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.

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>![Warning]</td>
<td>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.</td>
</tr>
<tr>
<td>![Attention]</td>
<td>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.</td>
</tr>
<tr>
<td>![Important]</td>
<td>Identifies information that is critical for successful application and understanding of the product.</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>![Shock Hazard]</td>
<td>Labels may be on or inside the equipment, for example, a drive or motor, to alert people that dangerous voltage may be present.</td>
</tr>
<tr>
<td>![Burn Hazard]</td>
<td>Labels may be on or inside the equipment, for example, a drive or motor, to alert people that surfaces may reach dangerous temperatures.</td>
</tr>
<tr>
<td>![Arc Flash Hazard]</td>
<td>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).</td>
</tr>
</tbody>
</table>

Allen-Bradley, FactoryTalk, Logix5000, Rockwell Software, Rockwell Automation, RSLinx, and RSLogix are trademarks of Rockwell Automation, Inc.

Trademarks not belonging to Rockwell Automation are property of their respective companies.
This manual contains new and updated information. Changes throughout this revision are marked by change bars, as shown to the right of this paragraph.

### New and Updated Information

This table contains the changes made to this revision.

<table>
<thead>
<tr>
<th>Topic</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Updated the example for logic to detect in-feed starved and discharge blocked conditions.</td>
<td>24</td>
</tr>
<tr>
<td>Updated the link to the RAPID Equipment Interface OEM toolkit.</td>
<td>34</td>
</tr>
</tbody>
</table>
You must complete the steps described in the following graphic to implement the RAPID Equipment Interface Add-On Instruction. The tasks required to complete each step are listed in the respective chapters.

1. Unzip the RAPID Equipment Interface OEM Toolkit.zip file
2. Chapter 1 - Install the RAPID Equipment Interface
3. Chapter 2 - RAPID Equipment Interface Data Mapping
4. Chapter 3 - Test the Application
Notes:
# Table of Contents

**Preface**
- About This Publication .................................................. 9
- RAPID Equipment Interface Toolkit .................................. 9
- RAPID Equipment Interface Recommendations ..................... 10

**Chapter 1**
- Install the RAPID Equipment Interface
  - Begin You Begin .......................................................... 11
  - Import the RAPID Equipment Interface Add-On Instruction ...... 12
  - Verify There Are No Conflicts ...................................... 13

**Chapter 2**
- RAPID Equipment Interface Data Mapping
  - Data Mapping for RAPID Line Performance ......................... 16
    - AOI Tag Mapping Requirements for RAPID Line Performance Functions ................................................. 16
    - Data Mapping Logic Requirements for RAPID Line Performance Functions ................................................. 17
  - Data Mapping for RAPID Line Control ............................... 27
    - AOI Tag Mapping Requirements for RAPID Line Control Functions ......................................................... 27
    - Data Mapping Logic for Line Control Functions ............... 28
  - RAPID Equipment Interface with ISA-TR88.00.02 (PackML) .... 31
    - Additional Resources .................................................. 32

**Chapter 3**
- Test the Application
  - Testing Tool Instructional Videos .................................. 34

**Appendix A**
- RAPID_EM Data Structure .................................................. 35

**Appendix B**
- RAPID Equipment Interface Add-On Instruction Online Help File (AOI Inputs/Outputs) ........................................ 37

**Appendix C**
- Memory Usage in Controller Program .................................... 39
About This Publication

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.

Users of this document 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 target equipment's control program, and associated equipment functions.

