OMRON

Sysmac Library

User's Manual for OPC UA PackML Library

SYSMAC-XR101

SYSMAC always in control

W638-E1-02

NOTE

1. All rights reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form, or by any means, mechanical, electronic, photocopying, recording, or otherwise, without the prior written permission of OMRON.
2. No patent liability is assumed with respect to the use of the information contained herein. Moreover, because OMRON is constantly striving to improve its high-quality products, the information contained in this manual is subject to change without notice.
3. Every precaution has been taken in the preparation of this manual. Nevertheless, OMRON assumes no responsibility for errors or omissions.
   Neither is any liability assumed for damages resulting from the use of the information contained in this publication.

Trademarks

- Sysmac and SYSMAC are trademarks or registered trademarks of OMRON Corporation in Japan and other countries for OMRON factory automation products.
- Microsoft, Windows, Excel, Visual Basic, and Microsoft Edge are either registered trademarks or trademarks of Microsoft Corporation in the United States and other countries.
- EtherCAT® is registered trademark and patented technology, licensed by Beckhoff Automation GmbH, Germany.
- ODVA, CIP, CompoNet, DeviceNet, and EtherNet/IP are trademarks of ODVA.
- The SD and SDHC logos are trademarks of SD-3C, LLC.

Other company names and product names in this document are the trademarks or registered trademarks of their respective companies.

Copyrights

- Microsoft product screen shots used with permission from Microsoft.
- This product incorporates certain third party software. The license and copyright information associated with this software is available at http://www.fa.omron.co.jp/nj_info_e/.

Introduction

Thank you for purchasing an NJ/NX-series CPU Unit.
This manual contains information that is necessary to use the function blocks (hereafter, sometimes abbreviated to FB) in the OPC UA PackML Library. Please read this manual and make sure you understand the functionality and performance of the product before you attempt to use it in a control system.
This manual provides function block specifications. It does not describe application restrictions or combination restrictions for Controllers, Units, and components.
Make sure to read the user's manual for each product before use.
Keep this manual in a safe place where it will be available for reference during operation.

Features of the Library

- PackML
  PackML (Packaging Machine Language) is the standard for packaging machines which was standardized by OMAC (The Organization for Machine Automation and Control). Mode and state of devices, and interface with peripheral devices are defined with the standard. PackML enables the consistent operation even in the production lines using devices of multiple suppliers. Refer to TR88.00.02-2015 for details.
- OPC UA for PackML
  OPC UA for PackML is the specifications created between OMAC and OPC Foundation, and uses OPC UA as a communication interface for PackML. For details, refer to OPC 30050 - UA Companion Specification for Packml 1.01.

The OPC UA PackML Library is a collection of software functional objects for using the OPC UA as the communications protocol for PackML in accordance with the OPC UA PackML specifications.

Intended Audience

This manual is intended for the following personnel,
who must also have knowledge of electrical systems (an electrical engineer or the equivalent).
- Personnel in charge of introducing FA systems.
- Personnel in charge of designing FA systems.
- Personnel in charge of installing and maintaining FA systems.
- Personnel in charge of managing FA systems and facilities.

For programming, this manual is intended for personnel who understand the programming language specifications in international standard IEC 61131-3 or Japanese standard JIS B 3503.

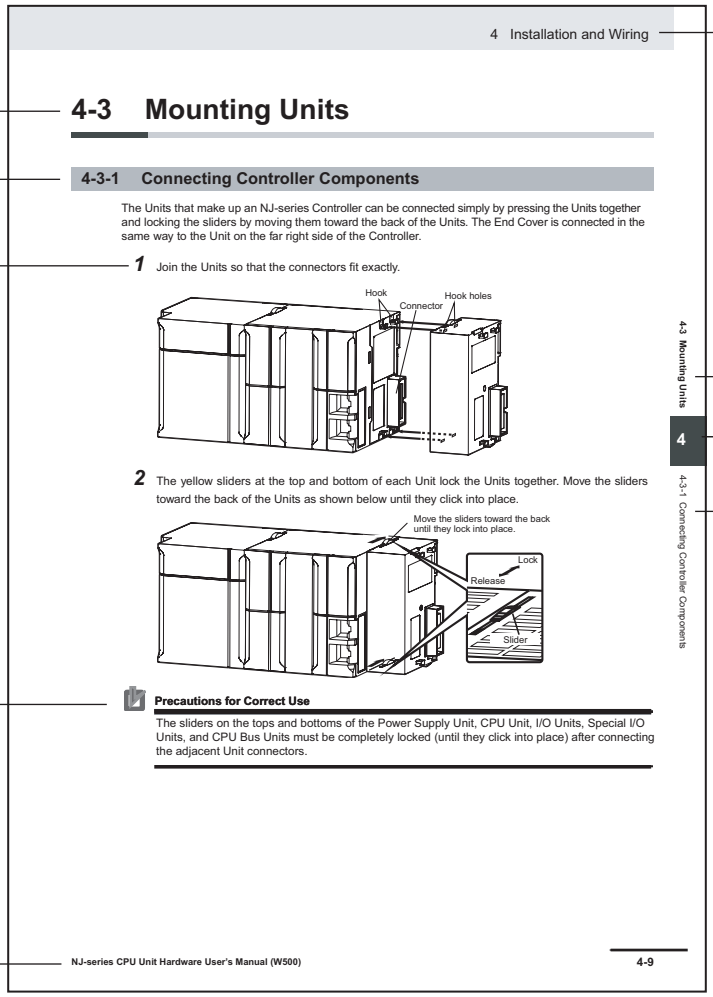
Applicable Products

For the model numbers and versions of an NJ/NX-series CPU Unit and the Sysmac Studio that this library supports, refer to *Sysmac Library Version Information* in the *SYSMAC-XR□□□ Sysmac Library Catalog (Cat. No. P102)*.
You can download the catalog from the OMRON website (https://www.fa.omron.co.jp/products/family/3459/download/catalog.html).

Manual Structure

Page Structure

The following page structure is used in this manual.



Level 1 heading
Level 2 heading
Level 3 heading
Gives the current headings.
Page tab
Gives the number of the main section.

Note This illustration is provided only as a sample. It may not literally appear in this manual.

Special Information

Special information in this manual is classified as follows:

Precautions for Safe Use

Precautions on what to do and what not to do to ensure safe usage of the product.

Precautions for Correct Use

Precautions on what to do and what not to do to ensure proper operation and performance.

Additional Information

Additional information to read as required.
This information is provided to increase understanding and make operation easier.

Version Information

Information on differences in specifications and functionality for CPU Units with different unit versions and for different versions of the industrial-use PC, Sysmac Studio are given.

Sections in this Manual

CONTENTS

Section 3 Common Specifications of FBs

Section 4 Individual Specifications of FBs/FUNs

Index

Terms and Conditions Agreement

Warranty, Limitations of Liability

Warranties

- **Exclusive Warranty**

  Omron's exclusive warranty is that the Products will be free from defects in materials and workmanship for a period of twelve months from the date of sale by Omron (or such other period expressed in writing by Omron). Omron disclaims all other warranties, express or implied.

- **Limitations**

  OMRON MAKES NO WARRANTY OR REPRESENTATION, EXPRESS OR IMPLIED, ABOUT NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE OF THE PRODUCTS. BUYER ACKNOWLEDGES THAT IT ALONE HAS DETERMINED THAT THE PRODUCTS WILL SUITABLY MEET THE REQUIREMENTS OF THEIR INTENDED USE.

  Omron further disclaims all warranties and responsibility of any type for claims or expenses based on infringement by the Products or otherwise of any intellectual property right.

- **Buyer Remedy**

  Omron's sole obligation hereunder shall be, at Omron's election, to (i) replace (in the form originally shipped with Buyer responsible for labor charges for removal or replacement thereof) the non-complying Product, (ii) repair the non-complying Product, or (iii) repay or credit Buyer an amount equal to the purchase price of the non-complying Product; provided that in no event shall Omron be responsible for warranty, repair, indemnity or any other claims or expenses regarding the Products unless Omron's analysis confirms that the Products were properly handled, stored, installed and maintained and not subject to contamination, abuse, misuse or inappropriate modification. Return of any Products by Buyer must be approved in writing by Omron before shipment. Omron Companies shall not be liable for the suitability or unsuitability or the results from the use of Products in combination with any electrical or electronic components, circuits, system assemblies or any other materials or substances or environments. Any advice, recommendations or information given orally or in writing, are not to be construed as an amendment or addition to the above warranty.

  See https://www.omron.com/global/ or contact your Omron representative for published information.

Limitation on Liability; Etc

OMRON COMPANIES SHALL NOT BE LIABLE FOR SPECIAL, INDIRECT, INCIDENTAL, OR CONSEQUENTIAL DAMAGES, LOSS OF PROFITS OR PRODUCTION OR COMMERCIAL LOSS IN ANY

WAY CONNECTED WITH THE PRODUCTS, WHETHER SUCH CLAIM IS BASED IN CONTRACT, WARRANTY, NEGLIGENCE OR STRICT LIABILITY.

Further, in no event shall liability of Omron Companies exceed the individual price of the Product on which liability is asserted.

Application Considerations

Suitability of Use

Omron Companies shall not be responsible for conformity with any standards, codes or regulations which apply to the combination of the Product in the Buyer's application or use of the Product. At Buyer's request, Omron will provide applicable third party certification documents identifying ratings and limitations of use which apply to the Product. This information by itself is not sufficient for a complete determination of the suitability of the Product in combination with the end product, machine, system, or other application or use. Buyer shall be solely responsible for determining appropriateness of the particular Product with respect to Buyer's application, product or system. Buyer shall take application responsibility in all cases.

NEVER USE THE PRODUCT FOR AN APPLICATION INVOLVING SERIOUS RISK TO LIFE OR PROPERTY OR IN LARGE QUANTITIES WITHOUT ENSURING THAT THE SYSTEM AS A WHOLE HAS BEEN DESIGNED TO ADDRESS THE RISKS, AND THAT THE OMRON PRODUCT(S) IS PROPERLY RATED AND INSTALLED FOR THE INTENDED USE WITHIN THE OVERALL EQUIPMENT OR SYSTEM.

Programmable Products

Omron Companies shall not be responsible for the user's programming of a programmable Product, or any consequence thereof.

Disclaimers

Performance Data

Data presented in Omron Company websites, catalogs and other materials is provided as a guide for the user in determining suitability and does not constitute a warranty. It may represent the result of Omron's test conditions, and the user must correlate it to actual application requirements. Actual performance is subject to the Omron's Warranty and Limitations of Liability.

Change in Specifications

Product specifications and accessories may be changed at any time based on improvements and other reasons. It is our practice to change part numbers when published ratings or features are changed, or when significant construction changes are made. However, some specifications of the Product may

be changed without any notice. When in doubt, special part numbers may be assigned to fix or establish key specifications for your application. Please consult with your Omron's representative at any time to confirm actual specifications of purchased Product.

Errors and Omissions

Information presented by Omron Companies has been checked and is believed to be accurate; however, no responsibility is assumed for clerical, typographical or proofreading errors or omissions.

Statement of security responsibilities for assumed use cases and against threats

OMRON SHALL NOT BE RESPONSIBLE AND/OR LIABLE FOR ANY LOSS, DAMAGE, OR EXPENSES DIRECTLY OR INDIRECTLY RESULTING FROM THE INFECTION OF OMRON PRODUCTS, ANY SOFTWARE INSTALLED THEREON OR ANY COMPUTER EQUIPMENT, COMPUTER PROGRAMS, NETWORKS, DATABASES OR OTHER PROPRIETARY MATERIAL CONNECTED THERETO BY DISTRIBUTED DENIAL OF SERVICE ATTACK, COMPUTER VIRUSES, OTHER TECHNOLOGICALLY HARMFUL MATERIAL AND/OR UNAUTHORIZED ACCESS.

It shall be the users sole responsibility to determine and use adequate measures and checkpoints to satisfy the users particular requirements for (i) antivirus protection, (ii) data input and output, (iii) maintaining a means for reconstruction of lost data, (iv) preventing Omron Products and/or software installed thereon from being infected with computer viruses and (v) protecting Omron Products from unauthorized access.

Safety Precautions

Definition of Precautionary Information

The following notation is used in this user's manual to provide precautions required to ensure safe usage of this library on the NJ/NX-series CPU Unit.
The safety precautions that are provided are extremely important for safety. Always read and heed the information provided in all safety precautions.
The following notation is used.

⚠ WARNING	Indicates a potentially hazardous situation which, if not avoided, could result in death or serious injury. Additionally, there may be severe property damage.
⚠ Caution	Indicates a potentially hazardous situation which, if not avoided, may result in minor or moderate injury, or property damage.

Symbols

	The circle and slash symbol indicates operations that you must not do. The specific operation is shown in the circle and explained in text. This example indicates that disassembly is prohibited.
	The triangle symbol indicates precautions (including warnings). The specific operation is shown in the triangle and explained in text. This example indicates a precaution for electric shock.
	The triangle symbol indicates precautions (including warnings). The specific operation is shown in the triangle and explained in text. This example indicates a general precaution.
	The filled circle symbol indicates operations that you must do. The specific operation is shown in the circle and explained in text. This example shows a general precaution for something that you must do.

WARNING

⚠ WARNING

Using this function block (FB) in a device, confirm that the program and FB operates properly. Design a program so that safety measures such as fail-safe circuits are implemented outside of the FB.

