAPID Interface Verification Utility v4.apa file, double-click the .apa file for the RAPID Interface Test Tool.

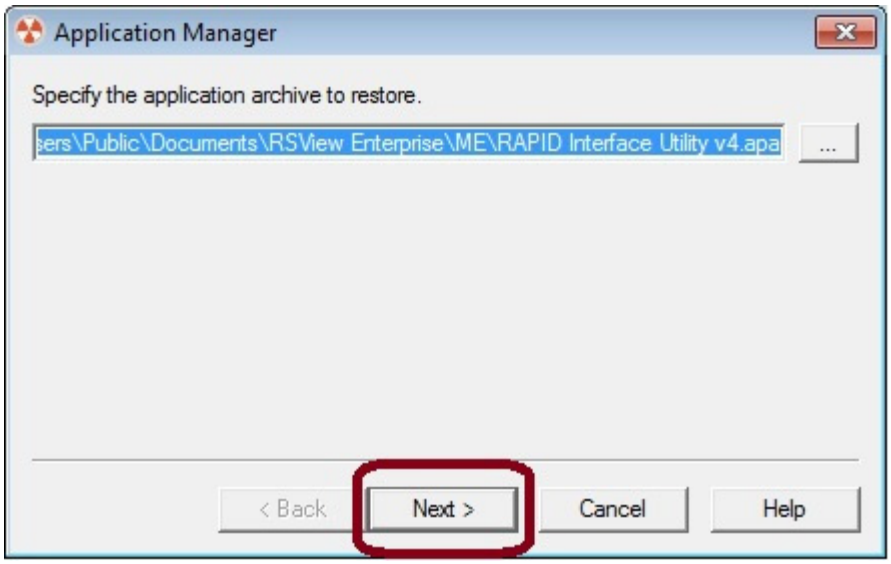
   The FactoryTalk View Application Manager opens.

Figure 6 - Select the RAPID Interface Test Tool



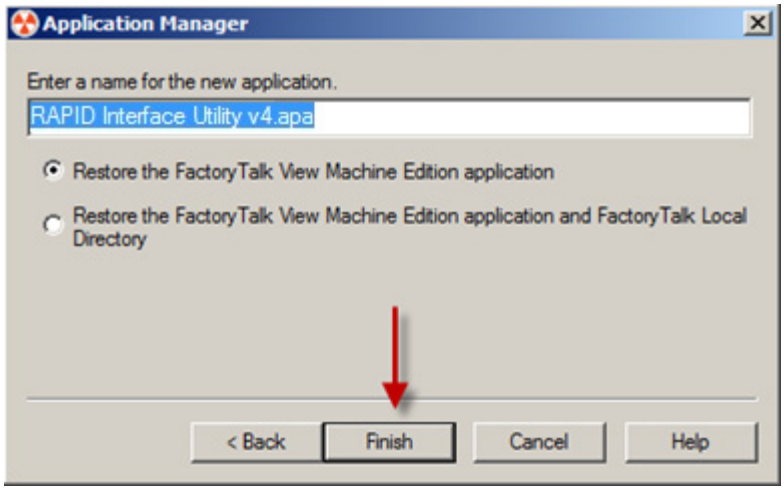
4. Click Next.

Figure 7 - FactoryTalk View Application Manager



5. Give the application the desired name (or use the default name) and click Finish.

Figure 8 - FactoryTalk View Application Manager

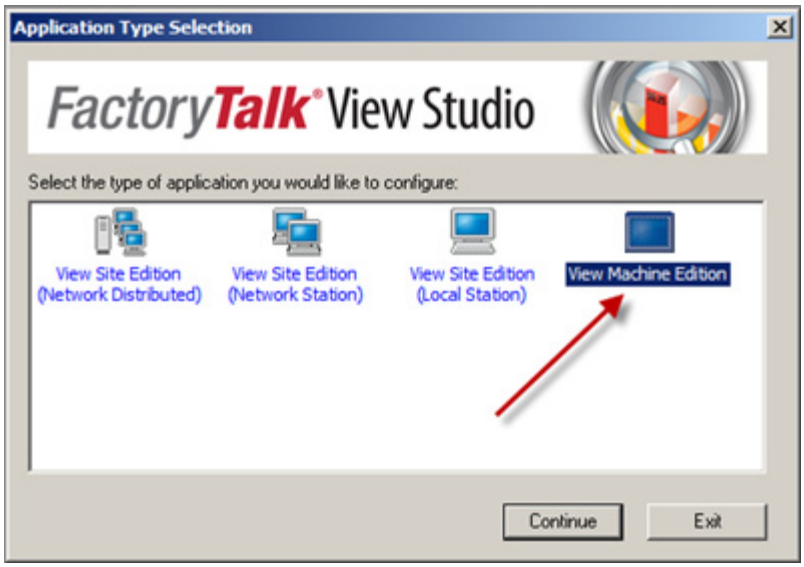


The application is now available in the FactoryTalk View Studio environment.

6. Open FactoryTalk View Studio and select View Machine Edition.

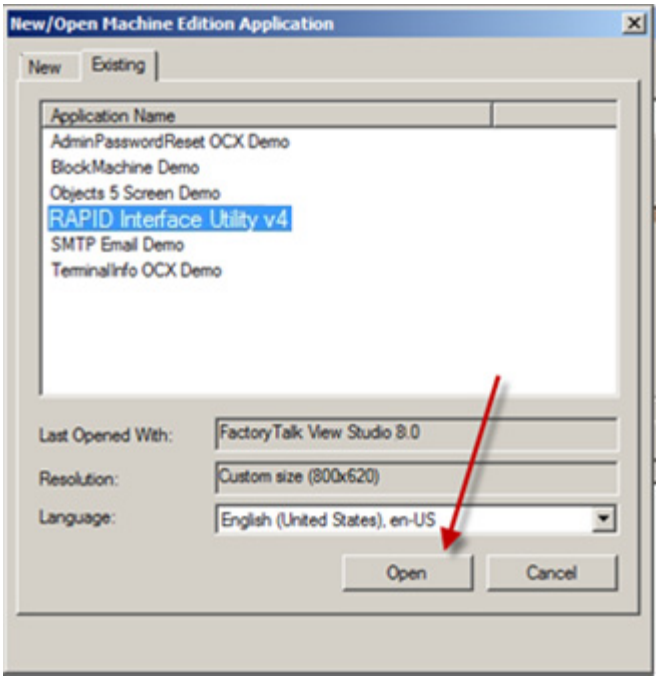
7. Click Continue.

Figure 9 - FactoryTalk View Studio Start-up Screen



8. Choose your application name and click Open.

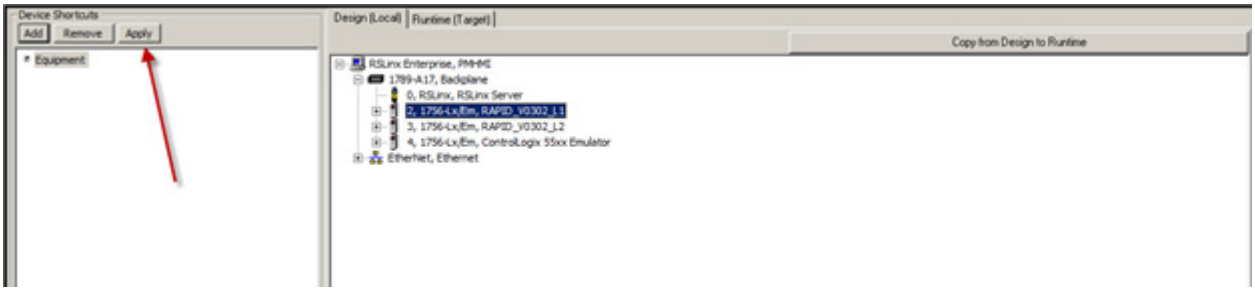
Figure 10 - Select Projects in FactoryTalk View Studio



9. Open in the FactoryTalk View Studio environment and then open the RSLinx® Enterprise Communication Setup.
10. Point the Device Shortcut (Equipment) to your controller.

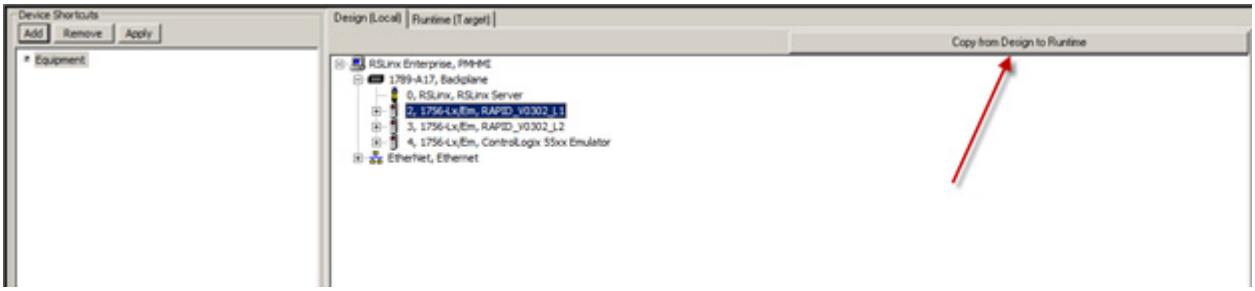
    In the example, the topic points to a controller in slot 2 on the virtual backplane.
11. Click Apply.

Figure 11 - Apply Changes in the RSLinx Enterprise Setup Screen



12. To make sure that the runtimes you create also point to this controller, click Copy from Design to Runtime.

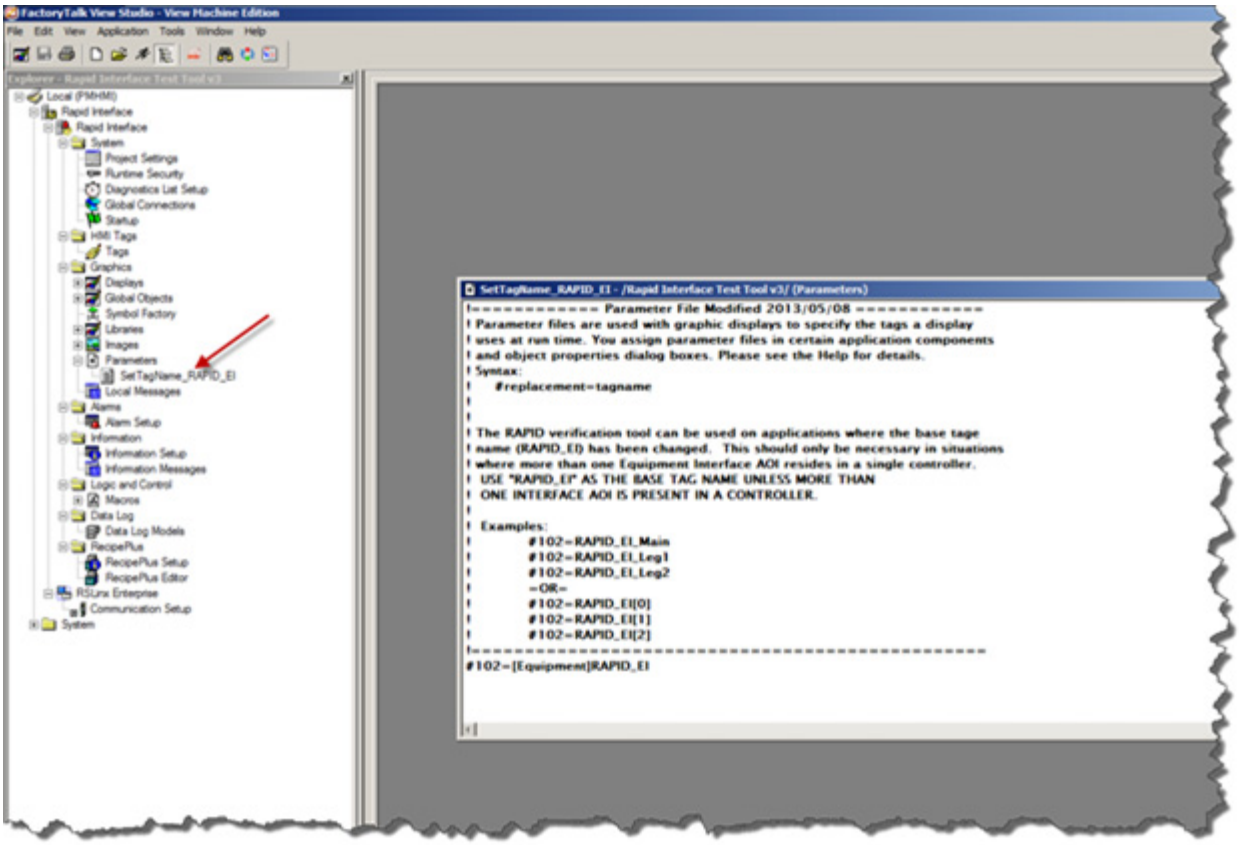
Figure 12 - Copy From Design to Runtime



13. Click Okay to save the changes.

14. Open the parameter file Set TagName_RAPID_EI from the Parameters folder.

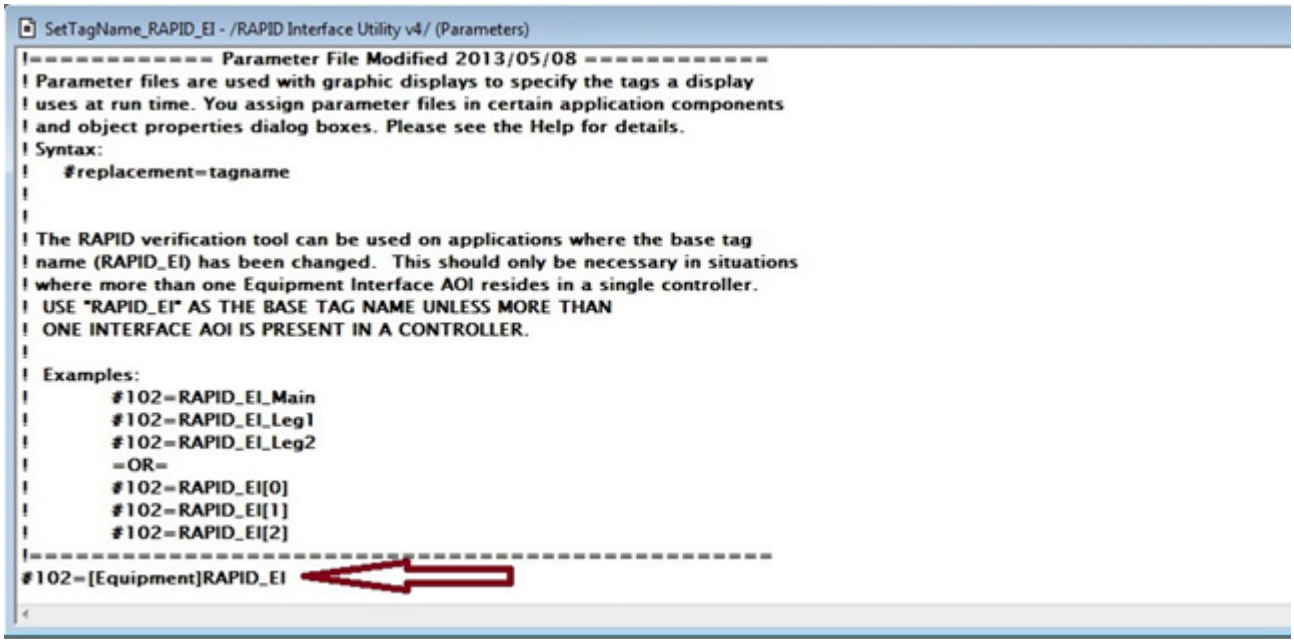
Figure 13 - Open the TagName_RAPID_EI Parameter File



In this parameter file, the name of the RAPID Equipment Interface Tag is specified. If you have just one RAPID Interface on your controller, the predefined name RAPID_EI is correct. If you use multiple RAPID Interface on one controller or you have given the Interface Tag another name, specify the name here. The displays of the test tool work with the Tag that is specified here for the parameter #102. Verify that the Tagname corresponds to the RAPID Interface that you want to check.