RAPID Equipment Interface Toolkit

This document is used with the RAPID Equipment Interface OEM Toolkit. You can access the toolkit at the Rockwell Automation Technical Support Knowledge Base, article 66060, at the following address:

https://rockwellautomation.custhelp.com/

The toolkit includes the following:

- Implementation instructions, that is, this document
- Import files containing 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 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 very 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 determine how to 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 signed so modifications can be detected. **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 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. This is particularly 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** Remember the following:

Machine programs 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_EM 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 ok 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 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 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.
Install the RAPID Equipment Interface

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

<table>
<thead>
<tr>
<th>Topic</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Begin You Begin</td>
<td>11</td>
</tr>
<tr>
<td>Import the RAPID Equipment Interface Add-On Instruction</td>
<td>12</td>
</tr>
<tr>
<td>Verify There Are No Conflicts</td>
<td>13</td>
</tr>
</tbody>
</table>

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 described in this chapter manually. We recommend that you import the files provided in folder 02_AppFiles. If you use the base Add-On Instruction and UDT definition LSX files, additional work is required to fill in all of the tag references on the Add-On Instruction. The example files include the mapping UDTs.

Begin 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's logic, right-click a rung and choose Import Rungs.

![Import Rungs](image)

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

![Import LSX File](image)
The Import Configuration dialog box appears.

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

**Verify There Are No Conflicts**

Before you click OK to complete the import, verify there are no conflicting tags, data types, or Add-On Instructions, considering 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 UDT’s in the target program must match the RAPID definition of all UDT’s or communications are impacted.

Click OK to complete the import.
When the import process is complete, the rung shown below 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.

<table>
<thead>
<tr>
<th>Topic</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data Mapping for RAPID Line Performance</td>
<td>16</td>
</tr>
<tr>
<td>Data Mapping for RAPID Line Control</td>
<td>27</td>
</tr>
<tr>
<td>RAPID Equipment Interface with ISA-TR88.00.02 (PackML)</td>
<td>31</td>
</tr>
</tbody>
</table>

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 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 described in this chapter.

If the target equipment program was programmed per ISA-TR88.00.02 / PackML, note the information provided on page 31 regarding an alternative method to implementing the equipment interface for PackML based programs.

For more information on the data structure, see Appendix A.
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 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):
- CurrentSpeed: (REAL) Actual Production Rate (parts per minute)
- Blocked and Starved Conditions: (BOOL)
  - At Least one Downstream Blocked condition is required.
  - At Least one Upstream Starved condition is required.
- Total Count: Equipment Infeed Count. (Consumed)
- Good Count: Equipment Discharge Count (Processed)
- Unit Ratio: Unit of Measure ratio of total to good.
  - 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

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 .LSX 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 commonly associated with OEE and Machine Performance applications, as well as 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 needs to 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 also must be factored into this math.

<table>
<thead>
<tr>
<th>RAPID/Overall Effectiveness (OEE) Terms</th>
<th>ISA-TR88.00.02 / PackML Terms</th>
<th>Measured At</th>
<th>Example: Cartoner Making Twelve Packs</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total Parts</td>
<td>Consumed Parts</td>
<td>Infeed</td>
<td>Cans</td>
</tr>
<tr>
<td>Good Parts</td>
<td>Processed Parts</td>
<td>Discharge</td>
<td>Cartons</td>
</tr>
<tr>
<td>Unit Ratio</td>
<td>N/A</td>
<td>SKU-based</td>
<td>Value =12</td>
</tr>
</tbody>
</table>
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 has to be calculated, then special consideration has to be given to this situation as the calculation becomes invalid if a part count used in the calculation rolls over at a different 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 following 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.

By 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.

![Labeler Station Diagram]

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

![Pack Station Diagram]

**Unit Ratio** = (6) Quantity of consumed parts 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 and the Good parts count must be calculated by using the following formula:

Total Parts = (Discharge sensor counts * Unit Ratio)
- Unit of Measure is Bottles (This is 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, then this results in a unit ratio that is not consistent with the first unit ratio rule described in section Part Counting - Total and Good Parts Counts Provided.

This sensor/device is counting the parts that this machine has processed. Because defects are being counted downstream, this count is used to calculate the Inp_TotalParts value.

This sensor/device is counting defects produced by the pack station that is used to calculate and populate the Inp_GoodParts value. See the example below for how to map this tag value.