Caution

⚠ Caution

Read all related manuals carefully before you use this library.	
Emergency stop circuits, interlock circuits, limit circuits, and similar safety measures must be provided in external control circuits.	
Check the user program, data, and parameter settings for proper execution before you use them for actual operation.	
The Sysmac Library and manuals are assumed to be used by personnel that is given in Intended Audience in this manual. Otherwise, do not use them.	
Perform the test run by holding an emergency stop switch in hand or otherwise prepare for rapid motor operation in an application to control the motor. Also perform the test run by using parameters for which the motor does not rapidly accelerate or decelerate before you gradually adjust the parameters.	
In heating or cooling applications, perform the test run by using parameters for which rapid temperature changes will not occur before you gradually adjust the parameters.	
You must confirm that the user program and parameter values are appropriate to the specifications and operation methods of the devices.	
The sample programming shows only the portion of a program that uses the function or function block from the library.	
When you use actual devices, also use programs such as safety circuits, device interlocks, I/O with other devices, and other control procedures.	
Understand the contents of sample programming before you use the sample programming and create the user program.	
Create a user program that will produce the intended device operation.	
When you adjust the device with this function block incorporated into the device, check the surroundings sufficiently.	

Precautions for Correct Use

Using the Library

- When you use the library, functions or function blocks that are not described in the library manual may be displayed on the Sysmac Studio. Do not use functions or function blocks that are not described in the manual.
- You cannot change the source code of the functions or function blocks that are provided in the Sysmac Library.
- Use this library for the CPU Unit with the unit versions that support OPC UA for PackML.
  For the corresponding model numbers and unit versions of the CPU Unit and versions of the Sysmac Studio, refer to *Sysmac Library Version Information* in the *SYSMAC-XR□□□ Sysmac Library Catalog (Cat. No. P102)*.

Operation

- Specify the input parameter values within the valid range.
- In a function or function block with an Enabled output variable, if the value of Enabled is FALSE, do not use the processing result of the function or function block as a command value to the control target.
- In the function block with Execute, do not perform re-execution by the same instance. The output value of the function block will return to the default value.

Regulations and Standards

Refer to the following manuals for regulations and standards.

- *NX-series CPU Unit Hardware User's Manual (Cat. No. W535)*
- *NX-series NX502 CPU Unit Hardware User's Manual (Cat. No. W629)*
- *NX-series NX102 CPU Unit Hardware User's Manual (Cat. No. W593)*
- *NJ-series CPU Unit Hardware User's Manual (Cat. No. W500)*

Note NX1P2 CPU Units are not applicable to this library.

Related Manuals

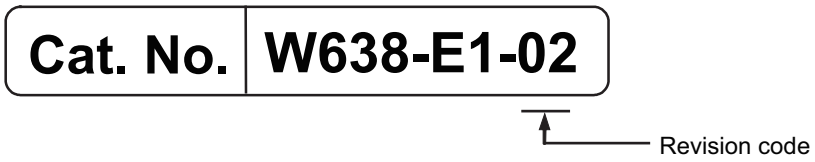
The followings are the manuals related to this manual. Use these manuals for reference.

Manual name	Cat. No.	Model numbers	Application	Description
NJ/NX-series CPU Unit OPC UA User's Manual	W588	NX701-□□□□ NX502-□□□□ NX102-□□□□ NJ501-1□00	Using the OPC UA.	Describes the OPC UA.
NX-series NX102 CPU Unit Hardware User's Manual	W593	NX102-□□□□	Learning the basic specifications of the NX102 CPU Units, including introductory information, designing, installation, and maintenance. Mainly hardware information is provided.	An introduction to the entire NX102 system is provided along with the following information on the CPU Unit. • Features and system configuration • Introduction • Part names and functions • General specifications • Installation and wiring • Maintenance and inspection
NJ/NX-series CPU Unit Software User's Manual	W501	NX701-□□□□ NX502-□□□□ NX102-□□□□ NX1P2-□□□□ NJ501-□□□□ NJ301-□□□□ NJ101-□□□□	Learning how to program and set up an NJ/NX-series CPU Unit. Mainly software information is provided.	The following information is provided on a Controller built with an NJ/NX-series CPU Unit. • CPU Unit operation • CPU Unit features • Initial settings • Programming based on IEC 61131-3 language specifications
NJ/NX-series Instructions Reference Manual	W502	NX701-□□□□ NX502-□□□□ NX102-□□□□ NX1P2-□□□□ NJ501-□□□□ NJ301-□□□□ NJ101-□□□□	Learning detailed specifications on the basic instructions of an NJ/NX-series CPU Unit.	The instructions in the instruction set (IEC 61131-3 specifications) are described.
NJ/NX-series CPU Unit Motion Control User's Manual	W507	NX701-□□□□ NX502-□□□□ NX102-□□□□ NX1P2-□□□□ NJ501-□□□□ NJ301-□□□□ NJ101-□□□□	Learning about motion control settings and programming concepts.	The settings and operation of the CPU Unit and programming concepts for motion control are described.

Manual name	Cat. No.	Model numbers	Application	Description
NJ/NX-series Motion Control Instructions Reference Manual	W508	NX701-□□□□ NX502-□□□□ NX102-□□□□ NX1P2-□□□□ NJ501-□□□□ NJ301-□□□□ NJ101-□□□□	Learning about the specifications of the motion control instructions.	The motion control instructions are described.
NJ/NX-series CPU Unit Built-in EtherNet/IP™ Port User's Manual	W506	NX701-□□□□ NX502-□□□□ NX102-□□□□ NX1P2-□□□□ NJ501-□□□□ NJ301-□□□□ NJ101-□□□□	Using the built-in EtherNet/IP port on an NJ/NX-series CPU Unit.	Information on the built-in EtherNet/IP port is provided. Information is provided on the basic setup, tag data links, and other features.
Sysmac Studio Version 1 Operation Manual	W504	SYSMAC -SE2□□□	Learning about the operating procedures and functions of the Sysmac Studio.	Describes the operating procedures of the Sysmac Studio.

Revision History

A manual revision code appears as a suffix to the catalog number on the front and back covers of the manual.



Revision code	Date	Revised content
01	April 2023	Original production
02	October 2023	Made changes accompanying the addition of NX502-1□00 and NX102-□□□□.

1

Sysmac Library Usage Procedure

This section describes the procedure to use Sysmac Library installed using the installer, and Sysmac Library in the CPU Unit.

1-1 Procedure to Use Sysmac Library Installed Using the Installer

This section describes the procedure to use Sysmac Library installed using the installer.

There are two ways to use libraries.

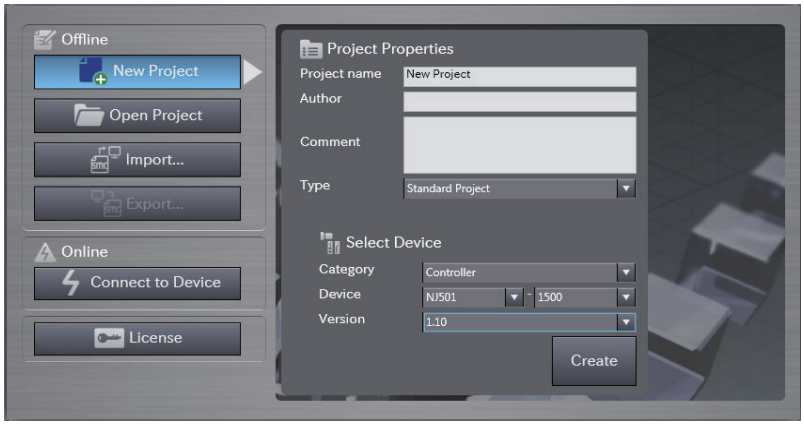
- Using a newly installed Sysmac Library
- Using an upgraded Sysmac Library

Version Information

Refer to *Applicable Products* on page 1 for the models and versions of Controller and Sysmac Studio that can use this library.

1-1-1 Using a Newly Installed Sysmac Library

1 Start the Sysmac Studio and open a project using Sysmac Library, or create a new one.

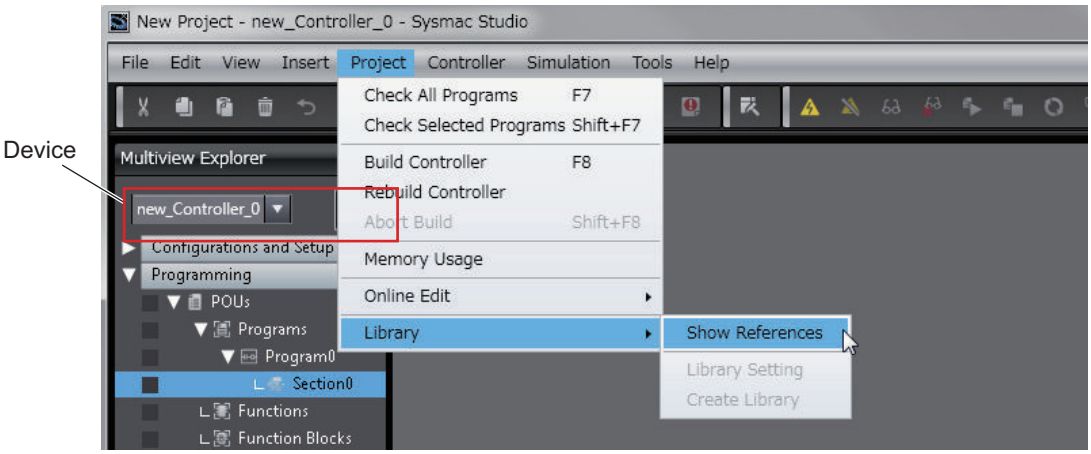


Precautions for Correct Use

If you create a new project, be sure to configure the settings as follows to enable use of the Sysmac Library. Without the settings below, you cannot proceed to Step 2 and later steps.

- Set the project type to Standard Project or Library Project.
- Set the device category to Controller.
- For the Controller and version in device selection, refer to *Applicable Products* on page 1.

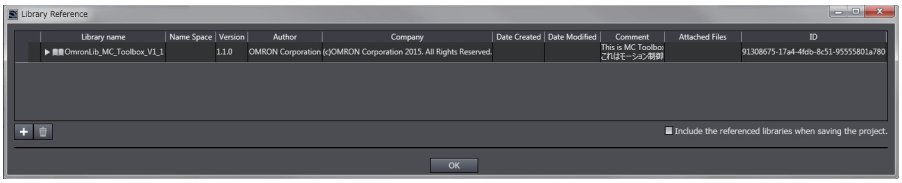
2 Select **Project** - **Library** - **Show References**.



Precautions for Correct Use

If you have more than one device registered in the project, make sure that the currently selected device is the NJ/NX-series CPU Unit. If the NJ/NX-series CPU Unit is not selected, the menu for browsing the library will not appear. When the selected device is the NJ/NX-series CPU Unit, the device icon displayed in Multiview Explorer changes to .

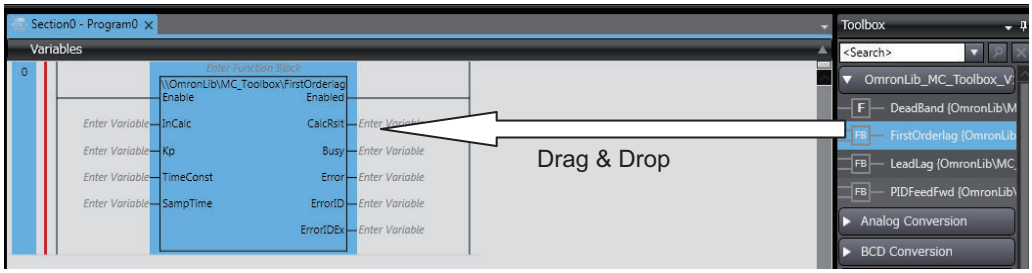
3 Add Sysmac Library to the list and click **OK**.



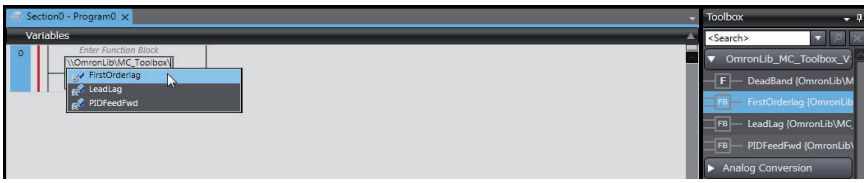
Sysmac Library is read into the project.
Now, when you select the Ladder Editor or ST Editor, the function blocks included in the Sysmac Library appear in the Toolbox.
For the procedure for adding and setting libraries in the above screen, refer to the *Sysmac Studio Version 1 Operation Manual (Cat No. W504)*.

4 Insert the Sysmac Library's function blocks and functions into the circuit using one of the following two methods.

- Select the desired function block in the Toolbox and drag and drop it onto the Ladder Editor.