In the example, several RAPID Interfaces are saved on one controller. They are stored in an array with the name RAPID_EI. To test one of the interfaces, the parameter must be set to the instance of the array that is being monitored (in this example RAPID_EI[1]).

Figure 14 - Set TagName_RAPID_EI Parameter File to the RAPID Interface Add-On Instruction Tagname



15. Save and close the parameter file.

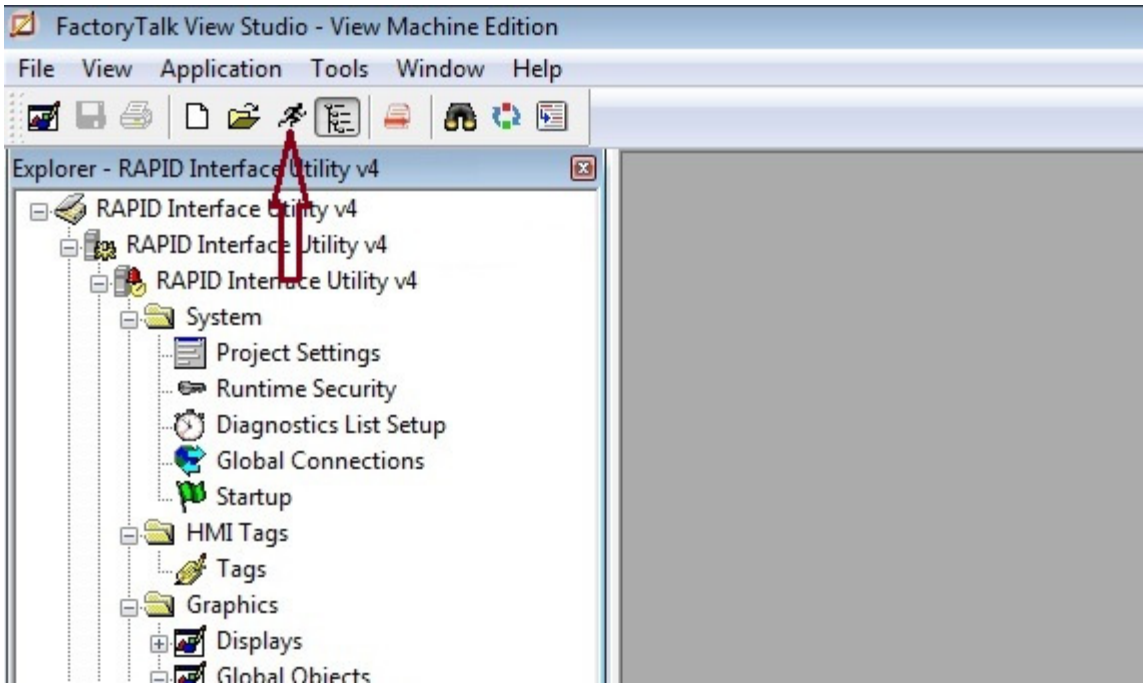
Use the Test Tool

You can use the test tool in one of two ways:

- Use the test application function of the FactoryTalk View Studio environment.
- Create a runtime and run the test application on an FactoryTalk View ME client such as a laptop or PanelView™ Plus.

To use the test tool from FactoryTalk View Studio, click Test Application or select Test Application from the Application Menu.

Figure 15 - Test Application Button in FactoryTalk View Studio

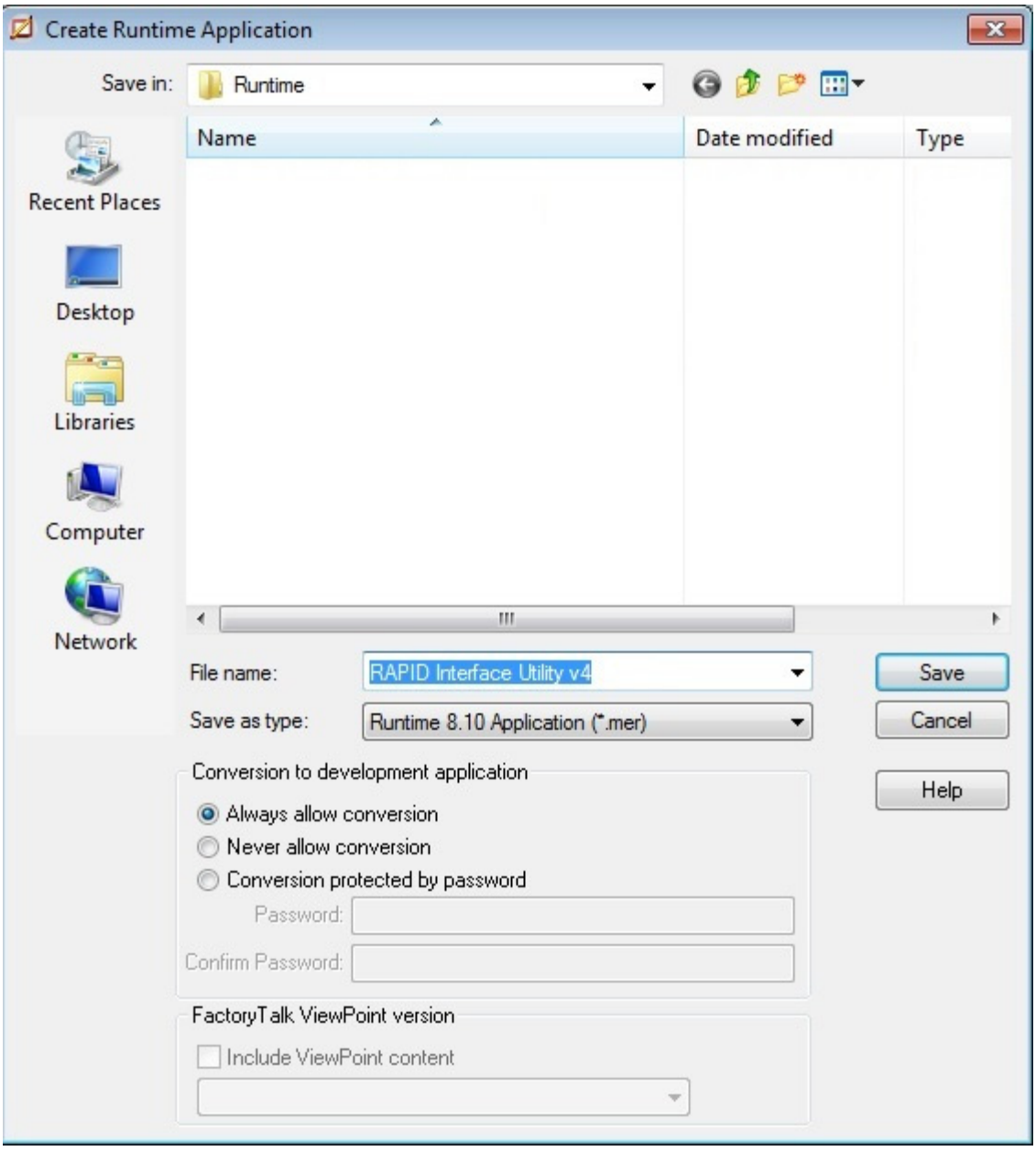


To use the test tool from an FactoryTalk View ME client:

1. Select Create Runtime Application from the Application menu.
2. Type the File Name and the storage location. Click Save.

   You can use the .mer file on a laptop computer or transfer it by using the Transfer Utility. For example, it can be transferred to a PanelView Plus terminal.

Figure 16 - Create Runtime Application Dialogue

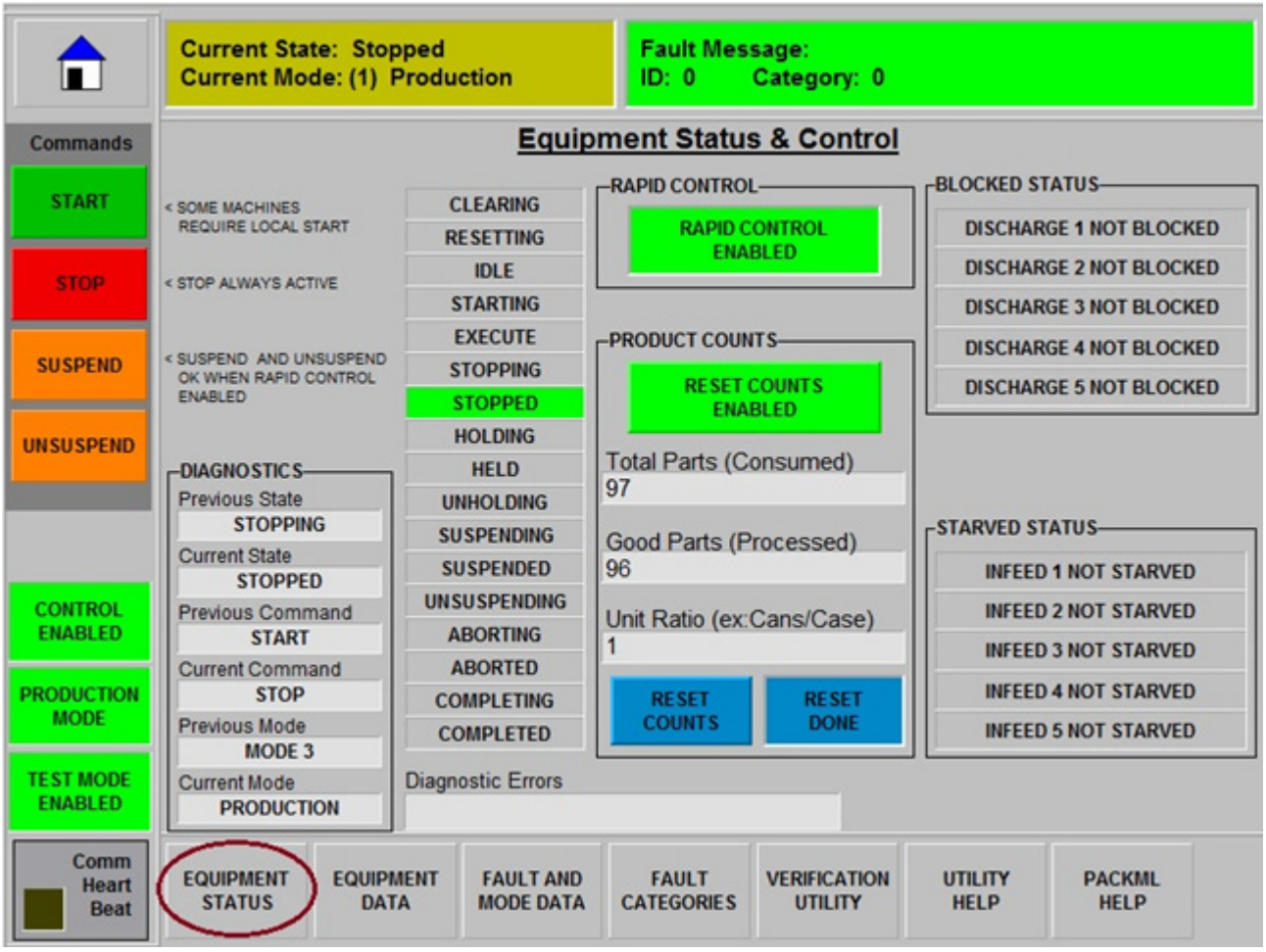


Start and Overview

When starting the utility from the Test Application button or as a client on your laptop or a PanelView Plus Panel, you access the Home screen. This screen contains information about the possible risks of using the Verification Utility on your machine. You must accept the terms and conditions of use to be able to continue with the use of the utility.

The next display that is shown is the Equipment Status Display. This display can be used to verify the interface manually and check the availability of required data during acceptance testing.

Figure 17 - Equipment Status Display



Diagnostics were added to the RAPID Equipment Interface Add-On Instruction to capture errors that can occur in the interface. A Diagnostic Error window on the Equipment Status Display shows any interface errors along with any machine faults. Interface errors include:

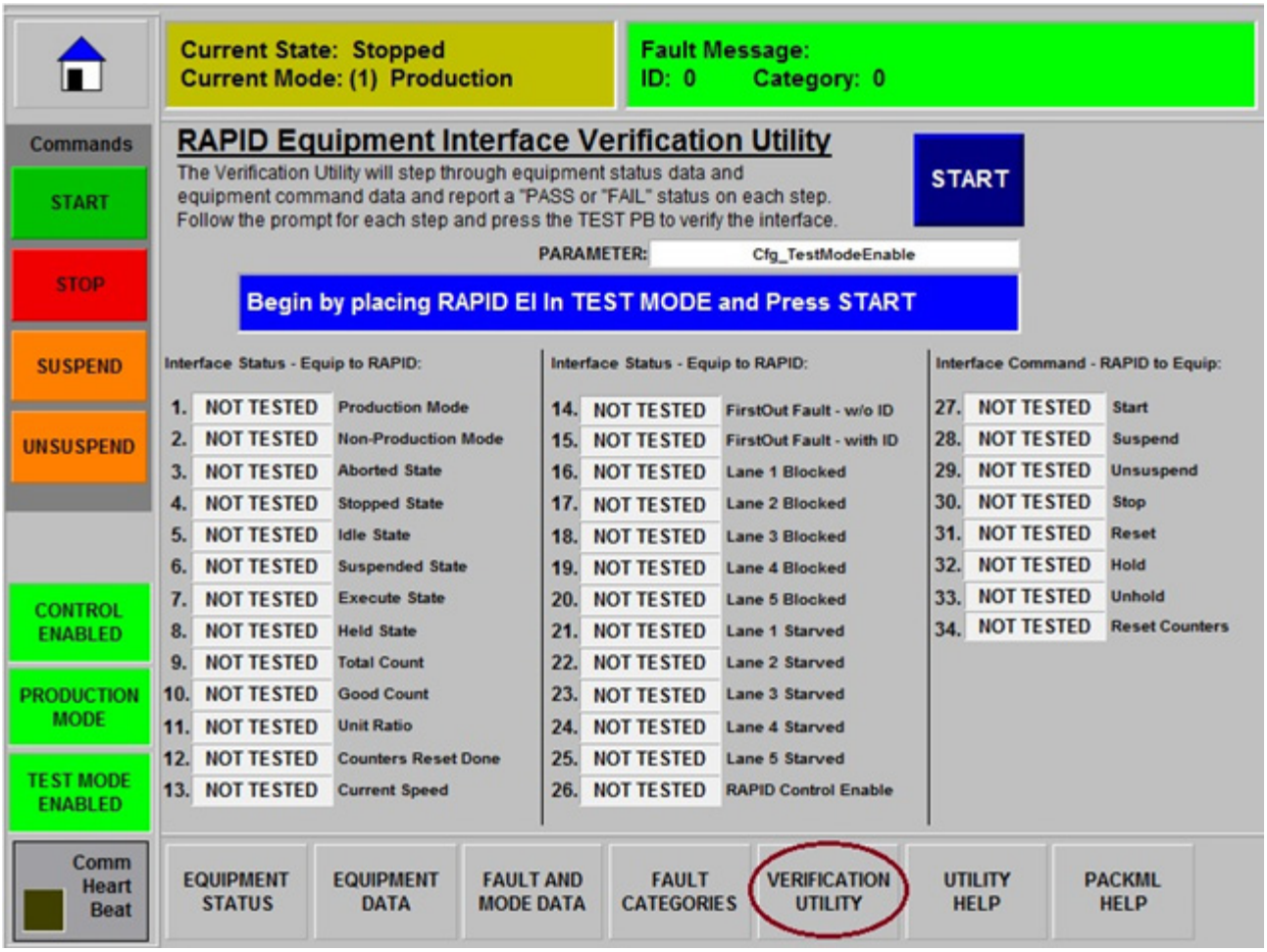
- Illegal Mode value (Mode range must be from 1...31)
- Illegal State value (must be legal State bit or value from 1...17)
- Fault Trigger when equipment is Executing
- No Fault Trigger when equipment is Aborted
- Fault Trigger with no Fault Message or ID
- Fault Trigger with no Fault Category
- RAPID Suspended with no Blocked or Starved conditions

- Reset Counter command timeout
- Unit Ratio invalid (Unit Ratio must be greater than or equal to zero)
- Part Counting error (Total and Good count increment when equipment is Executing)

The Verification Utility display enables automatic testing of equipment status data and equipment command data for one step at a time. Each step, when executed, reports a PASS or FAIL status.

TIP The verification does not start unless the tag Cfg_TestModeEnable of the Add-On Instruction Interface is set to 1, which places the interface in test mode.

Figure 18 - Verification Utility Display



The other interactive displays (Equipment Data and Fault Categories) are not accessing the data of the RAPID interface on the target machine (just the current speed). They are filled by the supervisory system and are briefly explained in a later section. They can be used to write values to the controller and show that this function is working as well.

The static help displays (Utility Help and PACKML Help) give you information about the utility itself and the OMAC PackML states and their transitions (see also Appendix B, RAPID Equipment Interface Add-On Instruction Online Help File (AOI Inputs/Outputs) on page 45).

Figure 19 - Utility Help Display

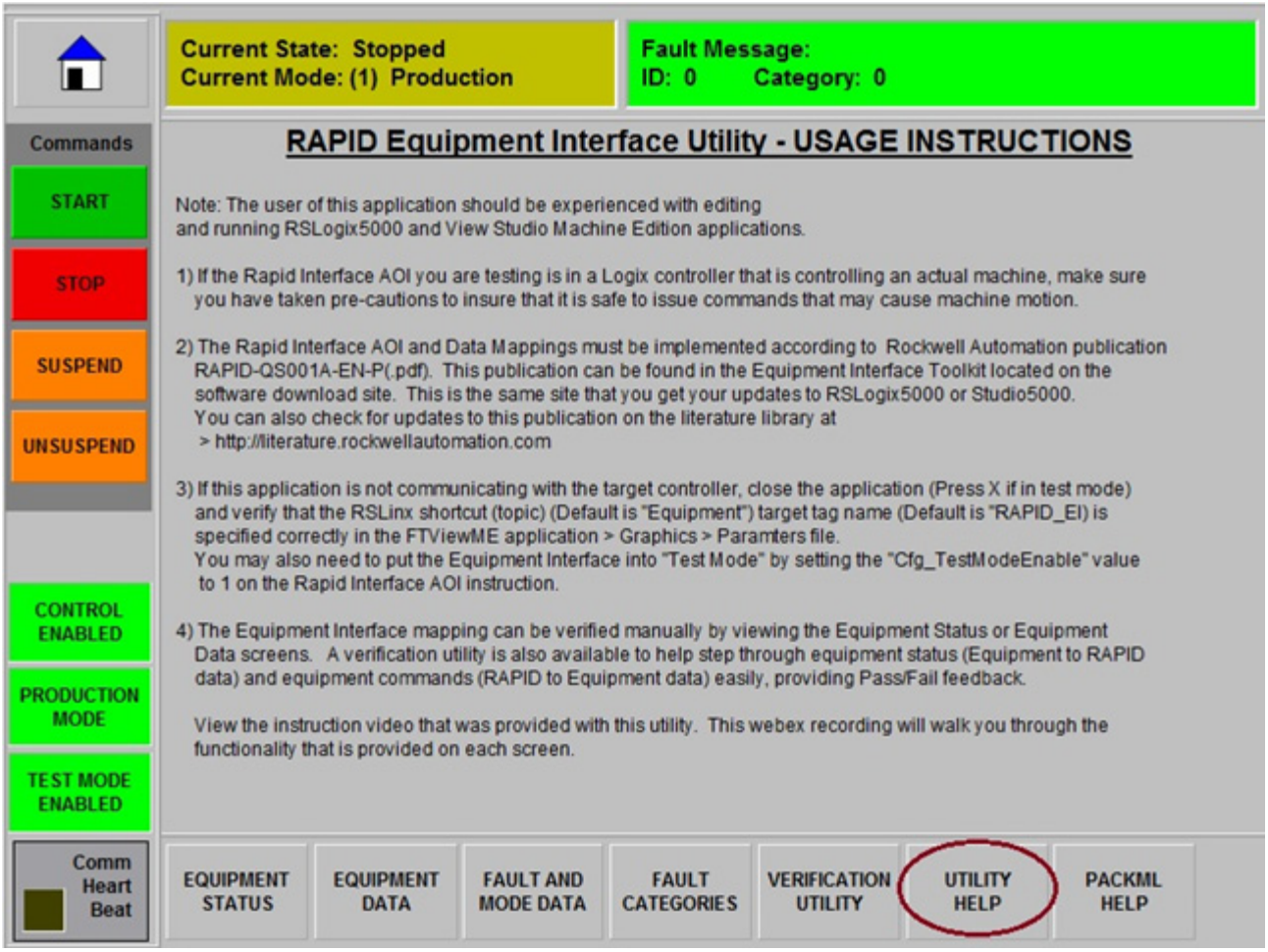
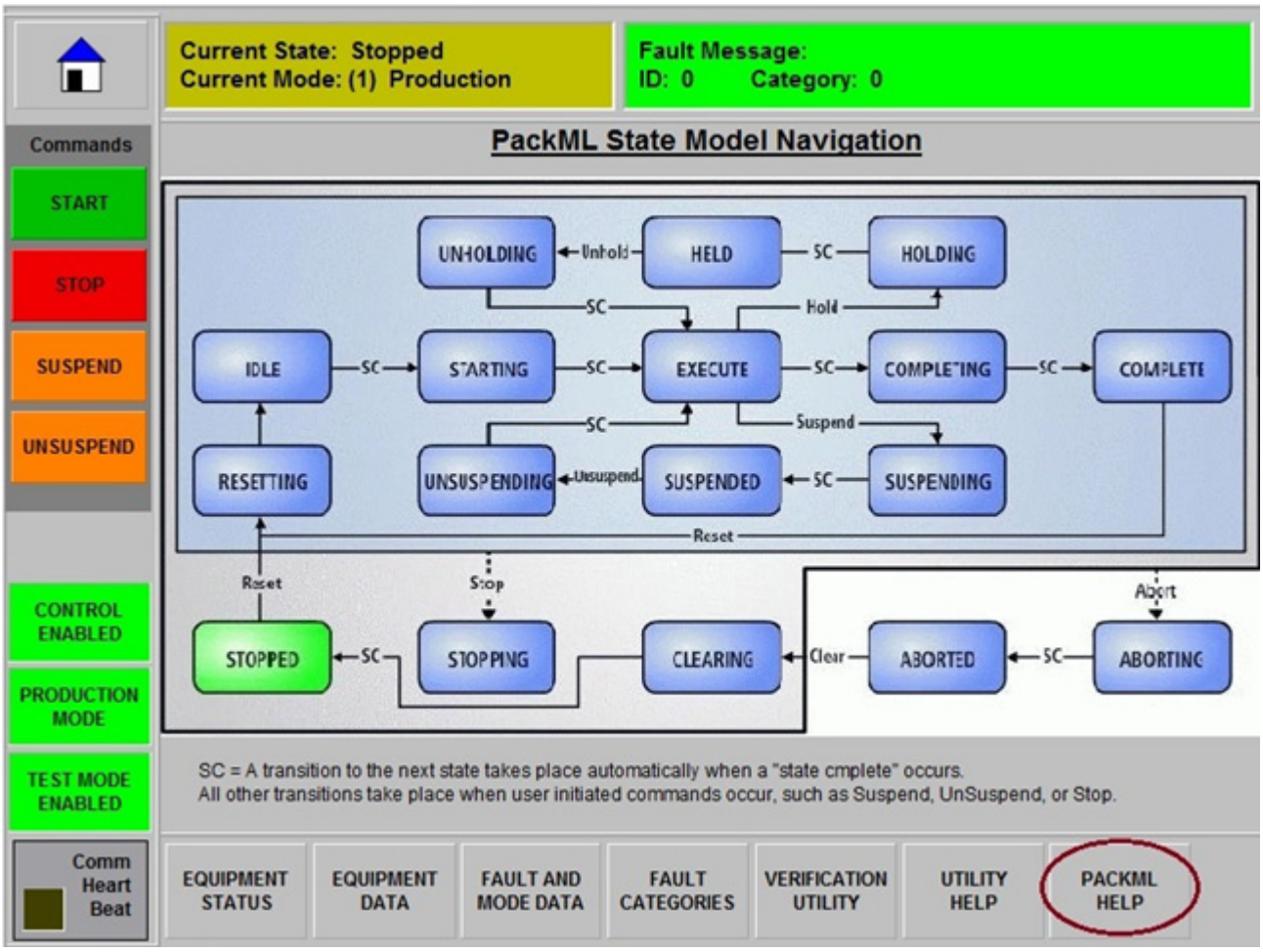
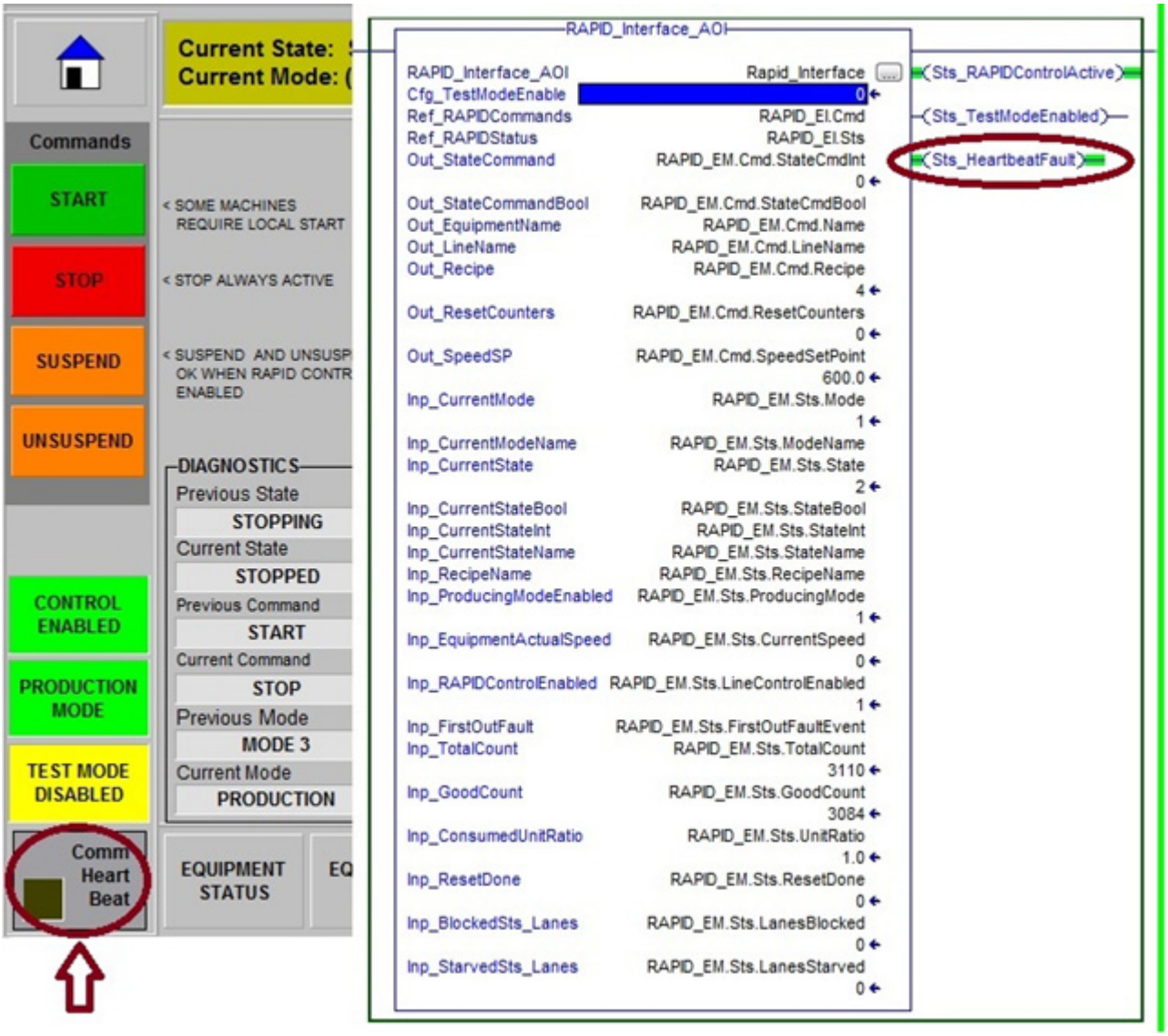


Figure 20 - PackML Help Display



If you use the Verification Utility as a client or in Test Application Mode in the development environment without the RAPID Server, the test tool Comm Heart Beat stays dark. When you look to your RAPID Interface Add-On Instruction on your controller, the Add-On Instruction shows a heartbeat fault.