**IMPORTANT:** Rollover of all counter values 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 to make sure that a rollover of a part counter does not cause the calculation to become invalid. This logic ensures 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 up one (1) 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 required to create one discharged product. If this information is not available in the machine, it likely is always one (1) 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 makes use of more than one in-feed or discharge lane.

**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 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 ensure 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 ensure 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, prior to 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 in lieu of a message, he must provide a cross reference list of IDs and corresponding messages to the RAPID system integrator, prior to 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 below, 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 used with this example.

---

**Current Speed Status (Inp_EquipmentActualSpeed)**

The equipment speed status is usually 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 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.

- **State**: (DINT) or StateBool (BOOL Structure)
- **CurrentSpeed**: (REAL) Actual Production Rate (parts per minute)
- **LineControlEnabled**: (BOOL) Indicates that the equipment lets remote commands from the line control.
- **Mode**: (DINT) 
  - Production = 1
  - Non-Production > 1
- **Blocked and Starved Conditions**: (BOOL) 
  - At Least one Downstream Blocked condition is required.
  - At Least one Upstream Starved condition is required.

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 the current equipment state:
- StateCmdBool (DINT)
- StateCmdBool (BOOL Structure)
  - The ISA88-TR88.00.02 Report (PackML): defines States.

Use one of these tags to indicate to the current equipment the desired machine state:
- RAPID_EI.Cmd
- RAPID_EI.Sts
  - The ISA88-TR88.00.02 Report (PackML): defines States.
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_RAPIDControlEnabled) 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 Un-suspend 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.
Chapter 2  RAPID Equipment Interface Data Mapping

The following example rungs show how the logic in the example State Command Logic on page 29 ties to the machine's state model command interface.

**Remaining Tags Required for RAPID Line Control Functions**

All other tags 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 so you still 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 1.
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.
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 FactoryView StudioME interface testing tool lets you to 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 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](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 used with the RAPID_EM data structure.

The following table describes the RAPID_EM status tags.

<table>
<thead>
<tr>
<th>UDT_RAPID_EM_Sts</th>
<th>Description</th>
<th>Data Format</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Required</strong>: Current Mode of Equipment.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>.Mode</td>
<td>Description</td>
<td>Data Format</td>
</tr>
<tr>
<td>1= Production Mode</td>
<td>.Mode</td>
<td>DINT</td>
</tr>
<tr>
<td>2= Clean In Place (CIP)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3= Change Over</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4= Maintenance/Manual</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5= Other</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Name of Current Mode if it is not as defined above in the target program.</td>
<td>.ModeName</td>
<td>STRING</td>
</tr>
<tr>
<td><strong>Required</strong> if .State is not being used. This tag presents the equipment’s current state with a unique boolean tag representing each state. All states are mutually exclusive, as defined by ISA-TR88.00.02.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>IMPORTANT: If .State is not zero, this tag is ignored.</td>
<td>StateBool</td>
<td>UDT_PML_States</td>
</tr>
<tr>
<td>Integer values are defined by ISA-TR88.00.02 (PackML) as:</td>
<td>.State</td>
<td>DINT</td>
</tr>
<tr>
<td>1= Clearing (Optional state)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2= Stopped [Required state: Default state if not Running, Suspended, or Faulted]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3= Starting (Optional state)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4= Idle (Optional state)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5= Suspended [Required state: External upstream/downstream permissive is present]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6= Execute [Required state: Running]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>7= Stopping (Optional state)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>8= Aborting (Optional state)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>9= Aborted [Required state: .FirstOutFault values must be provided when in this state]</td>
<td>.StateName</td>
<td>STRING</td>
</tr>
<tr>
<td>10= Holding (Optional state)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>11= Held (Optional state)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>12= UnHolding (Optional state)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>13= Suspending (Optional state)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>14= Unsuspending (Optional state)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>15= Resetting (Optional state)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>16= Completing (Optional state)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>17= Complete (Optional state)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Optional: Name of current mode if not as defined as above.</td>
<td>.RecipeName</td>
<td>STRING</td>
</tr>
<tr>
<td>Optional: Name of current recipe being used by equipment.</td>
<td>.ProducingMode</td>
<td>BOOL</td>
</tr>
<tr>
<td>[Reserved for future use: Design Speed is set on RAPID Supervisory HMI]</td>
<td>.DesignSpeed</td>
<td>DINT</td>
</tr>
<tr>
<td><strong>Requirement</strong>: Set to boolean value=1 (True) if you must monitor the current mode for performance.</td>
<td>.LineControlEnabled</td>
<td>BOOL</td>
</tr>
</tbody>
</table>
### Appendix A  RAPID_EM Data Structure