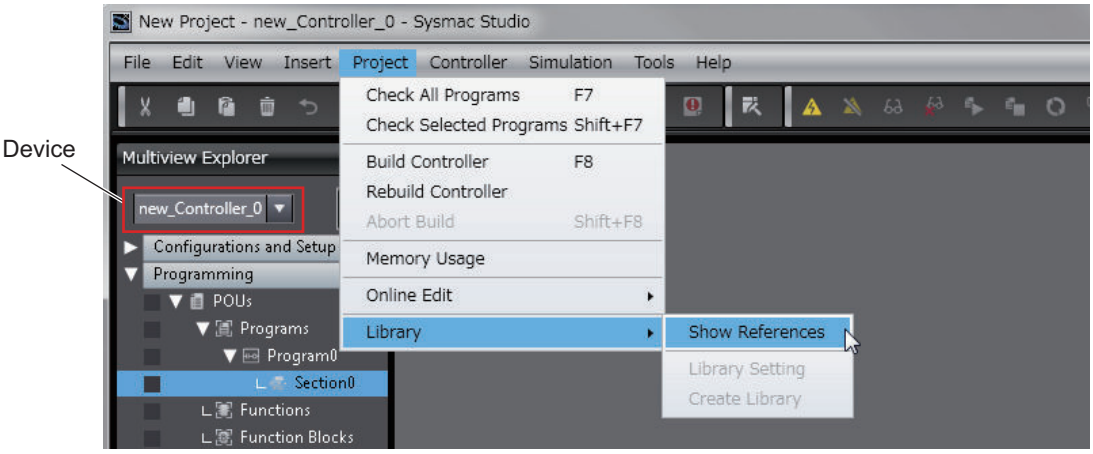
- Right-click the Ladder Editor, select **Insert Function Block** in the menu, and enter the fully qualified name (¥¥namespacename¥FBname).



1-1-2 Using an Upgraded Sysmac Library

1 Start Sysmac Studio and open a project in which any old-version Sysmac Library is included.

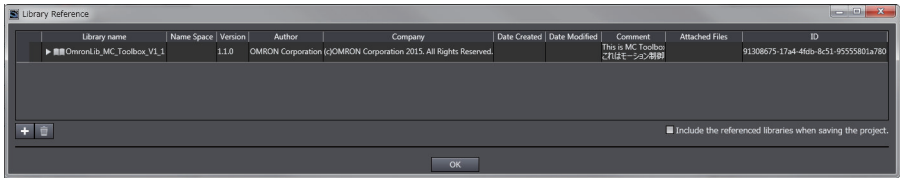
2 Select **Project** - **Library** - **Show References**.



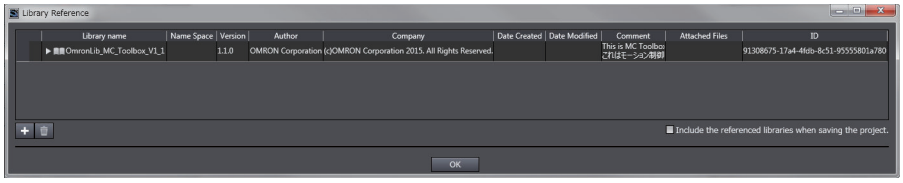
Precautions for Correct Use

If you have more than one device registered in the project, make sure that the currently selected device is the NJ/NX-series CPU Unit. If the NJ/NX-series CPU Unit is not selected, the menu for browsing the library will not appear. When the selected device is the NJ/NX-series CPU Unit, the device icon displayed in Multiview Explorer changes to  .

3 Select an old-version Sysmac Library and click the **Delete Reference** button.



4 Add Sysmac Library to the list and click **OK**.



Precautions for Correct Use

Upgrade the Sysmac Library version, and then execute All Program Check, and confirm that there are no errors in the Build Window Program Check results.
From the Main Menu, select **Project** - **All Program Check**.

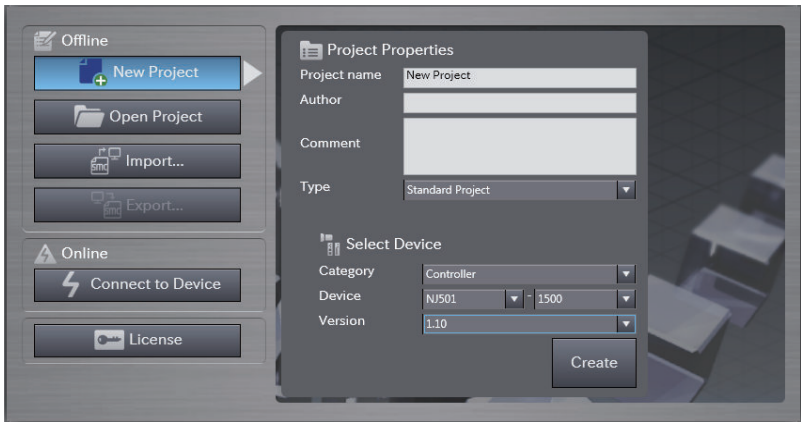
1-2 How to Use Sysmac Library in the CPU Unit

Even when Sysmac Library is not installed on your computer, you can use Sysmac Library by uploading it from the CPU Unit to your computer.
The procedure to use Sysmac Library in the CPU Unit is as follows.

Version Information

Refer to *Applicable Products* on page 1 for the version of Sysmac Studio that can use this library.

1 Start the Sysmac Studio and create a new project in which you want to use Sysmac Library.

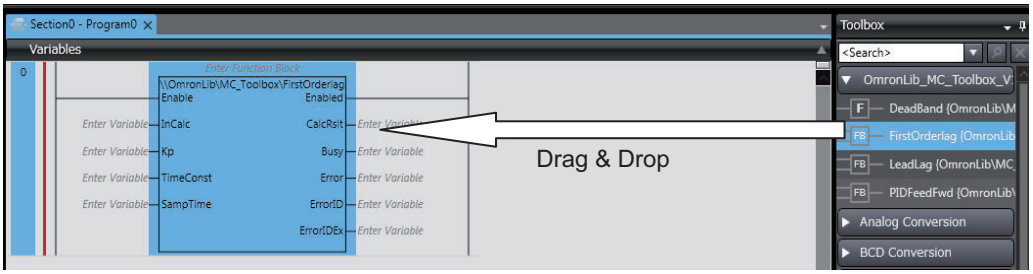


2 Connect the computer to the CPU Unit and place it online.

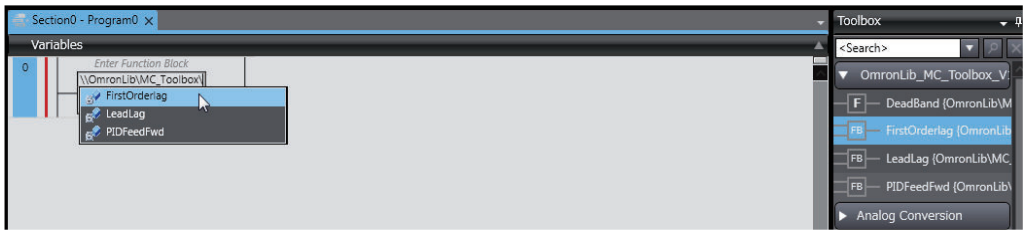
3 Upload the POUs in which Sysmac Library is used.
Now, when you select the Ladder Editor or ST Editor, the function blocks included in the Sysmac Library used in the uploaded POUs appear in the Toolbox.

4 Insert the Sysmac Library's function blocks into the circuit using one of the following two methods.

- Select the desired function block in the Toolbox and drag and drop it onto the Ladder Editor.



- Right-click the Ladder Editor, select **Insert Function Block** in the menu, and enter the fully qualified name (¥¥namespacename¥FBname).



Precautions for Correct Use

- The Sysmac Studio installs Sysmac Library library files to the specified folder on the computer if they are not present. However, the Sysmac Studio does not install libraries to the specified folder on the computer if they are present.
  The specified folder here means the folder in which library files are installed by the installer.
- Note that uploading Sysmac Library from a CPU Unit does not install the manual and help files for the Sysmac Library, unlike installation using the installer. Please install the manual and help files using the installer if you need them.

2

OPC UA PackML Library

This section describes the overview of the functions and usage method.

2-1 Overview

2-1-1 PackML

PackML (Packaging Machine Language) is the standard for packaging machines which was standardized by OMAC (The Organization for Machine Automation and Control). Mode and state of devices, and interface with peripheral devices are defined with the standard. PackML enables the consistent operation even in the production lines using devices of multiple suppliers. Refer to TR88.00.02-2015 for details.

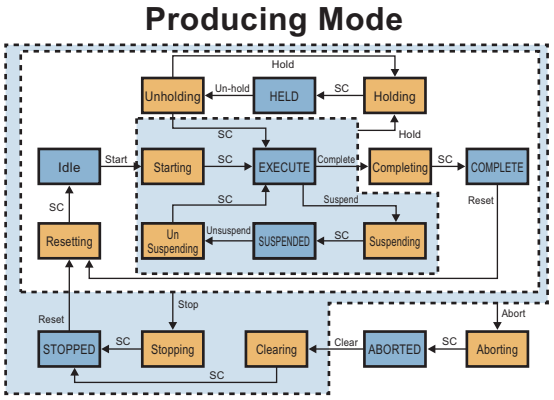
2-1-2 OPC UA for PackML

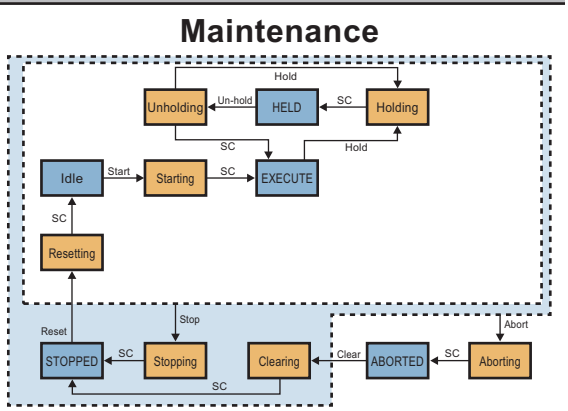
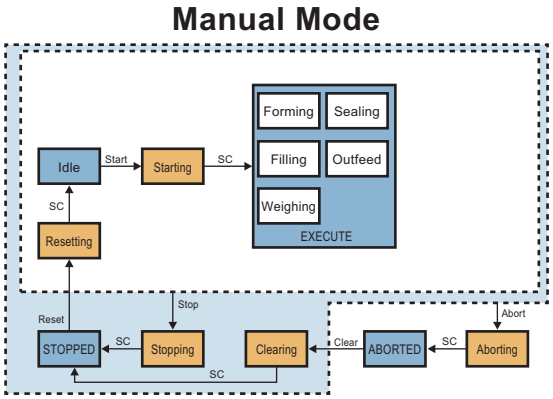
Overview

OPC UA for PackML is the specifications created between OMAC and OPC Foundation, and uses OPC UA as a communication interface for PackML. For details, refer to OPC 30050 - UA Companion Specification for Packml 1.01. (Hereafter referred to as OPC 30050 in this manual.)

Mode

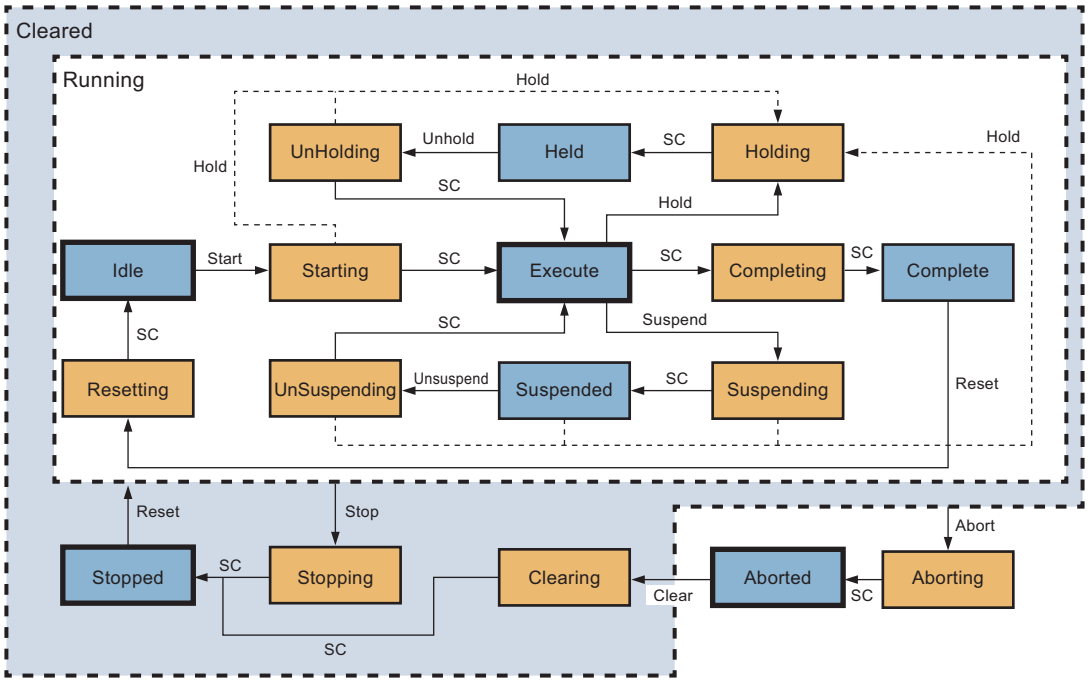
In OPC UA for PackML, the supported modes are defined in enumerations. The enumerator values of each mode are as follows. The additional modes need to be defined as 4 or more.

Mode	Value	Overview	State transition diagram
Produce	1	PackML Production Mode Routine production mode	

Mode	Value	Overview	State transition diagram
Maintenance	2	PackML Maintenance Mode Mode that is executed independently from the other machines in the production line	
Manual	3	PackML Manual Mode Mode that directly controls the individual mechanical elements	

State Transition

Nineteen types of state are defined for OPC UA for PackML. Also, the state transition is defined depending on the operation which is executed in each state. Although the state to use varies depending on the mode, the definitions of each state and state transition are common for each mode.