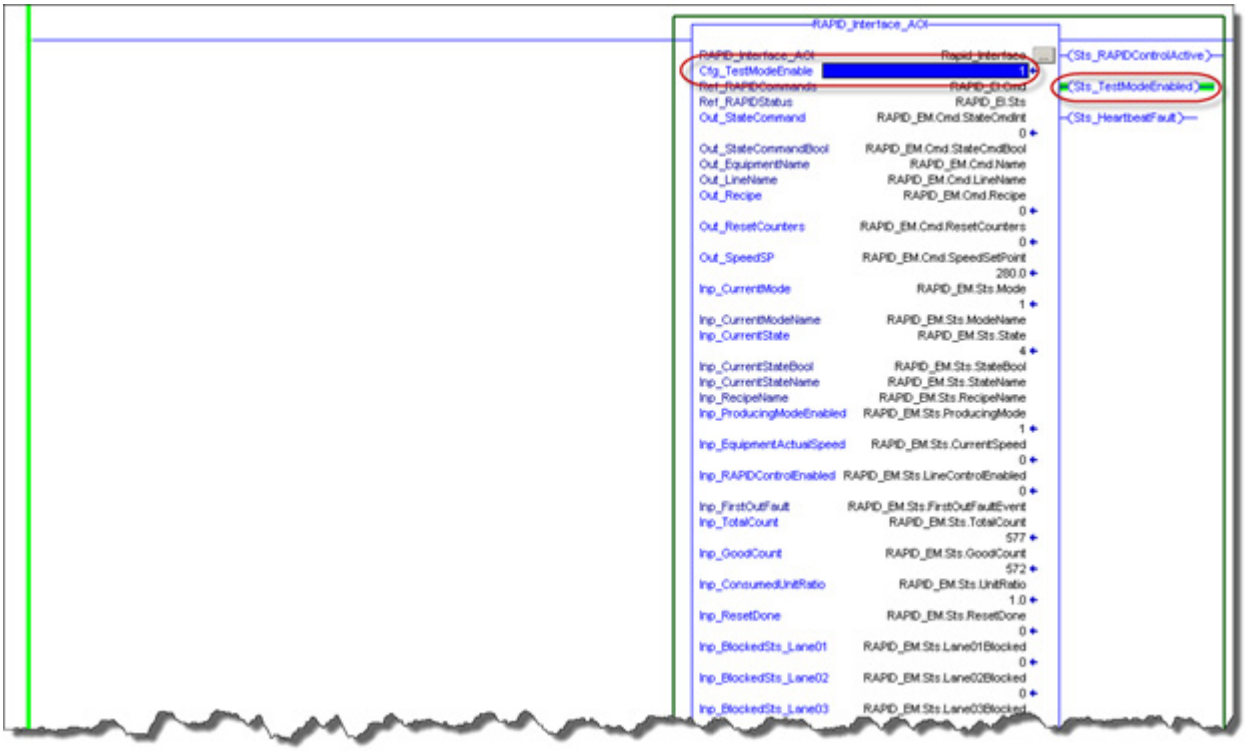
Figure 21 - Heartbeat Fault Due to Missing Supervisory System



This fault occurs because the Interface Add-On Instruction has no connection to the supervisory system. Due to the missing communication to the system, the RAPID Add-On Instruction is issuing a Stop command. In this example, the fault causes the machine to change its state to the Stopped State (see Figure 21).

To use the Verification Utility that is connected to the equipment controller, set Cfg_TestModeEnable on the Add-On Instruction Interface to 1. The Sts_HeartbeatFault becomes inactive and the Sts_TestModeEnabled becomes set. The value can be changed directly on the Add-On Instruction faceplate.

Figure 22 - Setting the RAPID Interface Add-On Instruction in Test Mode



Enable the test mode normally to use the Verification Utility without the RAPID server and without the logic addressing the heartbeat fault.

TIP Each download disables the test mode by setting the Cfg_TestModeEnable to 0. Each time that you perform similar tests that require an environment without the RAPID Server, change the Tag value to 1.

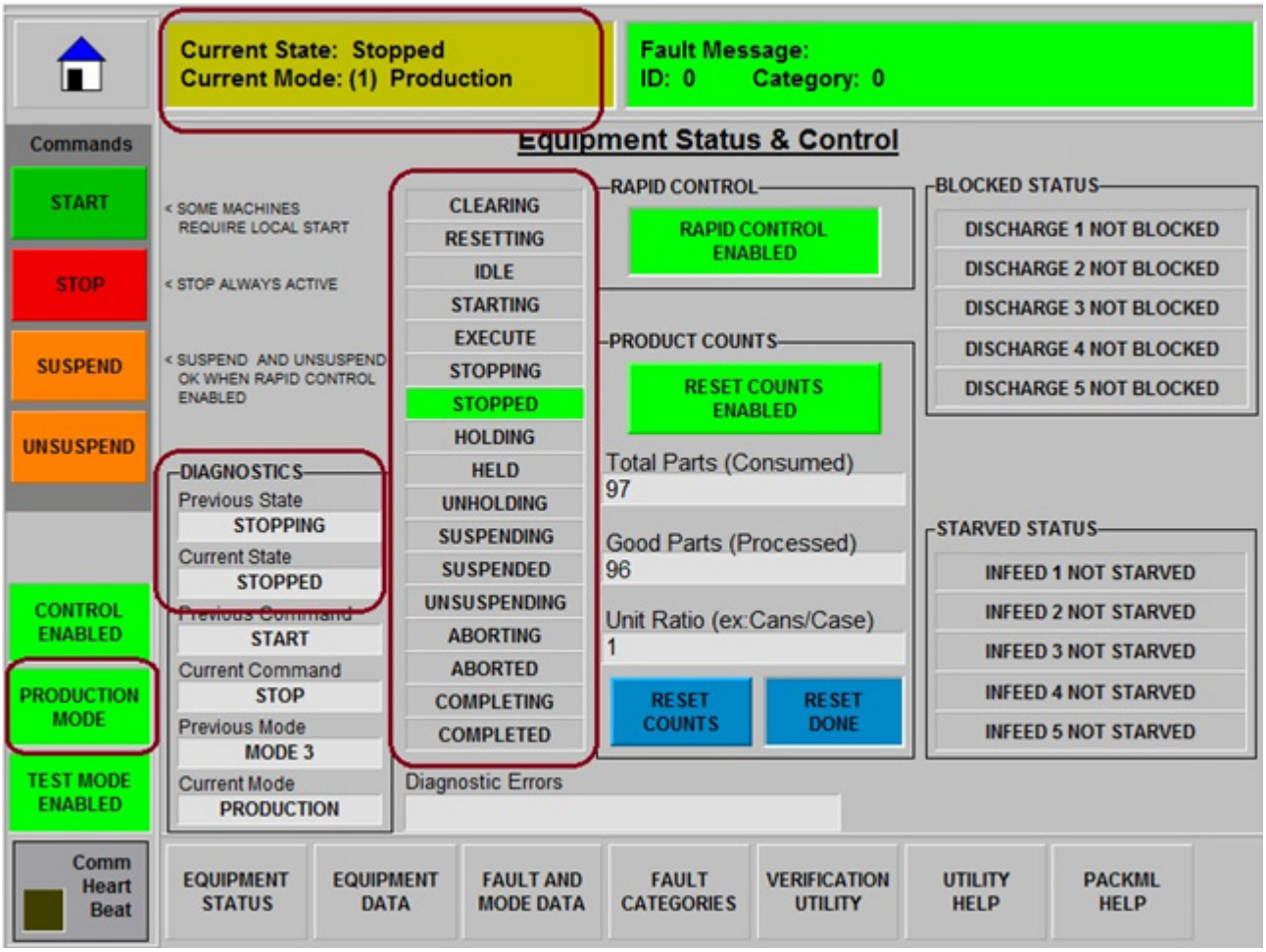
Beginning on page 64, you can learn how to manually verify the RAPID Equipment Interface by using the Equipment Status Display. Results can be recorded manually using the example templates that are located in appendices of this document.

Beginning on page 75 shows how to step automatically through and report the verification of the RAPID Equipment Interface by using the Verification Utility display.

Test the Interface Response

In the marked sections on the display Equipment Status, the current mode and current state of the machine can be supervised.

Figure 23 - Mode and State Indicators on the Equipment Status Display



Check the available states for the different modes.

A change from Production to the Manual mode (Mode 3 in this example) can be supervised as shown in Figure 24 and Figure 25. The programming must set all modes that do not belong to the key performance indicator (KPI)-relevant production time for RAPID_EM.Sts.ProducingMode to 0. Normally all modes >1 if OMAC PackML tags/modes are used.

Figure 24 - Mode Changed to Manual (3)