The following table describes the RAPID_EM command tags.

<table>
<thead>
<tr>
<th>UDT_RAPID_EM_Cmd</th>
<th>Description</th>
<th>Data Format</th>
</tr>
</thead>
<tbody>
<tr>
<td>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.</td>
<td>.StateCmdBool Suspends .Unsuspend .Stop .Start (optional)</td>
<td>UDT_PML_Commands</td>
</tr>
<tr>
<td>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]</td>
<td>.StateCmdInt</td>
<td>DINT</td>
</tr>
<tr>
<td>Optional: Can be used on local HMI</td>
<td>.Name</td>
<td>STRING</td>
</tr>
<tr>
<td>Optional: Can be used on local HMI</td>
<td>.LineName</td>
<td>STRING</td>
</tr>
<tr>
<td>Optional: Can be used to set current running machine recipe. Requires machine level programming to implement.</td>
<td>.Recipe</td>
<td>DINT</td>
</tr>
<tr>
<td>Reserved for future use</td>
<td>.SpeedSetPoint</td>
<td>REAL</td>
</tr>
<tr>
<td>Reserved for future use</td>
<td>.ResetCounters</td>
<td>BOOL</td>
</tr>
</tbody>
</table>
RAPID Equipment Interface Add-On Instruction Online Help File (AOI Inputs/Outputs)

This chapter describes the online help parameters 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 table below describes the online help parameters. The parameters are required for line control programming, line performance programming, or both.

These parameters must be controlled by the equipment program for RAPID to function as designed. We recommend that all parameters described in Chapter 1 and Chapter 2 are programmed.

The following table describes the RAPID_EM status tags.