SC: State Complete

State in thick frame: Mandatory state

----------▶: Optional state transition

The states are roughly classified in two types.

Acting State: State in which some processing is under execution.
In ANSI/ISA-88.00.01, it is the state called transient state and named - *ing.*

Wait State: State in which a series of defined conditions are achieved.
In ANSI/ISA-88.00.01, it is called *final state* or *rest state.*

No.	State name	State type	Description
1	Clearing	Acting state	Transitioned to this state when the Clear command is executed in the Aborted state.
2	Stopped	Wait state	Transitioned to this state by the following methods. • Execute the StateComplete command in the Stopping state • Execute the StateComplete command in the Clearing state
3	Starting	Acting state	Transitioned to this state when the Start command is executed in the Idle state.
4	Idle	Wait state	Transitioned to this state when the StateComplete command is executed in the Resetting state.
5	Suspended	Wait state	Transitioned to this state when the StateComplete command is executed in the Suspending state.

No.	State name	State type	Description
6	Execute	Wait state Acting state	Transitioned to this state by the following methods. • Execute the StateComplete command in the Starting state • Execute the StateComplete command in the Unholding state • Execute the StateComplete command in the Unsuspending state
7	Stopping	Acting state	Transitioned to this state when the Stop command is executed in the Running state.
8	Aborting	Acting state	Transitioned to this state when the Abort command is executed in the Cleared state.
9	Aborted	Wait state	Transitioned to this state when the StateComplete command is executed in the Aborting state.
10	Holding	Acting state	Transitioned to this state when the Hold command is executed in the Execute state. Also, as an optional transition, the transition is made to the Holding state when the Hold command is executed in the Unholding, Starting, Unsuspending, Suspended, or Suspending state.
11	Held	Wait state	Transitioned to this state when the StateComplete command is executed in the Holding state.
12	Unholding	Acting state	Transitioned to this state when the Unhold command is executed in the Held state.
13	Suspending	Acting state	Transitioned to this state when the Suspend command is executed in the Execute state.
14	Unsuspending	Acting state	Transitioned to this state when the Unsuspend command is executed in the Suspended state.
15	Resetting	Acting state	Transitioned to this state when the Reset command is executed in the Stopped state or Complete state.
16	Completing	Acting state	Transitioned to this state when the StateComplete command is executed in the Execute state.
17	Complete	Wait state	Transitioned to this state when the StateComplete command is executed in the Completing state.
18	Cleared	---	Indicate whole state except Aborting and Aborted. Transitioned to this state when the Clear command is executed in the Aborted state.
19	Running	---	Indicate whole state except Aborting, Aborted, Clearing, Stopping, and Stopped. Transitioned to this state when the Reset command is executed in the Stopped state.

Method

Sixteen types of method are defined for OPC UA for PackML. By executing these methods from the OPC UA client, you can set the state transition, mode change, machine speed, and the like.

Method	Input argument	Description of processing
SetUnitMode	Int32 RequestedMode	Change the unit mode to the content that is set with the argument.
SetMachSpeed	Float RequestedMachineSpeed	Change the machine and unit speeds to the value that is set with the argument.

Method	Input argument	Description of processing
SetProduct	PackMLProductDataType[] Product	Change the information related to the product to the content that is specified with the argument.
Abort	---	Executes the transition to the Aborting state when the current state is Cleared.
Clear	---	Executes the transition to the Clearing state when the current state is Aborted.
Stop	---	Executes the transition to the Stopping state when the current state is Running.
Reset	---	Executes the transition to the Resetting state when the current state is Stopped.
ToComplete	---	Executes the transition to the Completing state when the current state is Execute.
Start	PackMLDescriptorDataType[] Parameter	Executes the transition to the Starting state when the current state is Idle. Outputs the value passed with the input argument to the StartMethodParam output variable.
Unhold	---	Executes the transition to the Unholding state when the current state is Held.
Suspend	---	Executes the transition to the Suspending state when the current state is Execute.
Unsuspend	---	Executes the transition to the Unsuspending state when the current state is Suspended.
Hold	---	Executes the transition to the Holding state when the current state is Starting, Suspended, Execute, Unholding, Suspending, or Unsuspending.
RemoteCommand	PackMLRemoteInterfaceDataType[] RemoteInterface	Issue the command to the UA server for passing the information to the other internal system or upstream/downstream system.
SetInterlock	Int32 InterlockId Boolean State	Set the material interlock.
SetParameter	PackMLDescriptorDataType[] Parameter	Set the argument value to the specified parameter.

2-1-3 Node Exposure

Nodes to be exposed to the OPC UA client are shown in the following table. Refer to the OPC UA specifications for details on each node.
In addition, the nodes to be exposed using this library include the nodes with the modeling rule set as Optional because of the OPC UA specifications. Coordinate with the design on the client side for determining whether to handle the data of these nodes in the controller program.

OPC UA node	NodeClass	Availability (R: Readable from client W: Writable from client C: Callable from client No: Not available)
PMLBaseObjType_instance	Object	---
Admin	Object	---
AccTimeSinceReset	Variable	R/W
Alarm	Variable	R/W
AckDateTime	Variable	R/W
Category	Variable	R/W
DateTime	Variable	R/W
ID	Variable	R/W
Message	Variable	R/W
Trigger	Variable	R/W
Value	Variable	R/W
AlarmExtent	Variable	R/W
AlarmHistory	Variable	R/W
AckDateTime	Variable	R/W
Category	Variable	R/W
DateTime	Variable	R/W
ID	Variable	R/W
Message	Variable	R/W
Trigger	Variable	R/W
Value	Variable	R/W
AlarmHistoryExtent	Variable	R/W
MachDesignSpeed	Variable	R/W
ModeCumulativeTime	Variable	R/W
ModeCurrentTime	Variable	R/W
Parameter	Variable	R/W
ID	Variable	R/W
Name	Variable	R/W
Unit	Variable	R/W
Description	Variable	R/W
DisplayName	Variable	R/W
NamespaceUri	Variable	R/W
UnitId	Variable	R/W
Value	Variable	R/W
ProdConsumedCount	Variable	R/W
AccCount	Variable	R/W
Count	Variable	R/W
ID	Variable	R/W
Name	Variable	R/W
Unit	Variable	R/W
Description	Variable	R/W
DisplayName	Variable	R/W
NamespaceUri	Variable	R/W
UnitId	Variable	R/W

OPC UA node	NodeClass	Availability (R: Readable from client W: Writable from client C: Callable from client No: Not available)
ProdDefectiveCount	Variable	R/W
AccCount	Variable	R/W
Count	Variable	R/W
ID	Variable	R/W
Name	Variable	R/W
Unit	Variable	R/W
Description	Variable	R/W
DisplayName	Variable	R/W
NamespaceUri	Variable	R/W
UnitId	Variable	R/W
ProdProcessedCount	Variable	R/W
AccCount	Variable	R/W
Count	Variable	R/W
ID	Variable	R/W
Name	Variable	R/W
Unit	Variable	R/W
Description	Variable	R/W
DisplayName	Variable	R/W
NamespaceUri	Variable	R/W
UnitId	Variable	R/W
StateCumulativeTime	Variable	R/W
StateCurrentTime	Variable	R/W
StopReason	Variable	R/W
AckDateTime	Variable	R/W
Category	Variable	R/W
DateTime	Variable	R/W
ID	Variable	R/W
Message	Variable	R/W
Trigger	Variable	R/W
Value	Variable	R/W
StopReasonExtent	Variable	R/W
Warning	Variable	R/W
AckDateTime	Variable	R/W
Category	Variable	R/W
DateTime	Variable	R/W
ID	Variable	R/W
Message	Variable	R/W
Trigger	Variable	R/W
Value	Variable	R/W
WarningExtent	Variable	R/W
BaseStateMachine	Object	---
Abort	Method	C
AvailableStates	Variable	R

OPC UA node	NodeClass	Availability (R: Readable from client W: Writable from client C: Callable from client No: Not available)
AvailableTransitions	Variable	R
Clear	Method	C
CurrentState	Variable	R/W
EffectiveDisplayName	Variable	R/W
Id	Variable	R
Name	Variable	R
Number	Variable	R/W
LastTransition	Variable	R/W
EffectiveTransitionTime	Variable	R/W
Id	Variable	R
Name	Variable	R
Number	Variable	R/W
TransitionTime	Variable	R/W
MachineState	Object	---
AvailableStates	Variable	R
AvailableTransitions	Variable	R
CurrentState	Variable	R/W
EffectiveDisplayName	Variable	R/W
Id	Variable	R
Name	Variable	R
Number	Variable	R/W
ExecuteState	Object	---
AvailableStates	Variable	R
AvailableTransitions	Variable	R
CurrentState	Variable	R/W
EffectiveDisplayName	Variable	R/W
Id	Variable	R
Name	Variable	R
Number	Variable	R/W
Hold	Method	C
LastTransition	Variable	R/W
EffectiveTransitionTime	Variable	R/W
Id	Variable	R
Name	Variable	R
Number	Variable	R/W
TransitionTime	Variable	R/W
Reset	Method	C
Start	Method	C
InputArguments	Variable	R
Suspend	Method	C
ToComplete	Method	C
Unhold	Method	C
Unsuspend	Method	C

OPC UA node	NodeClass	Availability (R: Readable from client W: Writable from client C: Callable from client No: Not available)
LastTransition	Variable	R/W
EffectiveTransitionTime	Variable	R/W
Id	Variable	R
Name	Variable	R
Number	Variable	R/W
TransitionTime	Variable	R/W
Reset	Method	C
Stop	Method	C
Enabled*1	Variable	R
PackMLVersion	Variable	R
RemoteCommand	Method	No
InputArguments	Variable	No
SetInterlock	Method	C
InputArguments	Variable	R
SetMachSpeed	Method	C
InputArguments	Variable	R
SetParameter	Method	C
InputArguments	Variable	R
SetProduct	Method	C
InputArguments	Variable	R
SetUnitMode	Method	C
InputArguments	Variable	R
Status	Object	---
CurMachSpeed	Variable	R/W
Definition	Variable	R/W
EURange	Variable	R/W
EngineeringUnits	Variable	R/W
InstrumentRange	Variable	R/W
ValuePrecision	Variable	R/W
EquipmentBlocked	Variable	R/W
EquipmentStarved	Variable	R/W
MachSpeed	Variable	R/W
Definition	Variable	R/W
EURange	Variable	R/W
EngineeringUnits	Variable	R/W
InstrumentRange	Variable	R/W
ValuePrecision	Variable	R/W
MaterialInterlock	Variable	R/W
MaterialInterlocked	Variable	R/W
Parameter	Variable	R/W
ID	Variable	R/W
Name	Variable	R/W
Unit	Variable	R/W

OPC UA node	NodeClass	Availability (R: Readable from client W: Writable from client C: Callable from client No: Not available)
Description	Variable	R/W
DisplayName	Variable	R/W
NamespaceUri	Variable	R/W
UnitId	Variable	R/W
Value	Variable	R/W
Product	Variable	R/W
Ingredients	Variable	R/W
IngredientsID	Variable	R/W
Parameter	Variable	R/W
ID	Variable	R/W
Name	Variable	R/W
Unit	Variable	R/W
Description	Variable	R/W
DisplayName	Variable	R/W
NamespaceUri	Variable	R/W
UnitId	Variable	R/W
Value	Variable	R/W
ProcessVariables	Variable	R/W
ID	Variable	R/W
Name	Variable	R/W
Unit	Variable	R/W
Description	Variable	R/W
DisplayName	Variable	R/W
NamespaceUri	Variable	R/W
UnitId	Variable	R/W
Value	Variable	R/W
ProductID	Variable	R/W
RemoteParameter	Variable	No
CmdValue	Variable	No
ControlCmdNumber	Variable	No
Number	Variable	No
Parameter	Variable	No
ID	Variable	No
Name	Variable	No
Unit	Variable	No
Description	Variable	No
DisplayName	Variable	No
NamespaceUri	Variable	No
UnitId	Variable	No
Value	Variable	No
StateChangeInProcess	Variable	R/W
StateRequested	Variable	R/W
UnitModeChangeInProcess	Variable	R/W

OPC UA node	NodeClass	Availability (R: Readable from client W: Writable from client C: Callable from client No: Not available)
UnitModeCurrent	Variable	R/W
UnitModeRequested	Variable	R/W
UnitSupportedModes	Variable	R
TagID	Variable	R

*1. The Enabled node is the node Omron added, not the one defined in OPC UA for PackML. For PMLBaseObjType in this library, only while Enabled is TRUE, the value of each OPC UA node is valid, and a method can be called. This value is exposed as the Enabled node so that OPC UA clients can see the value of Enabled for a PMLBaseObjType instance.

2-2 Usage Method

- This library supports OPC UA for PackML and provides the following functions.
  1. Node exposing function that supports OPC UA for Pack ML
  2. Method execution function from the OPC UA client
  3. PackML mode/state control function (including the Dwell Time Measure function)
  4. Alarm management function
- For how to expose nodes to OPC UA communications using this library, refer to *Variables Based on Companion Specifications Published to OPC UA Communications* in the *NJ/NX-series CPU Unit OPC UA User's Manual (Cat. No. W588)*.
- This library includes a PackML support function for the SYSMAC-XR012 Packaging Machine Library.
  Devices that traditionally use SYSMAC-XR012 to perform the PackML function can easily support OPC UA PackML by replacing SYSMAC-XR012 with this library. Delete the reference of the OmronLib_PackML30_Vx_x.slr library file that supports the PackML function for SYSMAC-XR012 before setting the reference to this library.
  Replace PackMLModeStateMachine and PackMLModeStateTimer that are provided in the PackML support function for the SYSMAC-XR012 Packaging Machine Library with PMLBaseObjType. The assignment of in-out variables when replacing is shown in the following table.