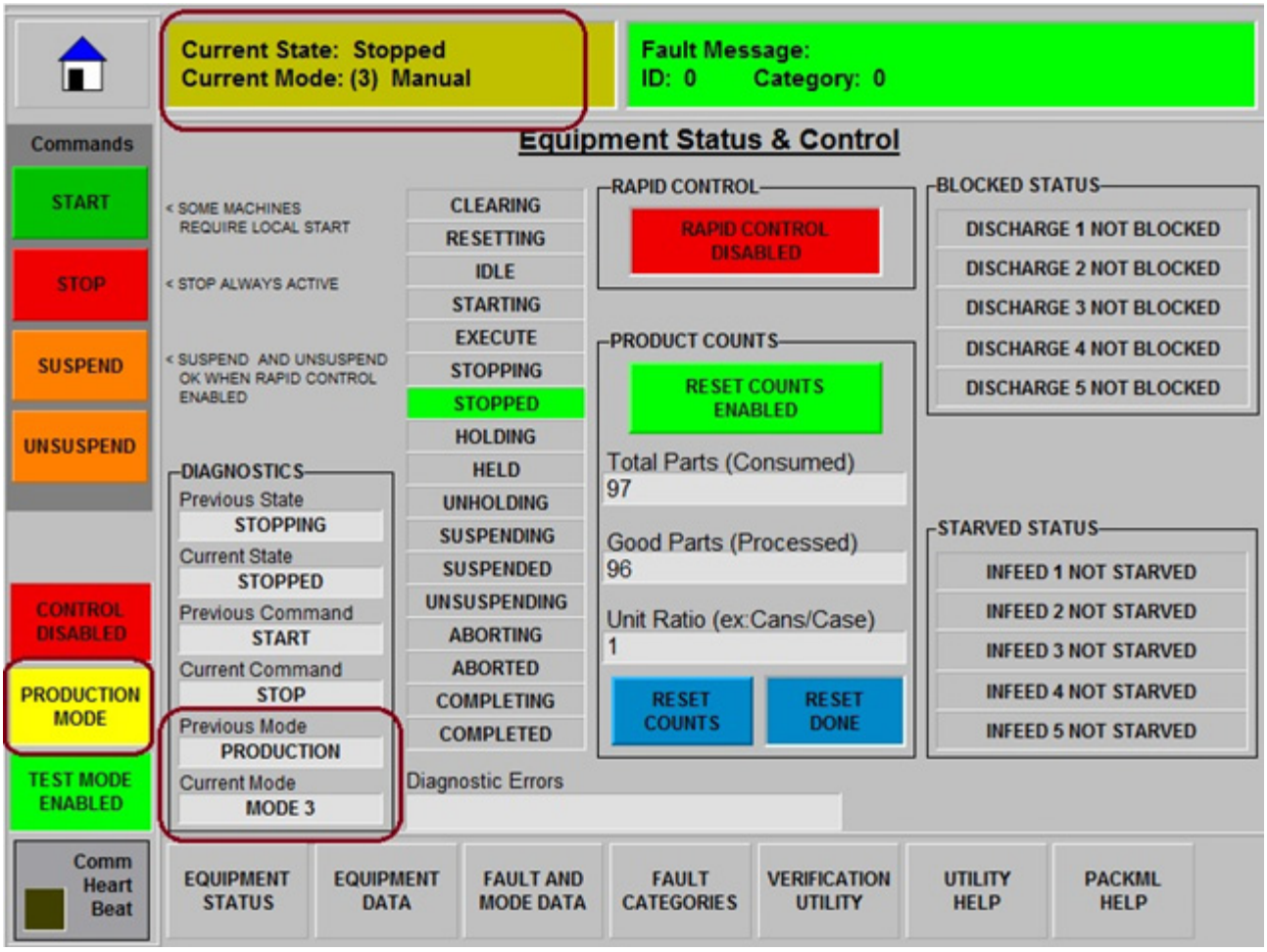


Figure 25 - Mode Information Displayed on Fault and Data Display

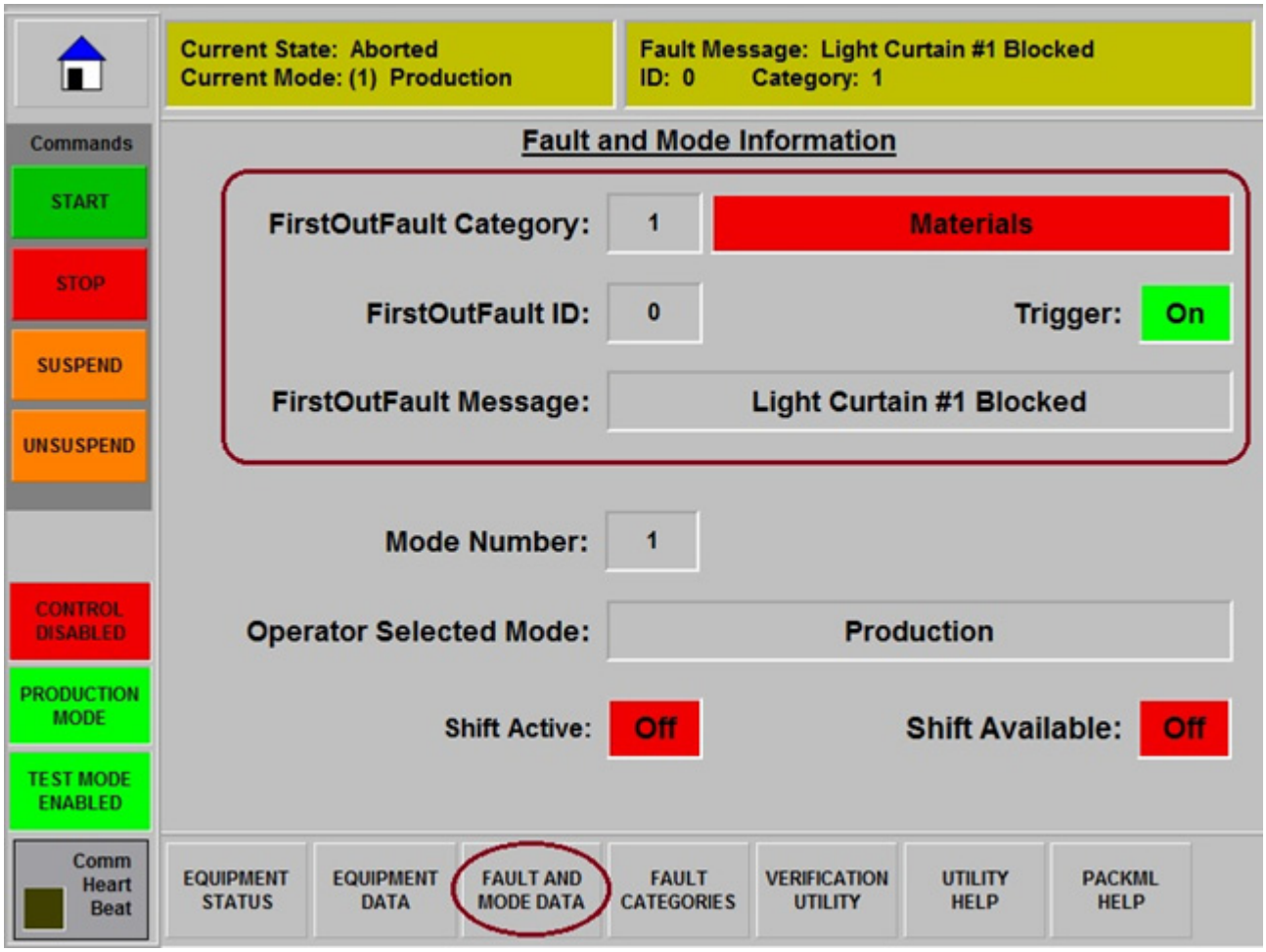
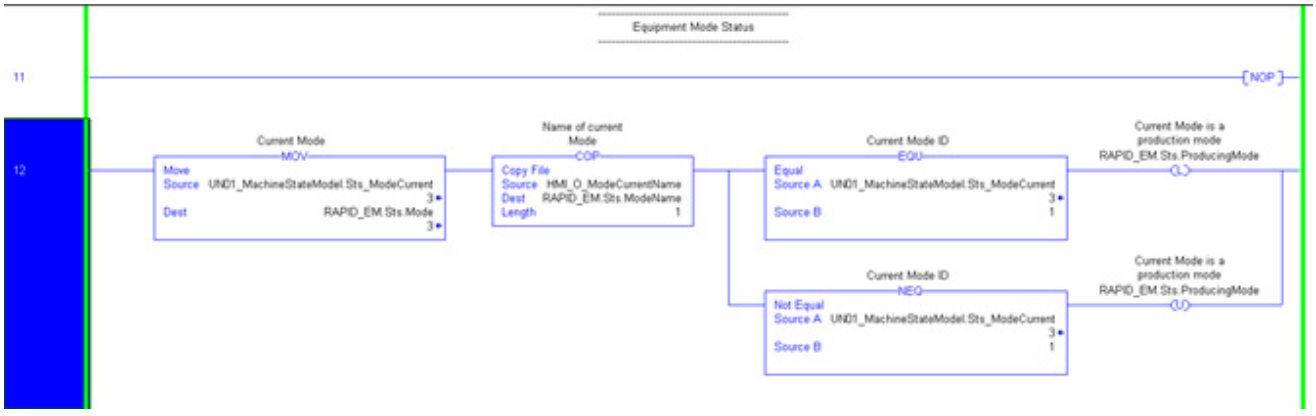


Figure 26 is an example of a corresponding logic (which can be found in the Machine Sample Code of the RAPID Toolkit in the CM98_RAPID_Interface routine). The marked indicator of the test tool supervises whether the Producing Mode is active or not.

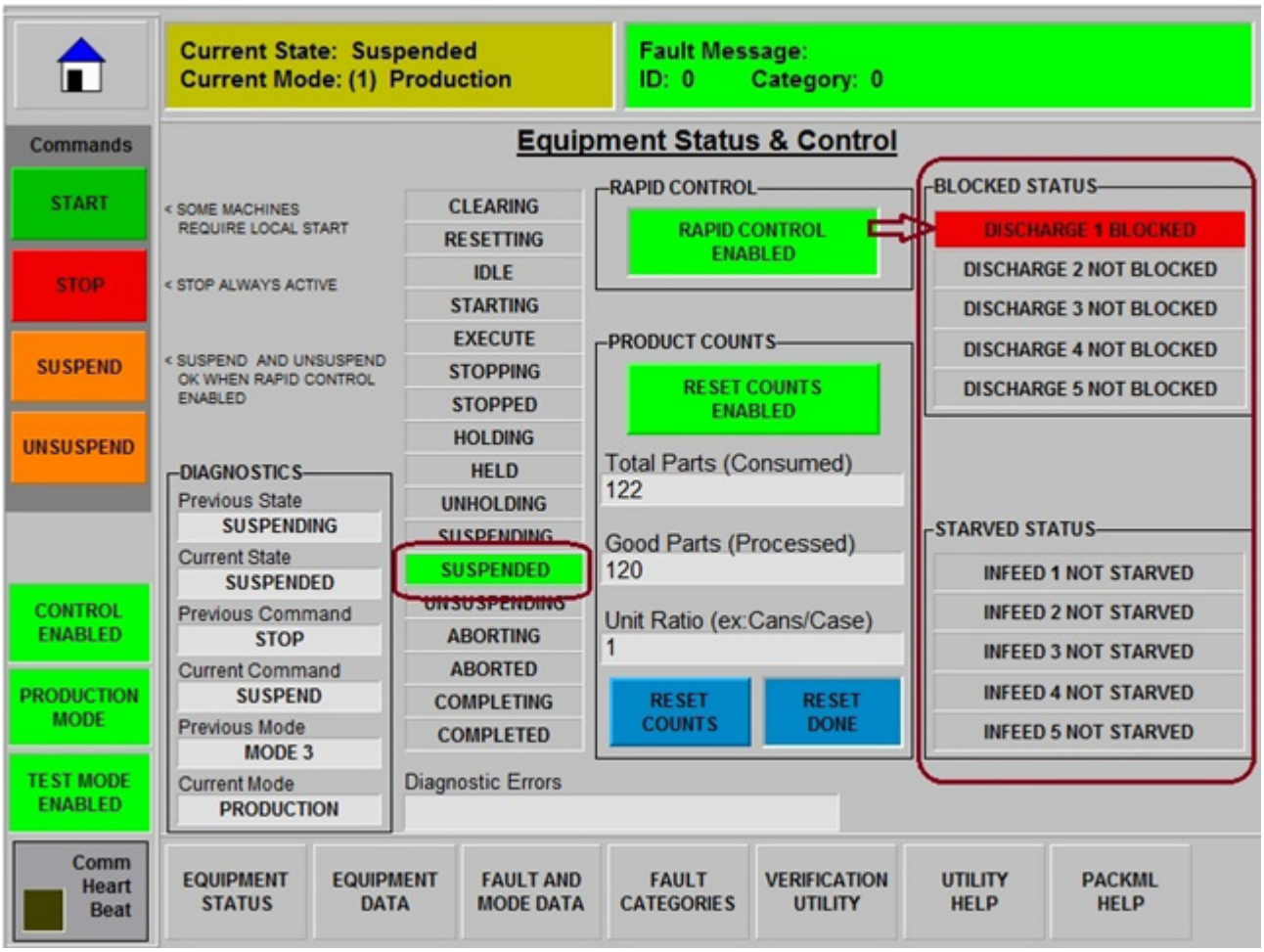
Figure 26 - Example Logic to Indicate Production Relevant Modes and Non-production Modes



Besides the OMAC PackML States, RAPID needs additional information when a machine is suspended due to up- or downstream reasons to be able to do a root-cause analysis.

The display extension on the right side gives additional information about the suspended state in starvation or blockage situations.

Figure 27 - Equipment Status Display Mode and State Indicators



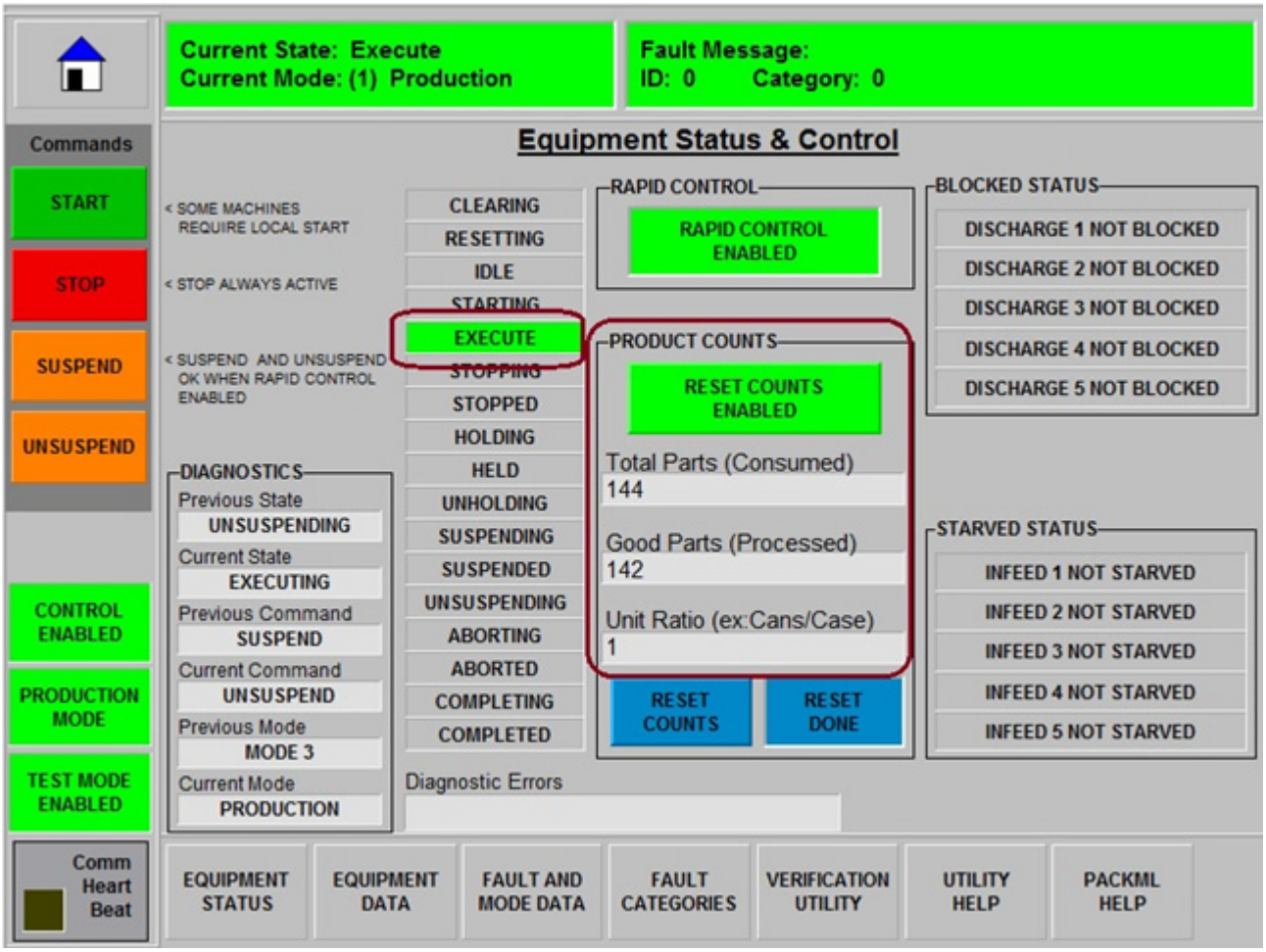
TIP The beacon stack lights of the machine must also reflect the RAPID states. The specification for the colors can be found in the 5-Lamp Beacon Stack Light Specification (EAS007).

Validate Counters in the Interface

When the machine is producing, the Total Parts Counter and the Good Parts counter increment as material is processed in the machine. To test if the interface works properly, the verification of the counts can be done with the status utility as shown in Figure 28.

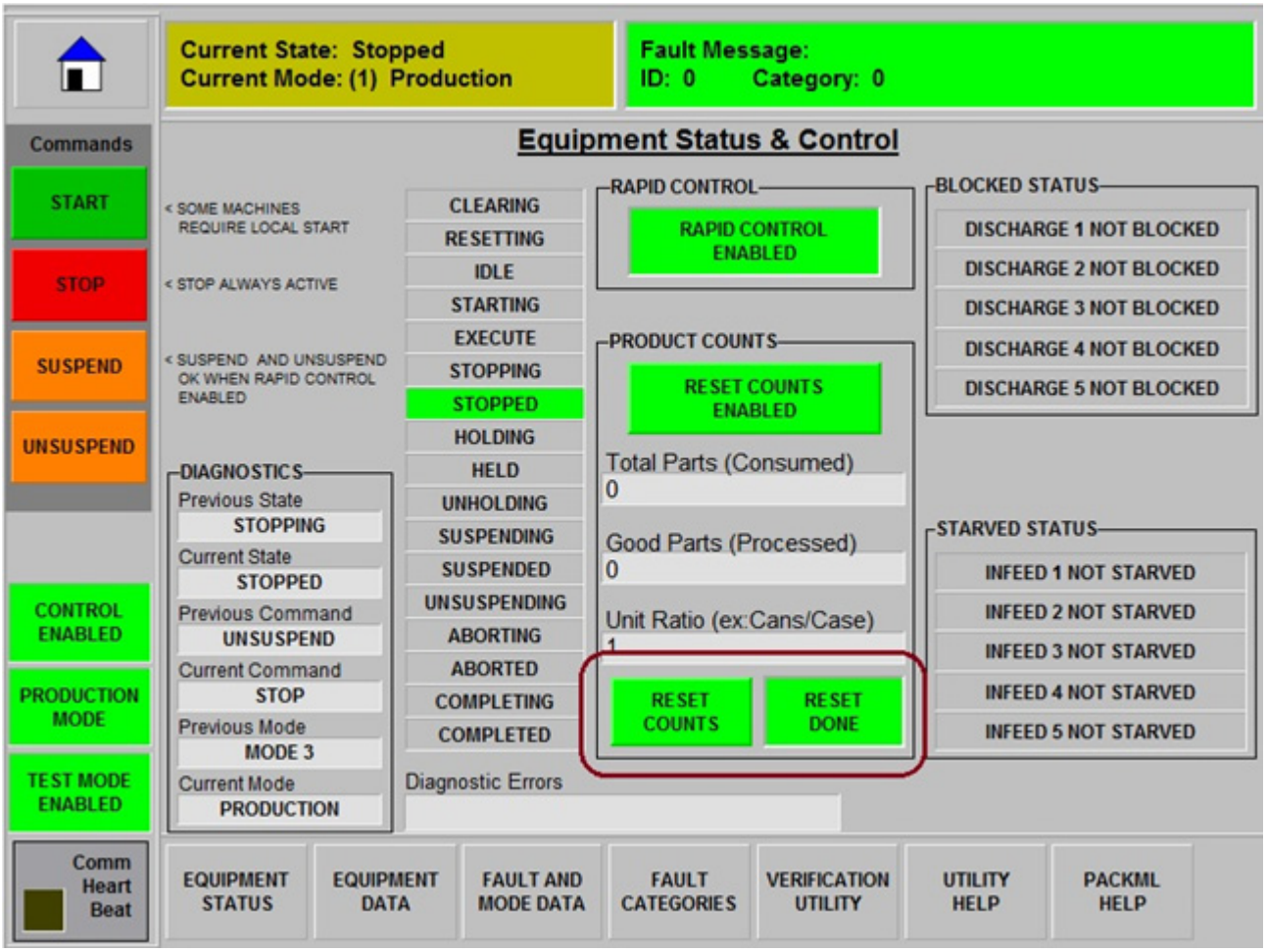
Also the Unit Ratio, which is set in the interface, in Figure 28.