<table>
<thead>
<tr>
<th>Required</th>
<th>Parameter Name</th>
<th>Data Type</th>
<th>Usage</th>
</tr>
</thead>
<tbody>
<tr>
<td>x-both</td>
<td>RAPID_Interface_AOI</td>
<td>RAPID_Interface_AOI</td>
<td>InOut</td>
</tr>
<tr>
<td>x-both</td>
<td>EnableIn</td>
<td>BOOL</td>
<td>Input</td>
</tr>
<tr>
<td>view only</td>
<td>EnableOut</td>
<td>BOOL</td>
<td>Output</td>
</tr>
<tr>
<td>x-both</td>
<td>Cfg_TestModeEnable</td>
<td>BOOL</td>
<td>Input</td>
</tr>
<tr>
<td>x-both</td>
<td>Ref_RAPIDCommands</td>
<td>UDT_RAPID_EI_Cmd</td>
<td>InOut</td>
</tr>
<tr>
<td>x-both</td>
<td>Ref_RAPIDStatus</td>
<td>UDT_RAPID_EI_Sts</td>
<td>InOut</td>
</tr>
<tr>
<td>x-control only</td>
<td>Out_StateCommand</td>
<td>DINT</td>
<td>Output</td>
</tr>
<tr>
<td>x-control only</td>
<td>Out_StateCommandBool</td>
<td>UDT_PML_Commands</td>
<td>InOut</td>
</tr>
<tr>
<td>optional</td>
<td>Out_EquipmentName</td>
<td>STRING</td>
<td>InOut</td>
</tr>
<tr>
<td>optional</td>
<td>Out_LineName</td>
<td>STRING</td>
<td>InOut</td>
</tr>
<tr>
<td>optional</td>
<td>Out_Recipe</td>
<td>DINT</td>
<td>Output</td>
</tr>
<tr>
<td>optional</td>
<td>Out_ResetCounters</td>
<td>BOOL</td>
<td>Output</td>
</tr>
<tr>
<td>optional</td>
<td>Out_SpeedSP</td>
<td>REAL</td>
<td>Output</td>
</tr>
<tr>
<td>x-both</td>
<td>Inp_CurrentMode</td>
<td>DINT</td>
<td>Input</td>
</tr>
<tr>
<td>optional</td>
<td>Inp_CurrentModeName</td>
<td>STRING</td>
<td>InOut</td>
</tr>
<tr>
<td>x-or below</td>
<td>Inp_CurrentState</td>
<td>DINT</td>
<td>Input</td>
</tr>
<tr>
<td>x-or above</td>
<td>Inp_CurrentStateBool</td>
<td>UDT_PML_States</td>
<td>InOut</td>
</tr>
<tr>
<td>optional</td>
<td>Inp_CurrentStateName</td>
<td>STRING</td>
<td>InOut</td>
</tr>
<tr>
<td>optional</td>
<td>Inp_RecipeName</td>
<td>STRING</td>
<td>InOut</td>
</tr>
<tr>
<td>x-both</td>
<td>Inp_ProducingModeEnabled</td>
<td>BOOL</td>
<td>Input</td>
</tr>
<tr>
<td>Required</td>
<td>Parameter Name</td>
<td>Data Type</td>
<td>Usage</td>
</tr>
<tr>
<td>-----------</td>
<td>--------------------------------------</td>
<td>-----------</td>
<td>-----------</td>
</tr>
<tr>
<td>optional</td>
<td>Inp_EquipmentActualSpeed</td>
<td>REAL</td>
<td>Input</td>
</tr>
<tr>
<td>x-control only</td>
<td>Inp_RAPIDControlEnabled</td>
<td>BOOL</td>
<td>Input</td>
</tr>
<tr>
<td>x-performance only</td>
<td>Inp_FirstOutFault</td>
<td>UDT_Event</td>
<td>InOut</td>
</tr>
<tr>
<td>x-performance only</td>
<td>Inp_TotalCount</td>
<td>DINT</td>
<td>Input</td>
</tr>
<tr>
<td>x-performance only</td>
<td>Inp_GoodCount</td>
<td>DINT</td>
<td>Input</td>
</tr>
<tr>
<td>x-performance only</td>
<td>Inp_ConsumedUnitRatio</td>
<td>REAL</td>
<td>Input</td>
</tr>
<tr>
<td>optional</td>
<td>Inp_ResetDone</td>
<td>BOOL</td>
<td>Input</td>
</tr>
<tr>
<td>x-both</td>
<td>Inp_BlockedSts_Lane01</td>
<td>BOOL</td>
<td>Input</td>
</tr>
<tr>
<td>optional</td>
<td>Inp_BlockedSts_Lane02</td>
<td>BOOL</td>
<td>Input</td>
</tr>
<tr>
<td>optional</td>
<td>Inp_BlockedSts_Lane03</td>
<td>BOOL</td>
<td>Input</td>
</tr>
<tr>
<td>optional</td>
<td>Inp_BlockedSts_Lane04</td>
<td>BOOL</td>
<td>Input</td>
</tr>
<tr>
<td>optional</td>
<td>Inp_BlockedSts_Lane05</td>
<td>BOOL</td>
<td>Input</td>
</tr>
<tr>
<td>x-both</td>
<td>Inp_StarvedSts_Lane01</td>
<td>BOOL</td>
<td>Input</td>
</tr>
<tr>
<td>optional</td>
<td>Inp_StarvedSts_Lane02</td>
<td>BOOL</td>
<td>Input</td>
</tr>
<tr>
<td>optional</td>
<td>Inp_StarvedSts_Lane03</td>
<td>BOOL</td>
<td>Input</td>
</tr>
<tr>
<td>optional</td>
<td>Inp_StarvedSts_Lane04</td>
<td>BOOL</td>
<td>Input</td>
</tr>
<tr>
<td>optional</td>
<td>Inp_StarvedSts_Lane05</td>
<td>BOOL</td>
<td>Input</td>
</tr>
<tr>
<td>view only</td>
<td>Sts_RAPIDControlActive</td>
<td>BOOL</td>
<td>Output</td>
</tr>
<tr>
<td>view only</td>
<td>Sts_TestModeEnabled</td>
<td>BOOL</td>
<td>Output</td>
</tr>
<tr>
<td>view only</td>
<td>Sts_HeartbeatFault</td>
<td>BOOL</td>
<td>Output</td>
</tr>
</tbody>
</table>
## Memory Usage in Controller Program