In-Out Variable Replacement Table from PackMLModeStateMachine and PackMLModeStateTimer to PMLBaseObjType

OmronLib_PackML30_Vx_x.slr			OmronLib_PackML_OPCUAxx_Vy_y.slr		
Variable name	**Variable attribute**	**Data type**	**Variable name**	**Variable attribute**	**Data type**
PackMLModeStateMachine			PMLBaseObjType		
Cfg_DisabledStates	**In-out**	ARRAY[1..31] of OmronLib \PackML30\sPACKML_STATES_FLAG	Cfg_DisabledStates	**Input**	ARRAY[1..31] of OmronLib \PackML30\sPACKML_STATES_FLAG
ModeSwitchableStates	**In-out**	ARRAY[1..31] of OmronLib \PackML30\sPACKML_STATES_FLAG	**Cfg_ModeSwitchableStates**	**Input**	ARRAY[1..31] of OmronLib \PackML30\sPACKML_STATES_FLAG
Cmd_ModeSwitch	Input	DINT	**ModeNum Cmd_ModeChange**	Input	DINT
Cmd_StateTransition	Input	OmronLib \PackML30\sPACKML_TRANSITION_COMMAND	Cmd_StateTransition	Input	OmronLib \PackML30\sPACKML_TRANSITION_COMMAND

OmronLib_PackML30_Vx_x.slr			OmronLib_PackML_OPCUAxx_Vy_y.slr		
Variable name	Variable attribute	Data type	Variable name	Variable attribute	Data type
Cfg_DisabledStatesActual	Output	ARRAY[1..31] of OmronLib \PackML30\sPACKML_STATES_FLAG	Cfg_DisabledStatesActual	Output	ARRAY[1..31] of OmronLib \PackML30\sPACKML_STATES_FLAG
ModeChangeNotAllowed	Output	BOOL	ModeChangeNotAllowed	Output	BOOL
ModeCurrent	Output	DINT	ModeCurrent	Output	DINT
ModeRequested	Output	DINT	ModeRequested	Output	DINT
ModeChangeInProcess	**Output**	BOOL	**Status.UnitModeChangeInProcess**	**In-out**	BOOL
StateCurrent	Output	DINT	StateCurrent	Output	DINT
StateRequested	**Output**	DINT	**Status.StateRequested**	**In-out**	DINT
StateChangeInProcess	**Output**	BOOL	**Status.StateChangeInProcess**	**In-out**	BOOL
PackMLModeStateTimer			---	---	---
Cmd_ResetAllDwellTimes	Input	BOOL	Cmd_ResetAllDwellTimes	Input	BOOL
Sts_ModeCurrentDwellSeconds*1	In-out	**ARRAY[1..31] of DINT**	**Admin.ModeCurrentTime**	In-out	**ARRAY[0..31] of DINT**
Sts_ModeCumulativeDwellSeconds*1	In-out	**ARRAY[1..31] of DINT**	**Admin.ModeCumulativeTime**	In-out	**ARRAY[0..31] of DINT**
Sts_StateCurrentDwellTimes*1	In-out	**ARRAY[1..31,1..17] of DINT**	**Admin.StateCurrentTime**	In-out	**ARRAY[0..31,0..19] of DINT**
Sts_StateCumulativeDwellTimes*1	In-out	**ARRAY[1..31,1..17] of DINT**	**Admin.StateCumulativeTime**	In-out	**ARRAY[0..31,0..19] of DINT**
Sts_AccTimeSinceReset	**Output**	DINT	**Admin.AccTimeSinceReset**	**In-out**	DINT

*1. Extend the range of the array when replacing.

3

Common Specifications of FBs

This section describes the shared specifications of each function block in the Sysmac Library.

3-1 Common Variables

This section describes the specifications of variables (EN, Execute, Enable, Abort, ENO, Done, CalcRslt, Enabled, Busy, CommandAborted, Error, ErrorID, and ErrorIDEx) that are used for more than one function or function block. The specifications are described separately for functions, for execute-type function blocks, and for enable-type function blocks.

3-1-1 Definition of Input Variables and Output Variables

Common input variables and output variables used in functions and function blocks are as follows.

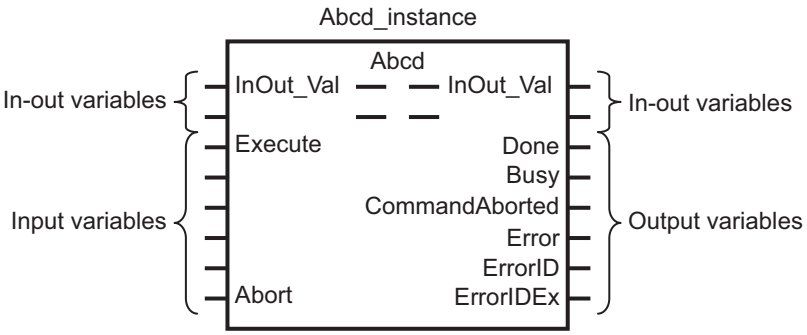
Variable	I/O	Data type	Function/function block type to use			Meaning	Definition
			Function block		Function		
			Execute-type	Enable-type			
EN	Input	BOOL			OK	Execute	The processing is executed while the variable is TRUE.
Execute		BOOL	OK			Execute	The processing is executed when the variable changes to TRUE.
Enable		BOOL		OK		Run	The processing is executed while the variable is TRUE.
Abort		BOOL	OK			Abort	The processing is aborted. You can select the aborting method.

Variable	I/O	Data type	Function/function block type to use			Meaning	Definition
			Function block		Function		
			Execute-type	Enable-type			
ENO	Output	BOOL			OK	Done	The variable changes to TRUE when the processing ends normally. It is FALSE when the processing ends in an error, the processing is in progress, or the execution condition is not met.
Done		BOOL	OK			Done	The variable changes to TRUE when the processing ends normally. It is FALSE when the processing ends in an error, the processing is in progress, or the execution condition is not met.
Busy		BOOL	OK	OK		Executing	The variable is TRUE when the processing is in progress. It is FALSE when the processing is not in progress.
CalcRslt		LREAL		OK		Calculation Result	The calculation result is output.
Enabled		BOOL		OK		Enabled	The variable is TRUE when the output is enabled. It is used to calculate the control amount for motion control, temperature control, etc.
Command Aborted		BOOL	OK			Command Aborted	The variable changes to TRUE when the processing is aborted. It changes to FALSE when the processing is re-executed the next time.
Error		BOOL	OK	OK		Error	This variable is TRUE while there is an error. It is FALSE when the processing ends normally, the processing is in progress, or the execution condition is not met.
ErrorID		WORD	OK	OK		Error Code	An error code is output.
ErrorIDEx		DWORD	OK	OK		Expansion Error Code	An expansion error code is output.

3-1-2 Execute-type Function Blocks

- Processing starts when Execute changes to TRUE.

- When Execute changes to TRUE, Busy also changes to TRUE. When processing is completed normally, Busy changes to FALSE and Done changes to TRUE.
- When continuously executes the function blocks of the same instance, change the next Execute to TRUE for at least one task period after Done changes to FALSE in the previous execution.
- If the function block has a CommandAborted (Instruction Aborted) output variable and processing is aborted, CommandAborted changes to TRUE and Busy changes to FALSE.
- If an error occurs in the function block, Error changes to TRUE and Busy changes to FALSE.
- For function blocks that output the result of calculation for motion control and temperature control, you can use the BOOL input variable Abort to abort the processing of a function block. When Abort changes to TRUE, CommandAborted changes to TRUE and the execution of the function block is aborted.

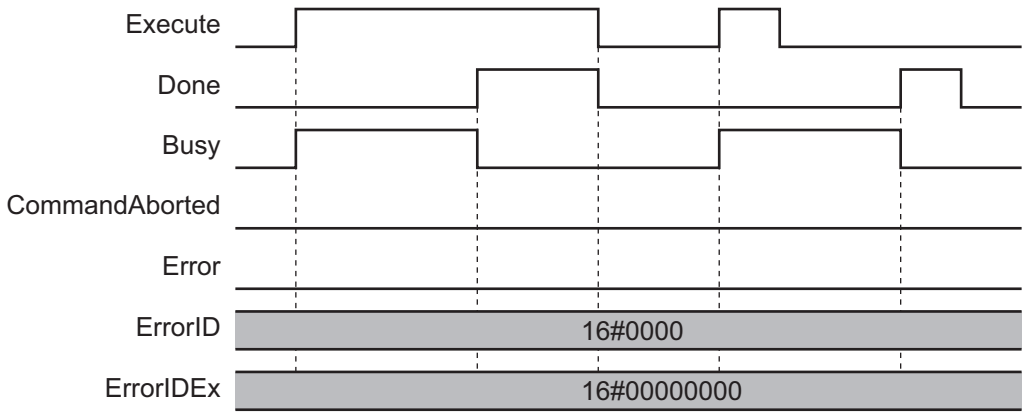


- If Execute is TRUE and Done, CommandAborted, or Error changes to TRUE, Done, CommandAborted, and Error changes to FALSE when Execute is changed to FALSE.
- If Execute is FALSE and Done, CommandAborted, or Error changes to TRUE, Done, CommandAborted, and Error changes to TRUE for only one task period.
- If an error occurs in a function block, the relevant error code and expansion error code are set in ErrorID (Error Code) and ErrorIDEx (Expansion Error Code). The error codes are retained even after Error changes to FALSE, but ErrorID is set to 16#0000 and ErrorIDEx is set to 16#0000 0000 when Execute changes to TRUE.

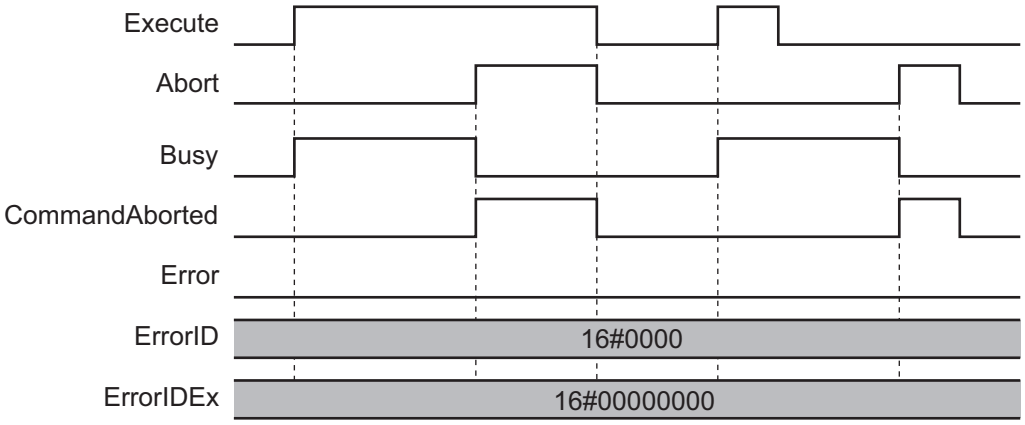
Timing Charts

This section provides timing charts for a normal end, canceled execution, aborted execution, and errors.

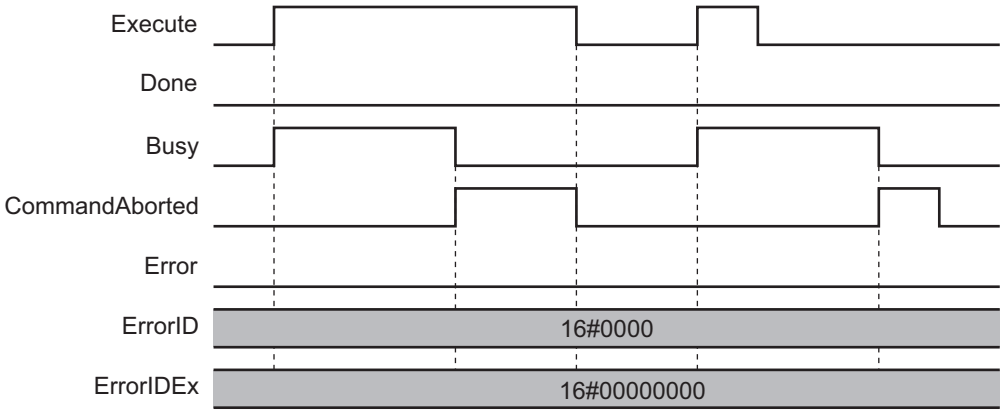
● Normal End



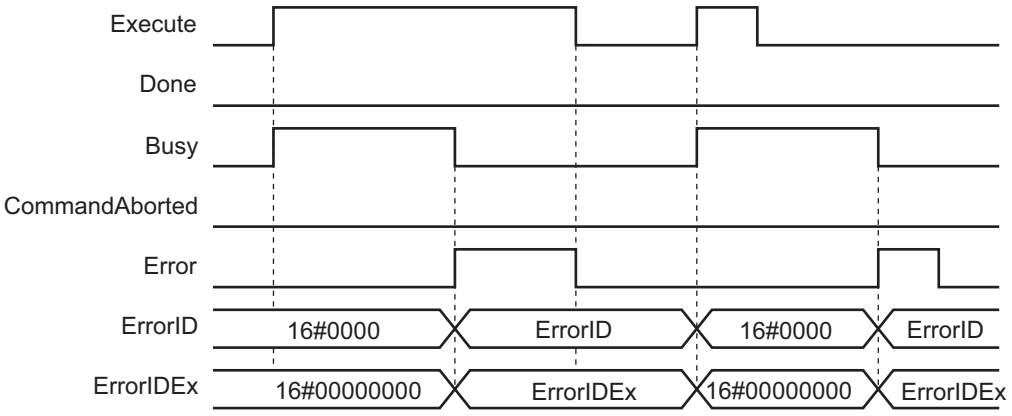
● Canceled Execution



● Aborted Execution



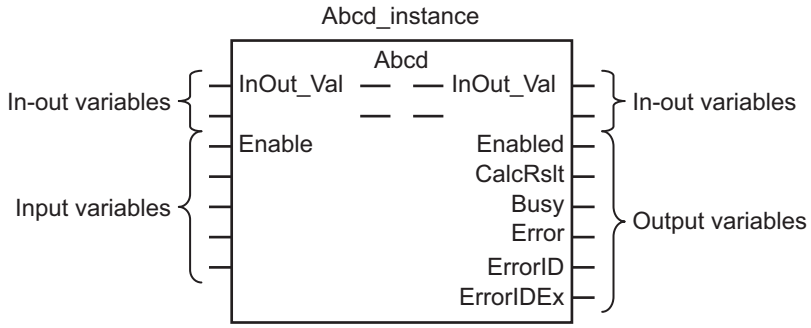
● Errors



3-1-3 Enable-type Function Blocks

- Processing is executed while Enable is TRUE.
- When Enable changes to TRUE, Busy also changes to TRUE. Enabled is TRUE during calculation of the output value.