Figure 28 - Equipment Status Display Counters and Unit Ratio



If the Reset Counter Option is enabled, you can also reset the counters here and see the reset done response of the machine. Test this function if the equipment supplier implements it.

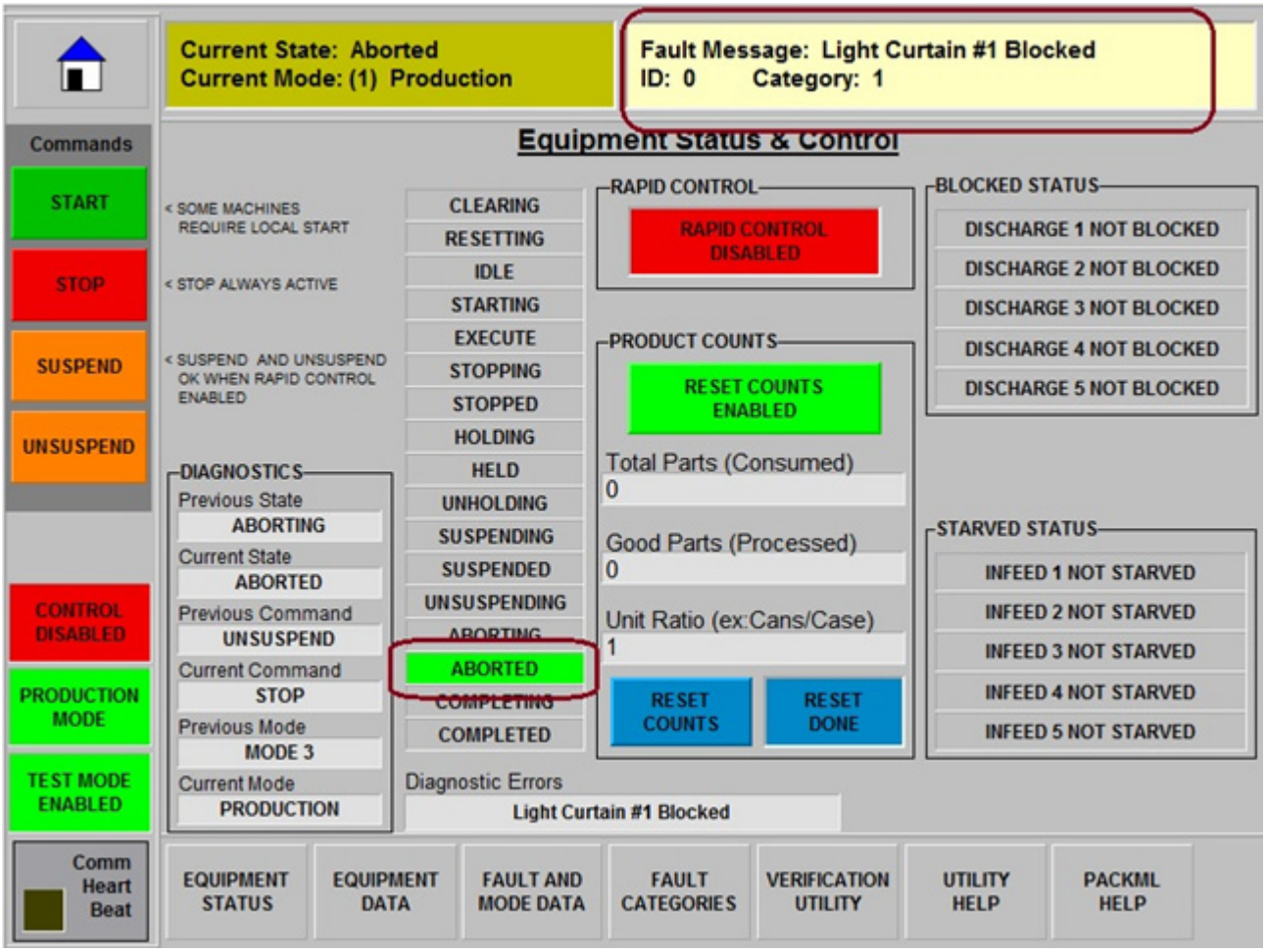
Figure 29 - Equipment Status Display Directly After the Reset Counts button was Clicked and the Reset Done Handshake Came From the Machine



Validate Faults on the Status Display

Faults of the machine can be verified by using the Fault Message and Fault Category Display combined with the states as shown in Figure 30.

Figure 30 - Fault Message on the Equipment Status Display



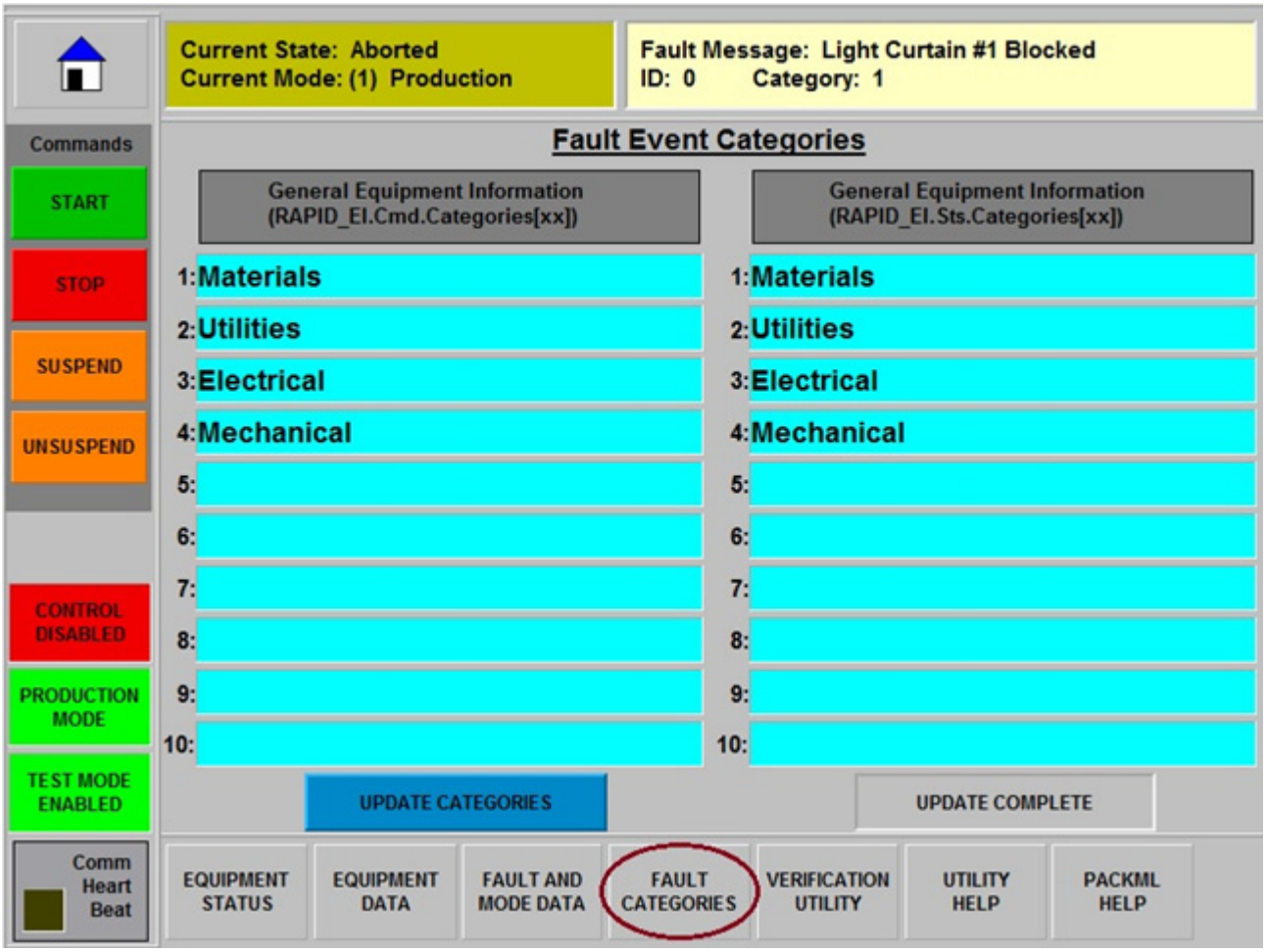
In this example, the faults are classified into four categories.

- Material
- Utilities
- Electrical
- Mechanical

A random selection of the total faults is tested during acceptance tests because many fault codes normally exist at a machine.

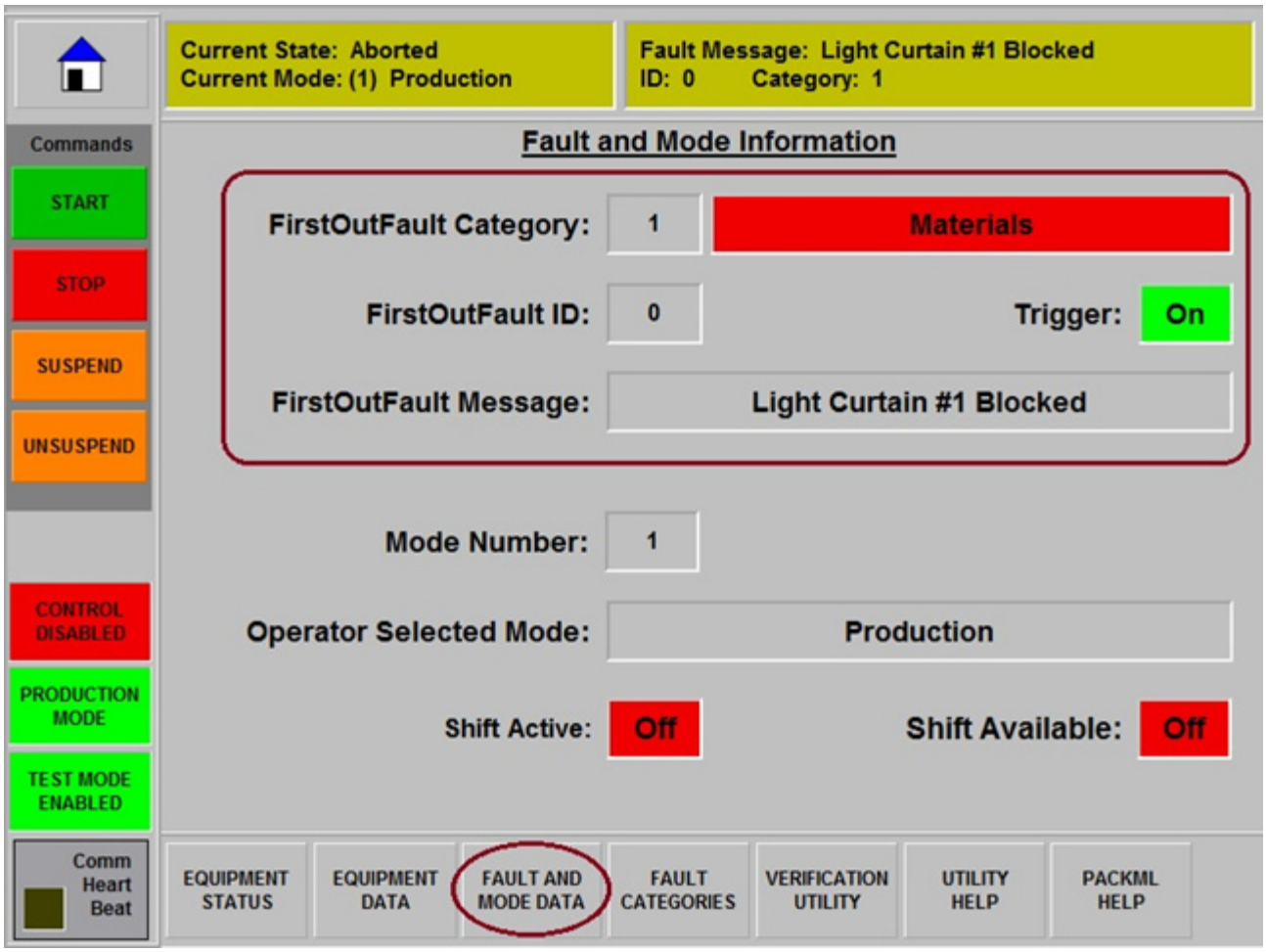
The fault categories can be written from the Verification Utility to the controller in the display Fault Categories. This function simulates the RAPID Server but is normally not needed during tests.

Figure 31 - Fault Categories Display



Fault status and information can also be visualized on the Fault and Mode Data display as shown in Figure 32.

Figure 32 - Fault Information on the Fault and Mode Data Display



Testing RAPID Line Control

If the equipment supplier has implemented Line Control, the function of standard line control can be tested with the Verification Utility.