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

<table>
<thead>
<tr>
<th>Example Program: Available Free Memory Prior to Implementing the Equipment Interface</th>
<th>Impact of Importing the Equipment Interface Add-On Instruction and Associated UDTs</th>
<th>Impact of Adding the Equipment Interface Data Mapping Logic</th>
</tr>
</thead>
<tbody>
<tr>
<td>2,048,204 bytes</td>
<td>Free Memory After Add-On Instructions and UDTs are Imported: 2,027,256 bytes</td>
<td>Memory Used by Add-On Instructions and UDTs: 20,948 bytes (21 KB) Fixed</td>
</tr>
<tr>
<td></td>
<td>Free Memory After Data Mapping Logic: 2,023,492 bytes</td>
<td>Memory Used by Data Mapping Logic: 3,764 bytes (4 KB) Can vary</td>
</tr>
</tbody>
</table>

**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**

![Estimated Memory Usage](image)

**Impact of Add-On Instructions and UDTs**

![Estimated Memory Usage](image)

**Impact of Mapping Logic**

![Estimated Memory Usage](image)
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 and application notes, sample code, and links to software service packs. You can also visit our Support Center at https://rockwellautomation.custhelp.com/ for software updates, support chats and forums, technical information, FAQs, and to sign up for product notification updates.

In addition, we offer multiple support programs for installation, configuration, and troubleshooting. For more information, contact your local distributor or Rockwell Automation representative, or visit http://www.rockwellautomation.com/services/online-phone.

Installation Assistance

If you experience a problem 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.

<table>
<thead>
<tr>
<th>United States or Canada</th>
<th>1.440.646.3434</th>
</tr>
</thead>
<tbody>
<tr>
<td>Outside United States or Canada</td>
<td>Use the Worldwide Locator at <a href="http://www.rockwellautomation.com/rockwellautomation/support/overview.page">http://www.rockwellautomation.com/rockwellautomation/support/overview.page</a>, or contact your local Rockwell Automation representative.</td>
</tr>
</tbody>
</table>

New Product Satisfaction Return

Rockwell Automation tests all of its products to help 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:

<table>
<thead>
<tr>
<th>United States</th>
<th>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.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Outside United States</td>
<td>Please contact your local Rockwell Automation representative for the return procedure.</td>
</tr>
</tbody>
</table>

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 Otomasyon Ticaret A.Ş., Kar Plaza İş Merkezi E Blok Kar.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 Kleerlaan 12a, 1831 Diegem, Belgium, Tel: (32) 2 663 0600, Fax: (32) 2 663 0640
Asia Pacific: Rockwell Automation, Level 14, Core E, Cyberport 3, 100 Cyberport Road, Hong Kong, Tel: (852) 2887 4788, Fax: (852) 2508 1846

Copyright © 2014 Rockwell Automation, Inc. All rights reserved. Printed in the U.S.A.