- If an error occurs in the function block, Error changes to TRUE and Busy and Enabled change to FALSE. When Enable changes to FALSE, Enabled, Busy, and Error change to FALSE.

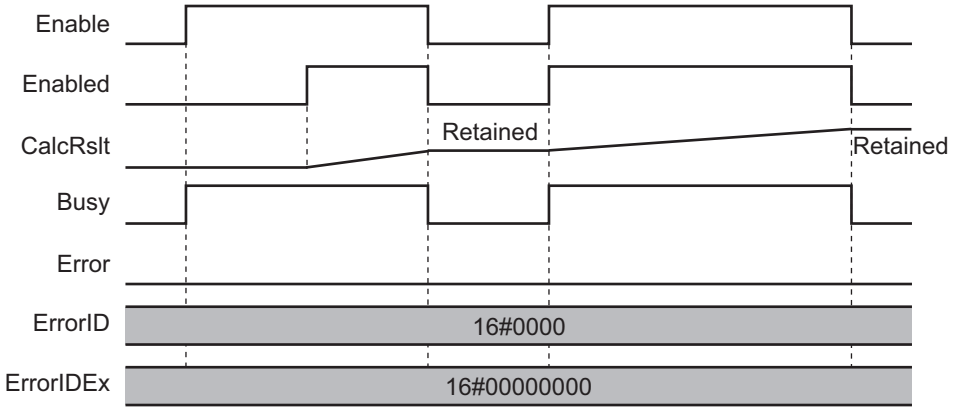


- If an error occurs in a function block, the relevant error code and expansion error code are set in ErrorID (Error Code) and ErrorIDEx (Expansion Error Code). The error codes are retained even after Error changes to FALSE, but ErrorID is set to 16#0000 and ErrorIDEx is set to 16#0000 0000 when Enable changes to TRUE.
- For function blocks that calculate the control amount for motion control, temperature control, etc., Enabled is FALSE when the value of CalcRslt (Calculation Result) is incorrect. In such a case, do not use CalcRslt. In addition, after the function block ends normally or after an error occurs, the value of CalcRslt is retained until Enable changes to TRUE. The control amount will be calculated based on the retained CalcRslt value, if it is the same instance of the function block that changed Enable to TRUE. If it is a different instance of the function block, the control amount will be calculated based on the initial value.

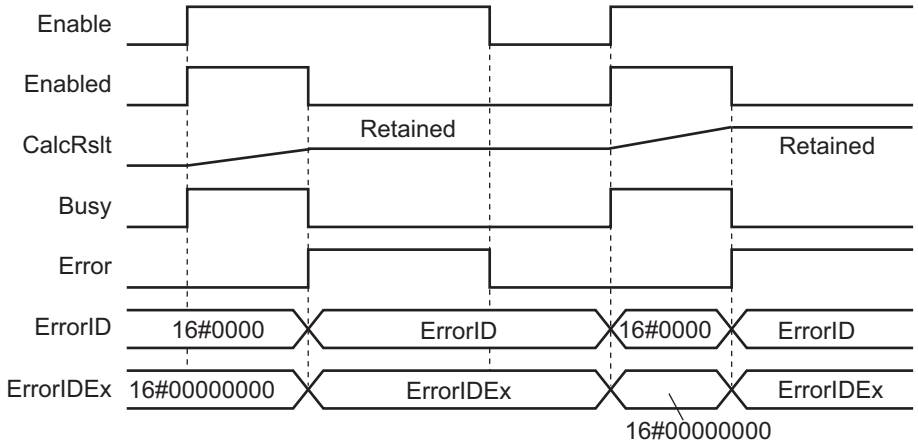
Timing Charts

This section provides timing charts for a normal end and errors.

● Normal End



● Errors


Enable
Enabled
CalcRslt
Retained
Retained
Busy
Error
ErrorID
16#0000
ErrorID
16#0000
ErrorID
ErrorIDEx
16#00000000
ErrorIDEx
ErrorIDEx
16#00000000

3-2 Precautions

This section provides precautions for the use of this function block.

3-2-1 Nesting

You can nest calls to this function block for up to four levels.
For details on nesting, refer to the *NJ/NX-series CPU Unit Software User's Manual (Cat. No. W501)* or the *NY-series IPC Machine Controller Industrial Panel PC / Industrial Box PC Software User's Manual (Cat. No. W558)*.

3-2-2 Instruction Options

You cannot use the upward differentiation option for this function block.

3-2-3 Re-execution of Function Blocks

Execute-type function blocks cannot be re-executed by the same instance.
If you do so, the output value will be the initial value.
For details on re-execution, refer to the *NJ/NX-series CPU Unit Motion Control User's Manual (Cat. No. W507)* or the *NY-series IPC Machine Controller Industrial Panel PC / Industrial Box PC Motion Control User's Manual (Cat. No. W559)*.

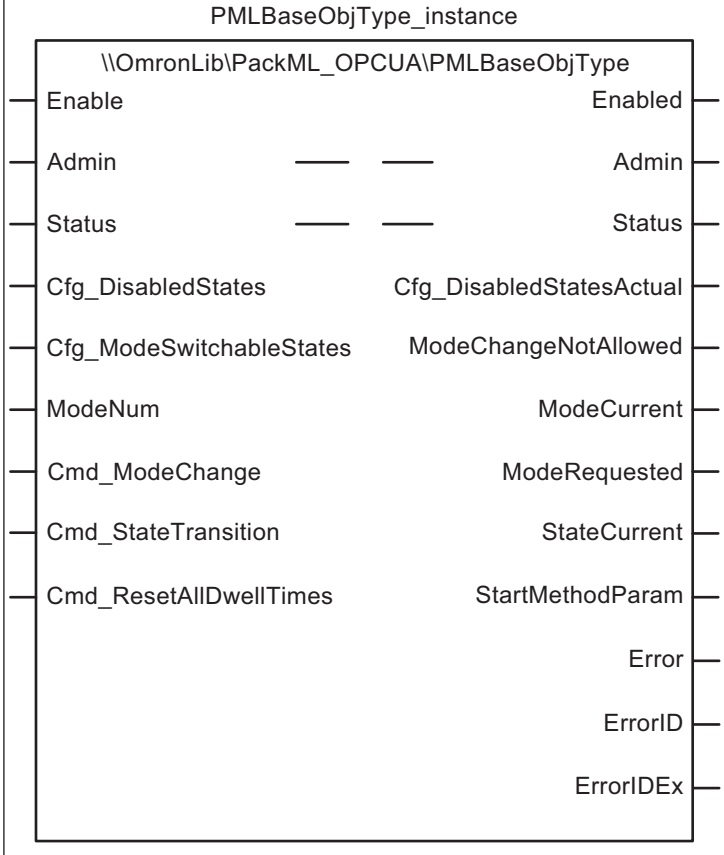
4

Individual Specifications of FBs/ FUNs

This section describes the individual specifications for each function and function block in the OPC UA PackML library.

PMLBaseObjType

This is a function-block representation of PackMLBaseObjectType defined in OPC 30050. Instantiate this function block, and publish the namespace OmronLib\PackML_OPCUA to a network to expose the information necessary for the PackML system to the OPC UA node. Then, publishing the namespace OmronLib\PackML_OPCUA\methods to a network enables the execution of methods from the OPC UA client. When a method is executed from the OPC UA client, state transition and mode change, or parameter, production information, machine-speed, and interlock settings are performed accordingly. In addition, the value of each OPC UA node is set according to the input of the function block and the execution of methods.

FB/FUN name	Name	FB/FUN	Graphic expression	ST expression
PMLBaseObjType	PackML Base Object Type	FB		PMLBaseObjType_instance(Enable:=, Admin:=, Status:=, Cfg_DisabledStates:=, Cfg_ModeSwitchableStates:=, ModeNum:=, Cmd_ModeChange:=, Cmd_StateTransition:=, Cmd_ResetAllDwellTimes:=, Enabled=>, Cfg_DisabledStatesActual=>, ModeChangeNotAllowed=>, ModeCurrent=>, ModeRequested=>, StateCurrent=>, StartMethodParam=>, Error=>, ErrorID=>, ErrorIDEx=>);

Library Information

Item	Description
Library file name	OmronLib_PackML_OPCUAxx_Vy_y.slr (x indicates the version of the OPC UA for PackML specifications (example: in the case of 1.0.1, the upper two digits are used to indicate 10), and y indicates the version of the library)
Namespace	OmronLib\PackML_OPCUA

Item	Description
Function block and function number	00243
Publish/Do not publish source code	Not published.

Input Variables

Variable	Name	Data type	Description	Valid range	Unit	Default
Enable	Run	BOOL	TRUE: The processing is executed. FALSE: The processing stops.	TRUE or FALSE	---	FALSE
Cfg_DisabledStates	Disabled-state Settings	ARRAY[1..31] of OmronLib\PackML30\sPACKML_STATES_FLAG	The unused states are specified per mode. The array index represents the mode number. TRUE is set for the unused state.	TRUE or FALSE	---	All FALSE
Cfg_ModeSwitchableStates	Mode Switch Permit Settings	ARRAY[1..31] of OmronLib\PackML30\sPACKML_STATES_FLAG	The states where mode change is permitted are specified per mode. The array index represents the mode number. TRUE is set for the state where mode change is permitted.	TRUE or FALSE	---	All FALSE
ModeNum	Mode Number	DINT	The mode number is specified in order to change the mode. The specified mode change is executed when the Cmd_ModeChange input variable is changed to TRUE.	1 to 31	---	0
Cmd_ModeChange	Mode Change Command	BOOL	When the variable changes to TRUE, the mode change specified in ModeNum is executed. The input does not change modes when the same number as the current mode or the value out of the valid range is specified for ModeNum.	TRUE or FALSE	---	FALSE

Variable	Name	Data type	Description	Valid range	Unit	Default
Cmd_StateTransition	Transition Command	OmronLib \PackML30\sPACKML_TRANSITION_COMMAND	The transition request to the state machine is specified. This input and execution of the state transition method from the OPC UA client allow more than one transitions to be specified. However, only an executable transition for the current state is executed. When more than one transitions are executable for the current state, one transition is executed according to the existing priority. When the setting is made as all FALSE, the state transition does not occur and the current state is retained.	TRUE or FALSE	---	All FALSE
Cmd_ResetAllDwellTimes	Accumulated Time Reset Command	BOOL	When this variable is TRUE, accumulated dwell seconds in each mode and state are reset to 0.	TRUE or FALSE	---	FALSE

Output Variables

Variable	Name	Data type	Description	Valid range	Unit	Default
Enabled	Enabled	BOOL	TRUE: The output and OPC UA node are enabled, and the method execution is enabled. FALSE: The output and OPC UA node are disabled, and the method execution is disabled.	TRUE or FALSE	---	---
Cfg_DisabledStatesActual	Actual Disabled-state Settings	ARRAY[1..31] of OmronLib \PackML30\sPACKML_STATES_FLAG	The disabled-state settings used in FB are output.	TRUE or FALSE	---	---
ModeChangeNotAllowed	Mode Change Prohibited Flag	BOOL	The flag becomes TRUE when the set condition to switch the mode is not met although the valid-mode-change command is entered to the ModeNum input variable. Refer to <Permit Mode Switch Setting> for details.	TRUE or FALSE	---	---

Variable	Name	Data type	Description	Valid range	Unit	Default
ModeCurrent	Current Mode Number	DINT	The current mode number is output. When Enable changes to TRUE, the mode 1 (ProductionMode) is output.	1 to 31	---	---
ModeRequested	Mode Change Request Value	DINT	The mode number currently requested to switch is output.	1 to 31	---	---
StateCurrent	Current State Number	DINT	The current state number is output. When Enable changes to TRUE, the state 2 (Stopped) is output.	1 to 17	---	---
StartMethod_Param	Start Method Parameter	ARRAY[0..15] of OmronLib \PackML_OPCUA \sPackML_DescriptorDataType	The value of input argument parameter for executing the start method is output.	---	---	---
Error	Error	BOOL	TRUE: Error end. FALSE: Normal end, execution in progress, or execution condition not met. There is no error in this function block. FALSE is always output.	FALSE*1	---	---
ErrorID	Error Code	WORD	This is the error ID for an error end. The value is 16#0 for a normal end. There is no error in this function block. 16#0 is always output.	*1	---	---
ErrorIDEx	Expansion Error Code	DWORD	This is the error ID for an error end. The value is 16#0 for a normal end. There is no error in this function block. 16#0 is always output.	*1	---	---

*1. Refer to *Troubleshooting* on page 4-20 for details.