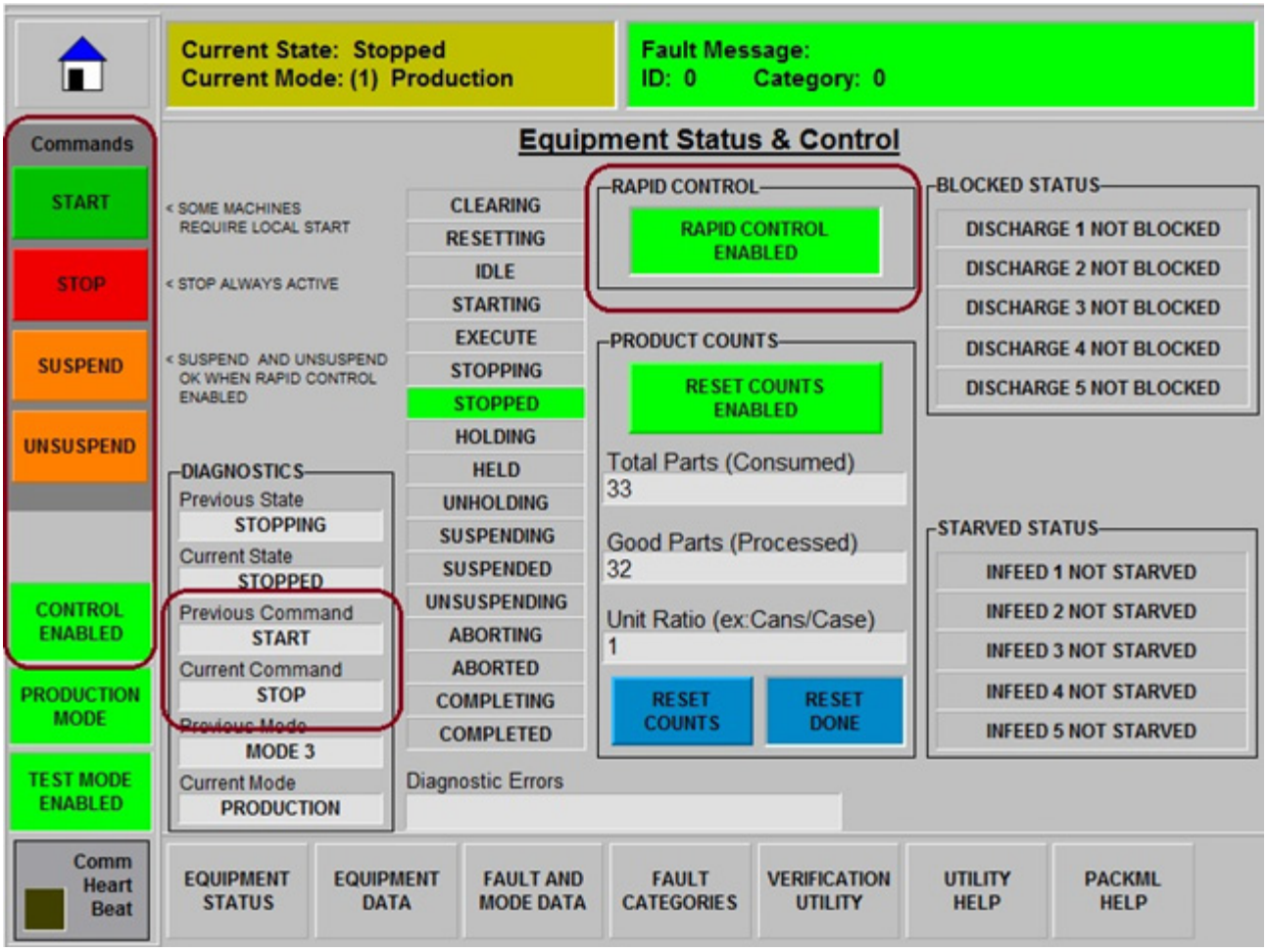
To control the machine with the Verification Utility, the equipment supplier must enable line control in the interface by setting the RAPID_EM.Sts.LineControlEnabled Bit to 1 as in Figure 33.

Figure 33 - RAPID_EM.Sts.LineControlEnabled Set to 1

The RAPID Verification Utility indicates whether Line Control is enabled and allows commands to be given to the machine.

To verify that the machine reacts as expected by the equipment supplier, test the commands.

Figure 34 - Line Control Test Buttons on the Equipment Status Display



TIP For line control requirements, refer to the end user or system integrator of each RAPID Line Integration Solution project/machined delivered, as the requirements can vary. The typical requirements are:

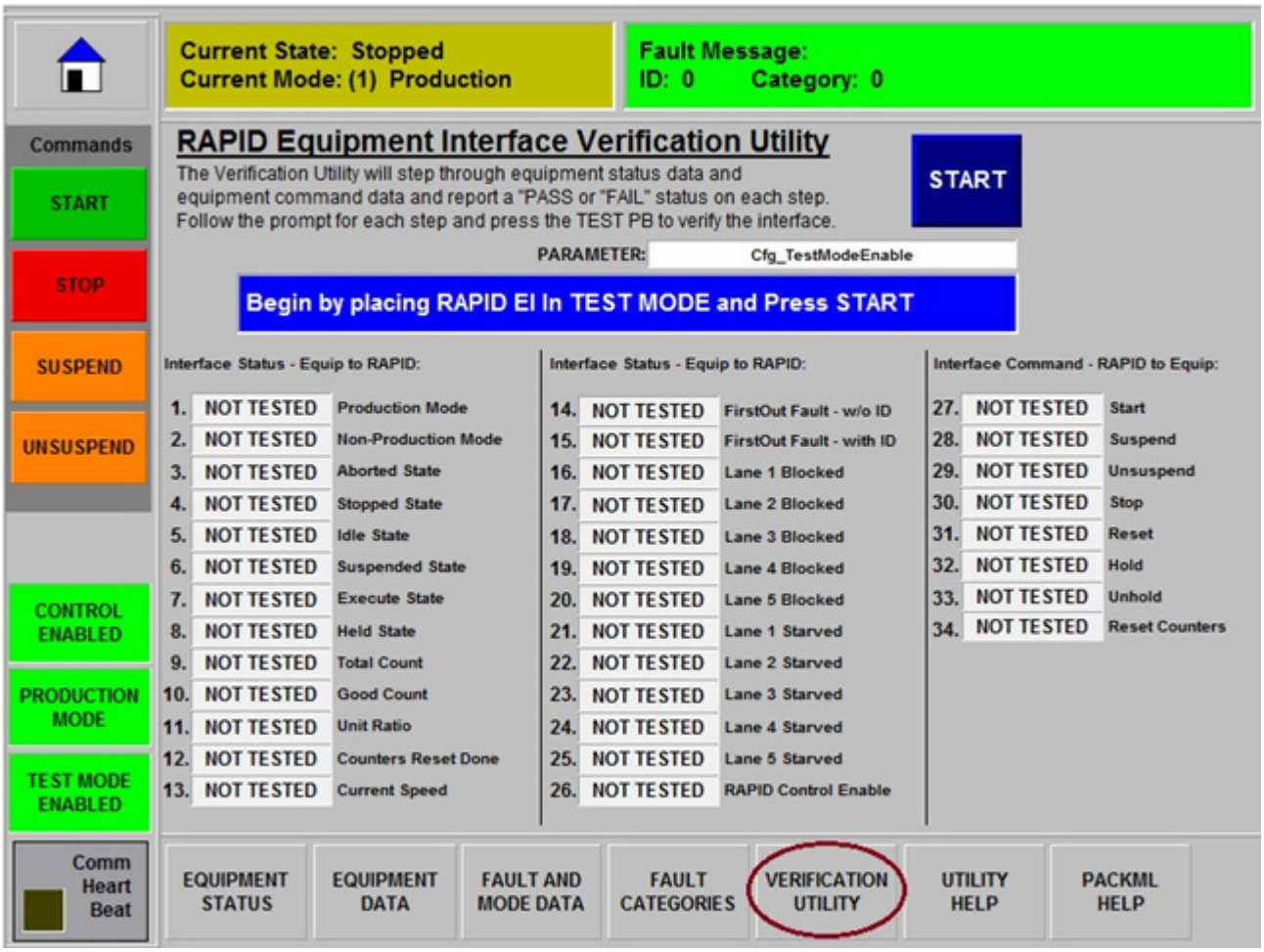
- No control commands required.
- Suspend/Unsuspend Only. (Line control only, no startup/shutdown sequencing)
- Suspend/UnSuspend/Stop functions. (Full line control w/manual start only)
- Full Control Required. (Suspend, UnSuspend, Start, and Stop)

Test the Interface Response

The Verification Utility display can be used to test equipment status data and equipment command data automatically. Each step, when executed, reports a PASS or FAIL status.

TIP The verification does not start unless the Tag Cfg_TestModeEnable of the Add-On Instruction Interface is set to 1, which places the interface in test mode.

Figure 35 - Verification Utility Display

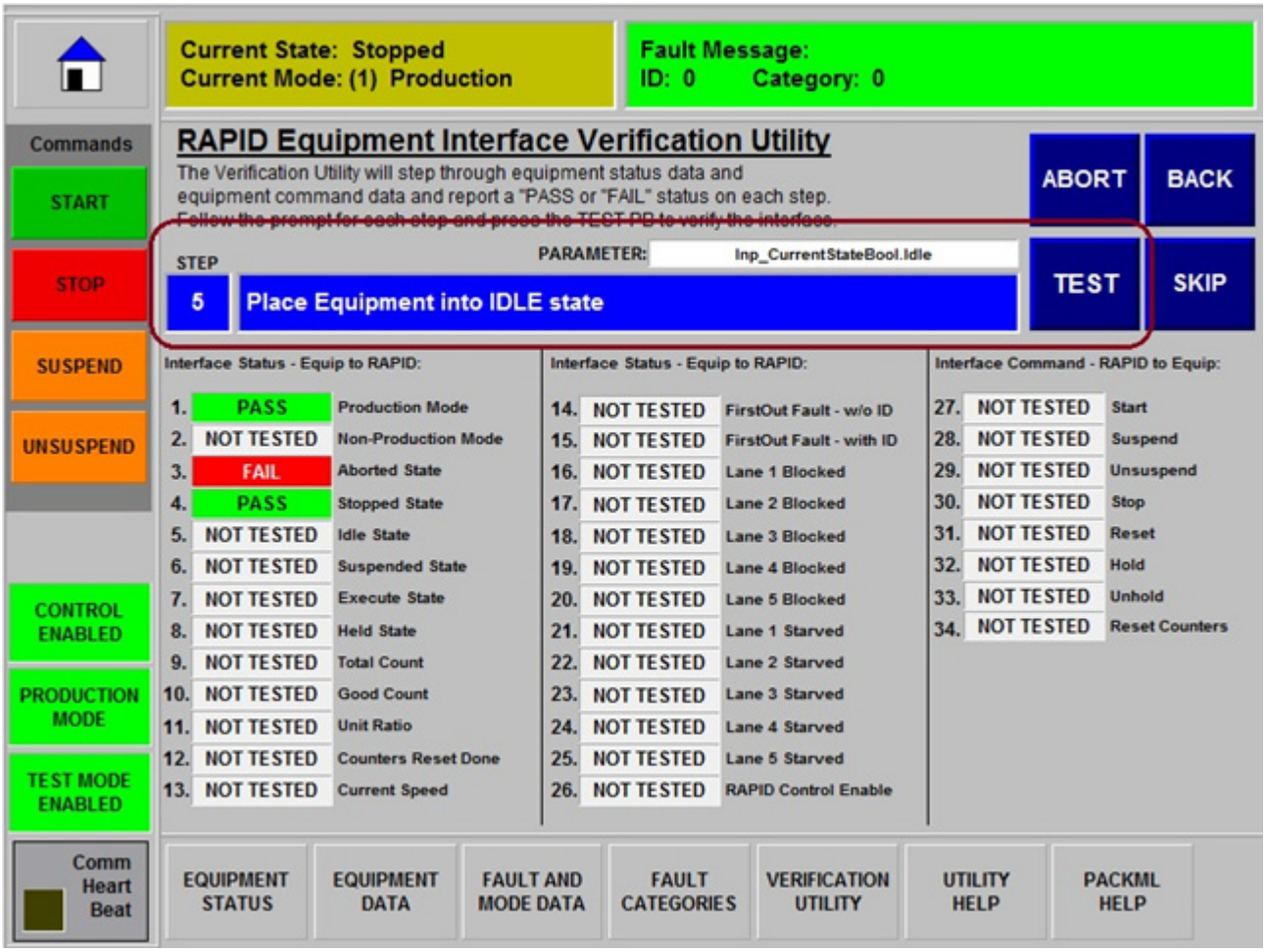


Once started, a verification can be skipped or pointed back to any step for retest. The verification can also be aborted at any time.

TIP All PASS and FAIL status information on the display is lost when the verification is aborted.

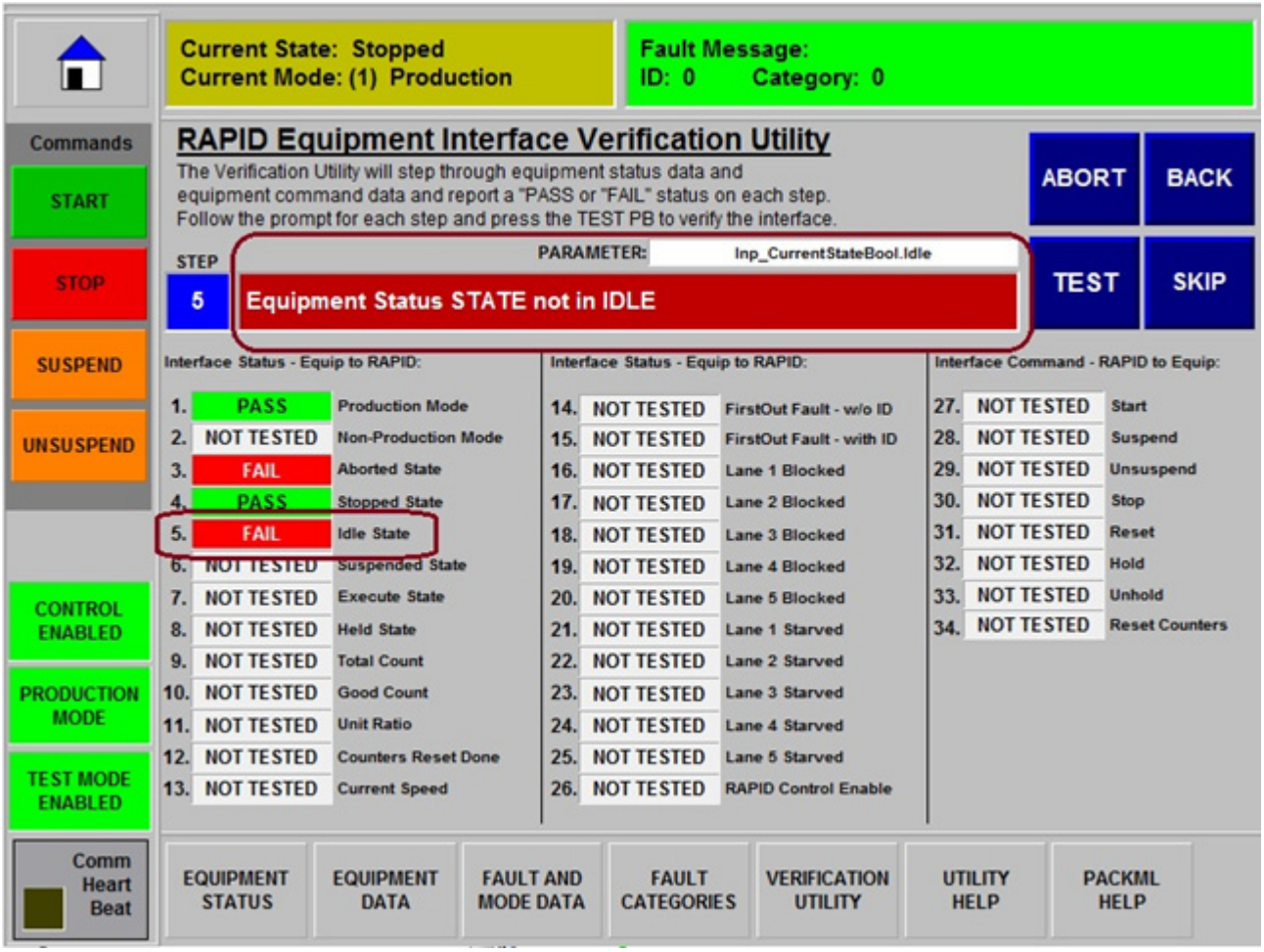
The Verification Utility wizard displays the step number and message for the current step as shown in Figure 36. The RAPID Equipment Interface Add-On Instruction parameter that is tested is also displayed.