In-Out Variables

Variable	Name	Data type	Description	Valid range	Unit	Default
Admin	Management Information	OmronLib \PackML_OPCUA \sPackML_AdminObject-Type	Variable that supports Admin of the OPC UA node. Refer to OPC 30050 for details on each node.	---	---	---
Status	Status Information	OmronLib \PackML_OPCUA \sPackML_StatusObject-Type	Variable that supports Status of the OPC UA node. Refer to OPC 30050 for details on each node.	---	---	---

Structures

● OmronLib\PackML_OPCUA\sPackML_AdminObjectType

The structure of the management information required for the PackML OPC UA server.

Member	Member name	Data type	Valid range	Unit	Description
Parameter	Parameter	ARRAY[0..15] of OmronLib\PackML_OPCUA \sPackML_DescriptorDataType	---	---	Admin.Parameter Parameter related to the local interface. It is used for indicating the downtime parameter for quality, alarm, and machine.
Alarm	Alarm	ARRAY[0..15] of OmronLib\PackML_OPCUA \sPackML_AlarmDataType	---	---	Admin.Alarm Alarm event related to the local interface. It is used for indicating the occurring alarm and machine downtime factor.
AlarmExtent	Maximum Number of Alarms	DINT	0 to 15	---	Admin.AlarmExtent Maximum number of available alarms. If a value outside the effective range is input, the input is ignored and 15 is set.
AlarmHistory	Alarm History	ARRAY[0..31] of OmronLib\PackML_OPCUA \sPackML_AlarmDataType	---	---	Admin.AlarmHistory Alarm history. It is used for indicating the alarm history and machine downtime factor.
AlarmHistoryExtent	Maximum Number of Alarm Histories	DINT	0 to 31	---	Admin.AlarmHistoryExtent Maximum number of available alarm histories. If a value outside the effective range is input, the input is ignored and 31 is set.

Member	Member name	Data type	Valid range	Unit	Description
Warning	Warning	ARRAY[0..31] of OmronLib\PackML_OPCUA\sPackML_AlarmDataType	---	---	Admin.Warning Warning. Event in which the machine does not need to be stopped, but the stop may be imminent.
WarningExtent	Maximum Number of Warnings	DINT	0 to 31	---	Admin.WarningExtent Maximum number of available warnings. If a value outside the effective range is input, the input is ignored and 31 is set.
StopReason	Stop Reason	OmronLib\PackML_OPCUA\sPackML_AlarmDataType	---	---	Admin.StopReason Stop reason. Event that is first captured during the cancel, pending, aborting, or stop event.
StopReasonExtent	Maximum Number of Stop Reasons	DINT	0 to 31	---	Admin.StopReasonExtent Maximum number of available stop reasons. If a value outside the effective range is input, the input is ignored and 31 is set.
ProdConsumedCount	Number of Used Raw Materials	ARRAY[0..31] of OmronLib\PackML_OPCUA\sPackML_CountDataType	---	---	Admin.ProdConsumedCount Number of materials used and consumed in the production machine.
ProdProcessedCount	Number of Produced Units	ARRAY[0..31] of OmronLib\PackML_OPCUA\sPackML_CountDataType	---	---	Admin.ProdProcessedCount Number of products processed in the production machine. Number of products including all good products and defective products.
ProdDefectiveCount	Number of Defective Products	ARRAY[0..31] of OmronLib\PackML_OPCUA\sPackML_CountDataType	---	---	Admin.ProdDefectiveCount Number of defective products processed in the production machine.
MachDesignSpeed	Maximum Designed Speed	REAL	Depends on data type.	/minute	Admin.MachDesignSpeed Maximum designed speed for the machine for each primary package per minute.
ModeCurrentTime	Dwell Time in Current Mode	ARRAY[0..31] of DINT	0 to 2147483647	Second	Admin.ModeCurrentTime Time that has elapsed after the mode was changed to the current mode. Array type that has the mode number in the index. The values of the modes except for the current mode are 0.
ModeCumulativeTime	Accumulated Dwell Time in Each Mode	ARRAY[0..31] of DINT	0 to 2147483647	Second	Admin.ModeCumulativeTime Accumulated dwell time in each mode after last reset. Array type that has the mode number in the index.

Member	Member name	Data type	Valid range	Unit	Description
StateCurrentTime	Dwell Time in Current State	ARRAY[0..31,0..19] of DINT	0 to 2147483647	Second	Admin.StateCurrentTime Time that has elapsed after the transition was made to the current state. Array type that has the mode number in the first index and the state number in the second index.
StateCumulativeTime	Accumulated Dwell Time in Each State	ARRAY[0..31,0..19] of DINT	0 to 2147483647	Second	Admin.StateCumulativeTime Accumulated dwell time in each state after last reset. Array type that has the mode number in the first index and the state number in the second index.
AccTimeSinceReset	Elapsed Time After Last Reset Execution	DINT	0 to 2147483647	Second	Admin.AccTimeSinceReset Elapsed time after last reset.

● OmronLib\PackML_OPCUA\sPackML_StatusObjectType

The structure of the status information required for the PackML OPC UA server.

Member	Member name	Data type	Valid range	Unit	Description
CurMachSpeed	Current Machine Speed	OmronLib\PackML_OPCUA\sCS_AnalogItemType	---	---	Status.CurMachSpeed The current value of the unit speed.
EquipmentBlocked	Processing Aborted Due to Downstream Equipment	BOOL	TRUE or FALSE	---	Status.EquipmentBlocked TRUE: The processing was aborted because the downstream equipment is unable to accept the material.
EquipmentStarved	Processing Aborted Due to Upstream Equipment	BOOL	TRUE or FALSE	---	Status.EquipmentStarved TRUE: The processing was aborted because the material is not fed from the upstream equipment.
UnitModeRequested	Mode Change Request Flag	BOOL	TRUE or FALSE	---	Status.UnitModeRequested TRUE: The mode change was requested.
UnitModeCurrent	Current Mode	OmronLib\PackML_OPCUA\ePackML_SupportedModesEnumType	Depends on data type.	---	Status.UnitModeCurrent Current mode
UnitModeChangeInProcess	Mode Change in Process Flag	BOOL	TRUE or FALSE	---	Status.UnitModeChangeInProcess TRUE: Mode change was requested and is in process. (After the mode change occurs, TRUE is set for one task period.)
StateRequested	State Transition Command Value	DINT	0 to 17	---	Status.StateRequested Destination state number of requested state transition.

Member	Member name	Data type	Valid range	Unit	Description
StateChangeInProcess	State Transition in Process Flag	BOOL	TRUE or FALSE	---	Status.StateChangeInProcess TRUE: State transition was requested and is in process. (After the state transition occurs, TRUE is set for one task period.)
MachSpeed	Machine Speed Setting	OmronLib\PackML_OPCUA\sCS_AnalogItemType	---	---	Status.MachSpeed The set value of the unit speed.
MaterialInterlock	Interlock Flag for Each Material	ARRAY[0..31] of BOOL	TRUE or FALSE	---	Status.MaterialInterlock Whether each material is ready for production. TRUE: Ready FALSE: Not ready
Parameter	Production Parameter	ARRAY[0..15] of OmronLib\PackML_OPCUA\sPackMLDescriptorDataType	---	---	Status.Parameter Current parameter used for production. The last parameter sent by the SetParameter method.
Product	Product List	ARRAY[0..3] of OmronLib\PackML_OPCUA\sPackML_ProductDataType	---	---	Status.Product List of products supported by this machine.

● OmronLib\PackML_OPCUA\sPackML_DescriptorDataType

This is the structure that corresponds to PackMLDescriptorDataType.

Member	Member name	Data type	Valid range	Unit	Description
ID	ID	DINT	Depends on data type.	---	A unique number assigned to a parameter.
Name	Name	STRING[80]	Depends on data type.	---	The name of the parameter.
Unit	Unit	OmronLib\PackML_OPCUA\sCS_EUInformationType	---	---	OPC UA engineering unit information.
Value	Value	REAL	Depends on data type.	---	A numeric value of the parameter.

● OmronLib\PackML_OPCUA\sPackML_AlarmDataType

This is the structure that corresponds to PackMLAlarmDataType and represents the PackML tag alarm structure.

Member	Member name	Data type	Valid range	Unit	Description
ID	ID	DINT	Depends on data type.	---	A unique number that is assigned to each type of alarm, stop, or warning.

Member	Member name	Data type	Valid range	Unit	Description
Value	Number	DINT	Depends on data type.	---	Message number for alarm, stop, or warning that is associated with ID. The user-specific details are displayed, or Alarm.ID is disassembled in more detail.
Message	Message	STRING[80]	Depends on data type.	---	Text string information of alarm, stop, or warning.
Category	Category	DINT	Depends on data type.	---	User-defined value that indicates the type of alarm, stop, or warning. Example: Electrical, mechanical, process, etc.
DateTime	Date and Time of Occurrence	DATE_AND_TIME	Depends on data type.	---	Date and time when the alarm, stop, or warning occurs.
AckDateTime	Date and Time of Acknowledgment	DATE_AND_TIME	Depends on data type.	---	Date and time when the alarm, stop, or warning is acknowledged.
Trigger	Trigger	BOOL	TRUE or FALSE	---	The trigger becomes TRUE when the alarm is active.

● OmronLib\PackML_OPCUA\sPackML_CountDataType

This is the structure that corresponds to PackMLCountDataType, and it is used for generating the outline information related to the system.

Member	Member name	Data type	Valid range	Unit	Description
ID	ID	DINT	Depends on data type.	---	User-defined value that indicates the consumed material (for produced or defective product).
Name	Name	STRING[80]	Depends on data type.	---	Material name linked with ID.
Unit	Unit	OmronLib\PackML_OPCUA\sCS_EUInformationType	---	---	Unit name associated with a specific material that is used for the machine.
Count	Quantity	DINT	Depends on data type.	---	Quantity of material (for produced or defective product) that is consumed for the current production job.
AccCount	Accumulated Quantity	DINT	Depends on data type.	---	Production (or consumption)

● OmronLib\PackML_OPCUA\sCS_AnalogItemType

This is the structure that corresponds to AnalogItemType.

Member	Member name	Data type	Valid range	Unit	Description
Value	Value	REAL	Depends on data type.	---	Value
InstrumentRange	Measurement Values Range	OmronLib\PackML_OPCUA\sCS_RangeType	---	---	The range of values that the instrument can return.
EURange	Normal Measurement Values Range	OmronLib\PackML_OPCUA\sCS_RangeType	---	---	The range of values likely to be obtained from normal operation. For example, it is used to scale a bar chart view automatically.
EngineeringUnits	Engineering Units	OmronLib\PackML_OPCUA\sCS_EUInformationType	---	---	A unit for a data item, such as degC, Hz, and second.
Definition	Definition	STRING[80]	Depends on data type.	---	A human-readable, vendor-specific string that specifies the calculation method for the value of a data item.
ValuePrecision	Value Precision	LREAL	Depends on data type.	---	A maximum precision that a server can hold for an item.

● OmronLib\PackML_OPCUA\sPackML_ProductDataType

This is the structure that corresponds to PackMLProductDataType and represents the PackML product information.

Member	Member name	Data type	Valid range	Unit	Description
ProductID	Product ID	DINT	Depends on data type.	---	A unique number assigned to a product.
ProcessVariables	Process Variables	ARRAY[0..15] of OmronLib\PackML_OPCUA\sPackML_DescriptorDataType	---	---	An array of process variables associated with the product.
Ingredients	Ingredients	ARRAY[0..4] of OmronLib\PackML_OPCUA\sPackML_IngredientsDataType	---	---	An array of ingredients associated with the product.

● OmronLib\PackML_OPCUA\sCS_EUInformationType

This is the structure that corresponds to EUInformation and represents the engineering unit information.

Member	Member name	Data type	Valid range	Unit	Description
NamespaceUri	Namespace URI	STRING[80]	Depends on data type.	---	The organizations defining the engineering units (company and standardizing organization) are identified.
UnitId	Unit ID	DINT	Depends on data type.	---	Identifier for evaluation by program.

Member	Member name	Data type	Valid range	Unit	Description
Display-Name	Display Name	STRING[80]	Depends on data type.	---	Display name of engineering units. Normally, it is indicated in abbreviation (example: time is indicated as "h", and speed is indicated as "m/s").
Description	Description	STRING[80]	Depends on data type.	---	Complete name of engineering units (example: "time", "meter per second").

● OmronLib\PackML_OPCUA\sCS_RangeType

This is the structure corresponding to Range and represents the range of numeric value.