Figure 36 - Step and Test Information Shown on the Verification Utility Display

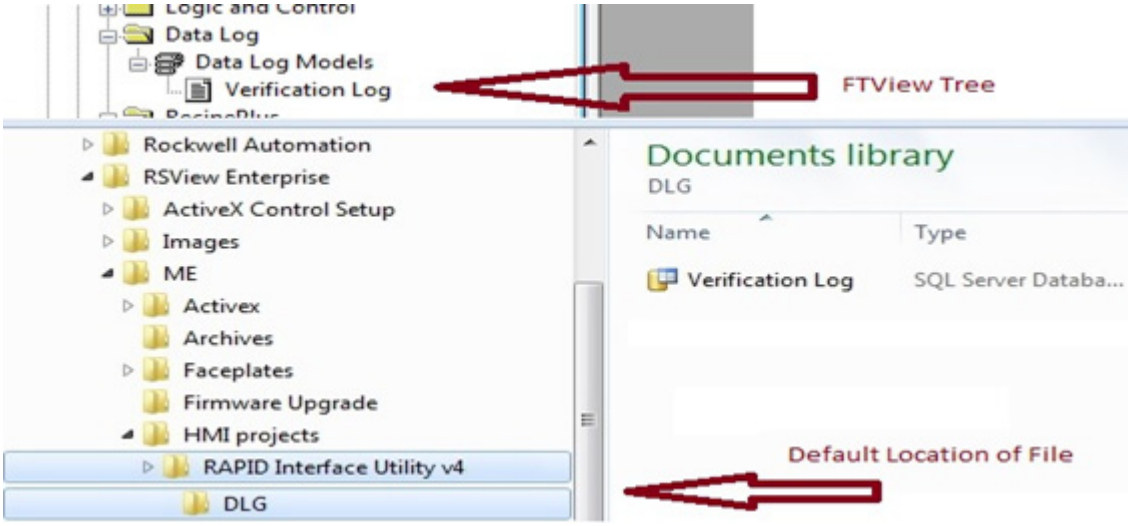


For each step, the equipment supplier places the machine that corresponds to the message displayed for that step then clicks Test. If the system passes the test, then the utility increments to the next step. If the system fails the test, then the message displays the reason for the failure and remains on the current step, as shown in Figure 37.

Figure 37 - Step 5 Test Failed Message



A Log is created once verification starts and is stored in the Verification Log file. The log file stores “PASS”, “FAIL”, and “Not Tested” information for each step. The file (which is created in the FactoryTalk View data logger) can be converted to .csv format for use in excel.



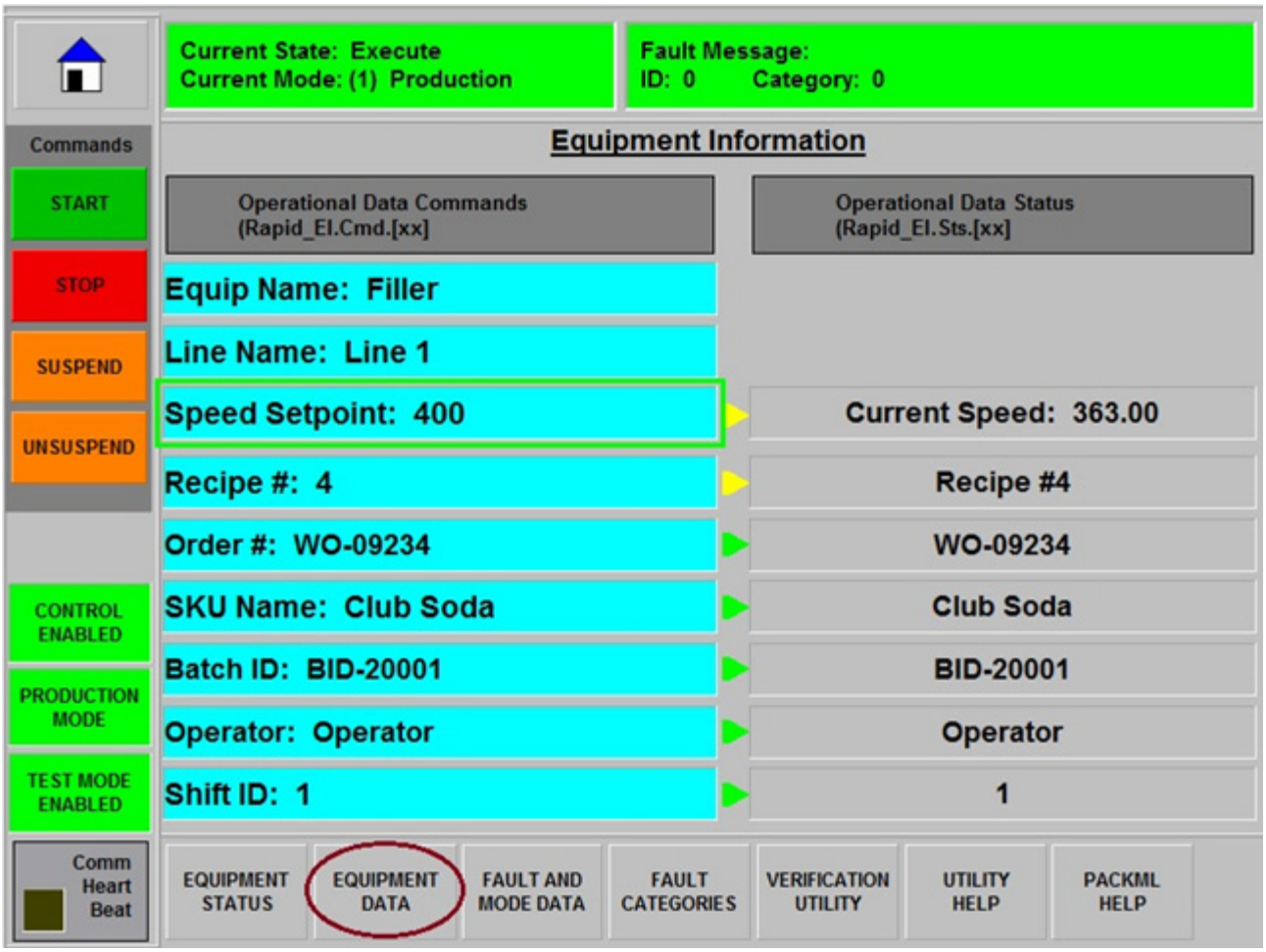
Equipment Data Display

The Equipment Data Display can be used to show the optional data, which can be sent to the machine by the Line Controller. If the equipment supplier uses this data to trigger activities on the machine, use the Verification Utility to trigger activities without the presence of a RAPID Server. Type information manually into the RAPID_EI.Cmd tags.

In the Figure 38, a Batch ID and a recipe number are entered manually.

This display also shows the Current Speed of the machine.

Figure 38 - The Equipment Data Display



If a RAPID Server is present, the data is populated from there.

Notes:

OMAC PackML State Model

Definitions of the OMAC PackML States that are taken from the PackML Standard ISA-TR88.00.02, Chapter 4.

TIP For further information, see the standard document ISA-TR88.00.02.

Types of States

For the purposes of understanding, three machine state types are defined:

- Acting State: A state that represents some processing activity. It implies the single or repeated execution of steps in a logical order, for a finite time or until a specific condition has been reached. In ISA-88 states that end in "-ing" are referred to Transient states.
- Wait State: A state that is used to identify that a machine has achieved a defined set of conditions. In such a state, the machine maintains a status until it transitions to an Acting state or the Dual state. In ISA-88, the Wait state was referred to as a Final or Quiescent state.
- Dual State: A Wait state that is causing the machine to behave as in an Acting state. The Dual state is representative of a Machine state that can be continuously transitioning between Acting and Waiting, and Looping, as defined by the logical sequence required. As noted in ISA-88, the Execute, or Running state, is a Transient state. This Machine state has been recharacterized to include the diversity of operation that is found in packaging and discrete machines.

Table 2 - OMAC PAckML States

State Name	Description
STOPPED State Type: Wait	The machine is powered and stationary after completing the STOPPING state. All communications with other systems are functioning (if applicable). A Reset command causes an exit from STOPPED to the RESETTING state.
STARTING State Type: Acting	This state provides the steps that must be taken to start the machine and is a result of a starting type command (local or remote). After this command, the machine begins to Execute.
IDLE State Type: Wait	This state indicates that RESETTING is complete. This state maintains the machine conditions that were established during the RESETTING state and performs operations that are required when the machine is in IDLE.
SUSPENDING State Type: Acting	This state is a result of a change in monitored conditions due to process conditions or factors. The trigger event causes a temporary suspension of the EXECUTE state. SUSPENDING often results from starvation of upstream material in-feeds (for example, container feed, beverage feed, crown feed, or lubricant feed) that is outside the dynamic speed control range. It can also result from a downstream outfeed blockage that helps prevent the machine from EXECUTING continued steady production. During the controlled sequence of SUSPENDING, the machine transitions to a SUSPENDED state. The operator can force the SUSPENDING state.

Table 2 - OMAC PAckML States

State Name	Description
SUSPENDED State Type: Wait	The machine can run at a relevant setpoint speed without product being produced while the machine is waiting for external process conditions to return to normal. When the offending process conditions return to normal, the SUSPENDED state transitions to UNSUSPENDING and hence continue towards the normal EXECUTE state. Note: The SUSPENDED state can be reached as a result of abnormal external process conditions and differs from HELD. HELD typically result from an operator request or an automatically detected machine fault condition that must be corrected before an operator request to transition to the UNHOLDING state is processed.
Acting UNSUSPENDING State Type	This state is a result of a machine generated request from SUSPENDED state to return to the EXECUTE state. In this state, the system can ramp up speeds, turn on vacuums, and re-engage clutches. This state is done before EXECUTE state, and prepares the machine for the EXECUTE state.
EXECUTE State Type: Dual	Once the machine is processing materials, it is deemed to be executing or in the EXECUTE state. Different machine modes result in specific types of EXECUTE activities. For example, if the machine is in the Production mode, EXECUTE results in products being produced. In Clean Out mode, the EXECUTE state refers to the action of cleaning the machine.
STOPPING State Type: Acting	This state executes the logic that brings the machine to a controlled stop as reflected by the STOPPED state. Normal STARTING of the machine cannot be initiated unless RESETTING had taken place.
ABORTING State Type: Acting	The ABORTED state can be entered at any time in response to the Abort command or on the occurrence of a machine fault. The aborting logic brings the machine to a rapid safe stop. Operation of the emergency stop causes the safety system.to stop the machine. It also provides a signal to initiate the ABORTING State.
ABORTED State Type: Wait	This state maintains machine status information relevant to the Abort condition. The machine can only exit the ABORTED state after an explicit Clear command, following manual intervention to correct and reset the detected machine faults.
HOLDING State Type: Acting	When the machine is in the EXECUTE state, the Hold command can be used to start HOLDING logic. This logic brings the machine to a controlled stop or to a state that represents HELD for the particular unit control mode. A machine can go into this state either when an internal equipment fault is automatically detected or by an operator command. The Hold command stops the machine while the operator intervenes manually in the process, such as to remove a broken bottle from the in-feed. The operator then restarts execution when normal conditions are restored. To restart production correctly after the HELD state, the system preserves data from the point at which the Hold command is given. All relevant process setpoints and the return status of procedures at the time of receiving the Hold command must be saved in the machine controller when executing the HOLDING procedure.
HELD State Type: Wait	The HELD state holds operation while material blockages are cleared, stops throughput while a downstream problem is resolved, or enables the safe correction of an equipment fault before production resumes.
UNHOLDING State Type: Acting	The UNHOLDING state is a response to an Operator command to resume the EXECUTE state. The Unhold command retrieves the saved setpoints and restores status conditions to prepare the machine to reenter the normal EXECUTE state. Note: An operator Unhold command is always required and UNHOLDING can never be initiated automatically.
COMPLETING State Type: Acting	This state is an automatic response from the EXECUTE state. Normal operation has run to completion (for example, processing of material at the infeed stops).
COMPLETE State Type: Wait	The machine has finished the COMPLETING state and is now waiting for a Reset command before transitioning to the RESETTING state.
RESETTING State Type: Acting	This state is the result of a RESET command from the STOPPED or complete state. RESETTING typically causes a machine to sound a horn and place the machine in a state where components are energized awaiting a START command.
CLEARING State Type: Acting	Initiated by a state command to clear any faults that occurred when ABORTING and faults that are present in the ABORTED state before proceeding to a STOPPED state.

A comparison of the ISA S88 states (ISA 88.01) with the OMAC PackML states in Table 2 is shown in Table 3.

Table 3 - Compare ISAS88 States with OMAC PAckML States

ISA 88.01 Example Procedural States	Technical Report Equipment States			
	Value	Unit/Machine States	Wait	Acting
<not defined>	1	Clearing		x
Stopped	2	Stopped	x	
<not defined>	3	Starting		x
Idle	4	Idle	x	
Paused	5	Suspended	x	
Execute	6	Execute	x	x
Stopping	7	Stopping		x
Aborting	8	Aborting		x
Aborted	9	Aborted	x	
Holding	10	Holding		x
Held	11	Held	x	
Restarting	12	Unholding		x
Pausing	13	Suspending		x
<not defined>	14	Unsuspending		x
<not defined>	15	Resetting		x
<not defined>	16	Completing		x
Complete	17	Complete	x	

Notes:

Rockwell Automation Support

Use the following resources to access support information.

Technical Support Center	Knowledgebase Articles, How-to Videos, FAQs, Chat, User Forums, and Product Notification Updates.	https://rockwellautomation.custhelp.com/
Local Technical Support Phone Numbers	Locate the phone number for your country.	http://www.rockwellautomation.com/global/support/get-support-now.page
Direct Dial Codes	Find the Direct Dial Code for your product. Use the code to route your call directly to a technical support engineer.	http://www.rockwellautomation.com/global/support/direct-dial.page
Literature Library	Installation Instructions, Manuals, Brochures, and Technical Data.	http://www.rockwellautomation.com/global/literature-library/overview.page
Product Compatibility and Download Center (PCDC)	Get help determining how products interact, check features and capabilities, and find associated firmware.	http://www.rockwellautomation.com/global/support/pcdc.page

Documentation Feedback

Your comments will help us serve your documentation needs better. If you have any suggestions on how to improve this document, complete the How Are We Doing? form at http://literature.rockwellautomation.com/idc/groups/literature/documents/du/ra-du002_-en-e.pdf.

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

Rockwell Otomasyon Ticaret A.Ş., Kar Plaza İş Merkezi E Blok Kat:6 34752 İçerenköy, İstanbul, Tel: +90 (216) 5698400

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 NV, Pegasus Park, De Kleetlaan 12a, 1831 Diegem, 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

Publication RAPID-QS001C-EN-P - August 2016
Supersedes Publication RAPID-QS001B-EN-P - May 2014