Member	Member name	Data type	Valid range	Unit	Description
Low	Minimum Value	LREAL	Depends on data type.	---	Minimum value
High	Maximum Value	LREAL	Depends on data type.	---	Maximum value

● OmronLib\PackML_OPCUA\sPackML_IngredientsDataType

This is the structure that corresponds to PackMLIngredientsDataType and represents the PackML parameter structure.

Member	Member name	Data type	Valid range	Unit	Description
Ingredien-tID	Ingredient ID	DINT	Depends on data type.	---	A unique number assigned to an ingredient.
Parameter	Parameter	ARRAY[0..4] of OmronLib \PackML_OPCUA \sPackML_DescriptorDa-taType	---	---	An array of parameters corresponding to an ingredient.

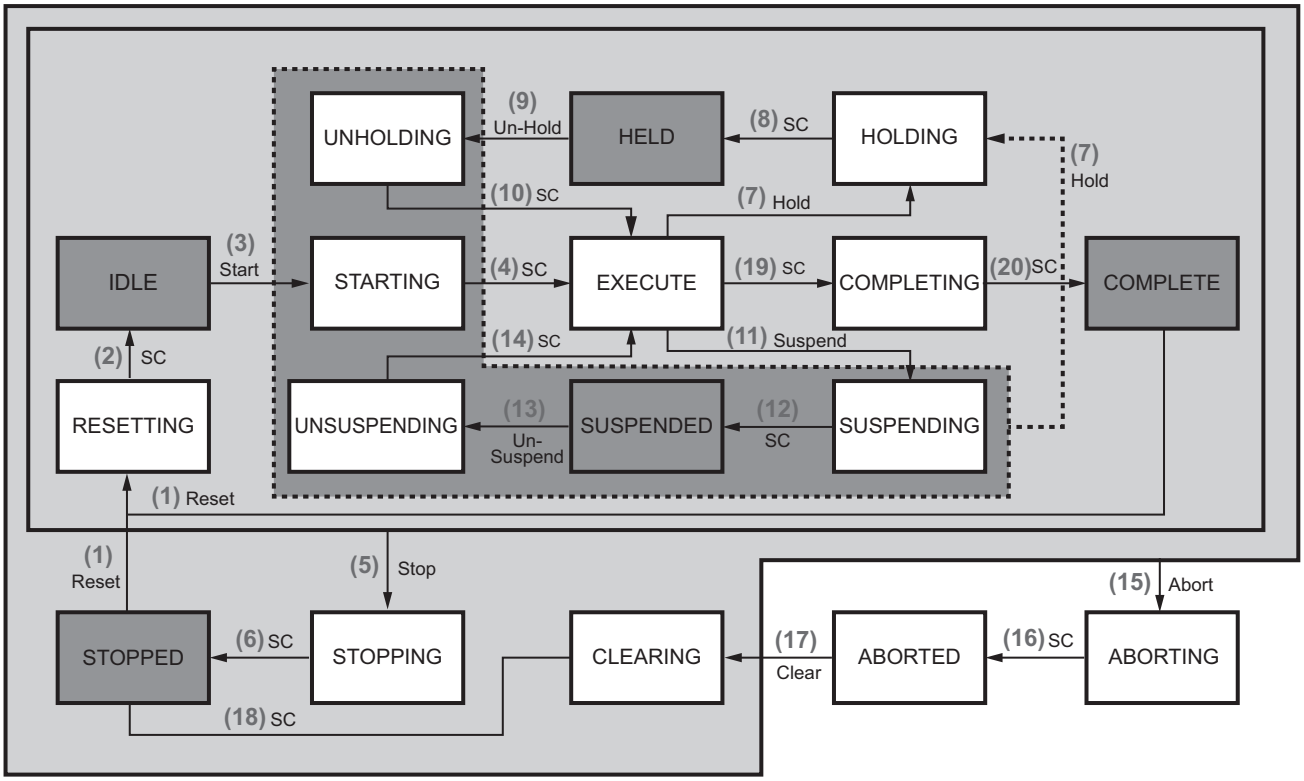
● OmronLib\PackML30\sPACKML_STATES_FLAG

Member	Data type	Description
ClearingState	BOOL	Structures that set the functions for each state
StoppedState	BOOL	
StartingState	BOOL	
IdleState	BOOL	
SuspendedState	BOOL	
ExecuteState	BOOL	
StoppingState	BOOL	
AbortingState	BOOL	
AbortedState	BOOL	
HoldingState	BOOL	
HeldState	BOOL	
UnholdingState	BOOL	
SuspendingState	BOOL	
UnsuspendingState	BOOL	
ResettingState	BOOL	
CompletingState	BOOL	
CompleteState	BOOL	

● OmronLib\PackML30\sPACKML_TRANSITION_COMMAND

Member	Data type	Description
Cmd_Reset	BOOL	The command to execute the state transition from Stopped or Complete to Resetting. (1)
Sts_Resetting_SC	BOOL	The request to execute the state transition from Resetting to Idle. (2)
Cmd_Start	BOOL	The command to execute the state transition from Idle to Starting. (3)
Sts_Starting_SC	BOOL	The request to execute the state transition from Starting to Execute. (4)
Cmd_Stop	BOOL	The command to execute the state transition from Idle, Resetting, Starting, Execute, Completing, Complete, Holding, Held, Unholding, Suspending, Suspended, or Unsuspending to Stopping. (5)
Sts_Stopping_SC	BOOL	The request to execute the state transition from Stopping to Stopped. (6)
Cmd_Hold	BOOL	The command to execute the state transition from Starting, Suspended, Execute, Unholding, Suspending, or Unsuspending to Holding. (7)
Sts_Holding_SC	BOOL	The request to execute the state transition from Holding to Held. (8)
Cmd_UnHold	BOOL	The command to execute the state transition from Held to UnHolding. (9)
Sts_UnHolding_SC	BOOL	The request to execute the state transition from UnHolding to Execute. (10)
Cmd_Suspend	BOOL	The command to execute the state transition to Execute to Suspending. (11)
Sts_Suspending_SC	BOOL	The request to execute the state transition from Suspending to Suspended. (12)
Cmd_UnSuspend	BOOL	The command to execute the state transition from Suspended to UnSuspending. (13)
Sts_UnSuspending_SC	BOOL	The request to execute the state transition from UnSuspending to Execute. (14)

Member	Data type	Description
Cmd_Abort	BOOL	The command to execute the state transition from the state except Aborting and Aborted state, to Aborting. (15)
Sts_Aborting_SC	BOOL	The request to execute the state transition from Aborting to Aborted. (16)
Cmd_Clear	BOOL	The command to execute the state transition from Aborted to Clearing. (17)
Sts_Clearing_SC	BOOL	The request to execute the state transition from Clearing to Stopped. (18)
Sts_Execute_SC	BOOL	The request to execute the state transition from Execute to Completing. (19)
Sts_Completing_SC	BOOL	The request to execute the state transition from Completing to Complete. (20)



Enumerations

● OmronLib\PackML_OPCUA\ePackML_SupportedModesEnumType

This is the enumeration that corresponds to the modes described in *Mode* on page 2-2.

Member	Enum value	Description
ProductionMode	1	Production mode. It corresponds to the PackML Production Mode in *Mode* on page 2-2.
MaintenanceMode	2	Maintenance mode. It corresponds to the PackML Maintenance Mode in *Mode* on page 2-2.
ManualMode	3	Manual mode. It corresponds to the PackML Manual Mode in *Mode* on page 2-2.
UserDefinedMode1	4	User-defined mode 1

Member	Enum value	Description
UserDefinedMode2	5	User-defined mode 2
UserDefinedMode3	6	User-defined mode 3
UserDefinedMode4	7	User-defined mode 4
UserDefinedMode5	8	User-defined mode 5
UserDefinedMode6	9	User-defined mode 6
UserDefinedMode7	10	User-defined mode 7
UserDefinedMode8	11	User-defined mode 8
UserDefinedMode9	12	User-defined mode 9
UserDefinedMode10	13	User-defined mode 10
UserDefinedMode11	14	User-defined mode 11
UserDefinedMode12	15	User-defined mode 12
UserDefinedMode13	16	User-defined mode 13
UserDefinedMode14	17	User-defined mode 14
UserDefinedMode15	18	User-defined mode 15
UserDefinedMode16	19	User-defined mode 16
UserDefinedMode17	20	User-defined mode 17
UserDefinedMode18	21	User-defined mode 18
UserDefinedMode19	22	User-defined mode 19
UserDefinedMode20	23	User-defined mode 20
UserDefinedMode21	24	User-defined mode 21
UserDefinedMode22	25	User-defined mode 22
UserDefinedMode23	26	User-defined mode 23
UserDefinedMode24	27	User-defined mode 24
UserDefinedMode25	28	User-defined mode 25
UserDefinedMode26	29	User-defined mode 26
UserDefinedMode27	30	User-defined mode 27
UserDefinedMode28	31	User-defined mode 28

Function

This function block corresponds to PackMLBaseObjectType, and exposes the current mode, current state, and other information to the OPC UA node for the mode change/state transition command according to the mode/state machine stipulated by PackML.

● State Machine

Based on the mode/state machine stipulated by PackML, the function block outputs the current mode and state according to the mode change/state transition command. You can execute the mode change/state transition command with an input variable in this function block or from the OPC UA client.

Refer to *Mode* on page 2-2 for the typical state machine configuration.

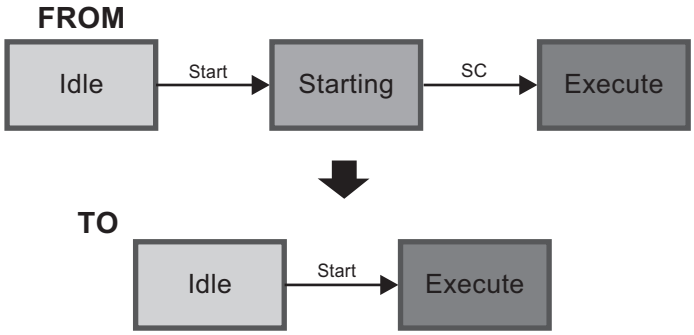
<Mode/State Machine Settings>

Configure the PackML mode/state machine operations based on the specifications of the application you create.

Configure the basic operations with the Cfg_DisabledStates and Cfg_ModeSwitchableStates input variables. Both input variables have the array of 1 to 31. This array index represents the mode number.

a. Per mode, set the unused states.
b. Per mode, set the states in which the mode is permitted to be switched.

<Unused State Settings>
For an unused state, set TRUE to the OmronLib\PackML30\sPACKML_STATES_FLAG array member of the element number for the mode of the Cfg_DisabledStates [1..31] input variable.
The setting is updated only when Enable changes to TRUE. You can check the setting currently under execution with the Cfg_DisabledStatesActual output variable.
When a transition command to an unused state is made, the transition is made to the next state unconditionally. The following example shows the state transition when the Starting state is not used.



The state transitions for the modes 1, 2, and 3 are fixed as shown in the table below, and they cannot be changed.

State	Cfg_DisabledStatesActual		
	[1] Mode1: Produce	[2] Mode2: Maintenance	[3] Mode3: Manual
Clearing	FALSE	FALSE	FALSE
Stopped	FALSE	FALSE	FALSE
Starting	FALSE	FALSE	FALSE
Idle	FALSE	FALSE	FALSE
Suspended	FALSE	TRUE	TRUE
Execute	FALSE	FALSE	FALSE
Stopping	FALSE	FALSE	FALSE
Aborting	FALSE	FALSE	FALSE
Aborted	FALSE	FALSE	FALSE
Holding	FALSE	FALSE	TRUE
Held	FALSE	FALSE	TRUE
Unholding	FALSE	FALSE	TRUE
Suspending	FALSE	TRUE	TRUE
Unsuspending	FALSE	TRUE	TRUE
Resetting	FALSE	FALSE	FALSE
Completing	FALSE	TRUE	TRUE
Complete	FALSE	TRUE	TRUE

If a setting content is not appropriate, it is corrected inside this function block according to the following rules.

a. Stopped, Idle, Execute, and Aborted are essential states and are automatically set to be used.
b. When the Wait states except for Stopped, Idle, Execute, and Aborted are set as the unused states, the related Acting (…ing) states are automatically set as the unused states accordingly.

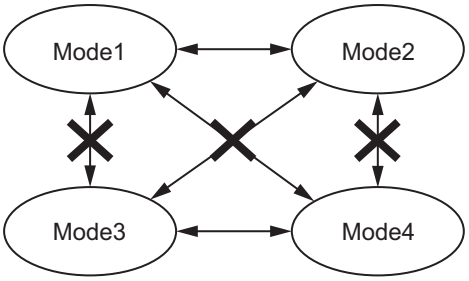
State set as "Disable State"	State as "Unused State" automatically
Resetting	---
Starting	---
Suspending	---
Unsuspending	---
Holding	---
UnHolding	---
Completing	---
Aborting	---
Clearing	---
Idle	---
Held	Holding, Unholding
Suspended	Suspending, Unsuspending
Complete	Completing
Stopped	---
Aborted	---
Execute	---

<Permit Mode Switch Setting>

For state to permit the mode switch, set TRUE to the OmronLib\PackML30\sPACKML_STATES_FLAG array member of the element number for the mode of the Cfg_ModeSwitchableStates [1..31] input variable.

- When both the current mode and requested mode have this flag set to TRUE for the current state, the mode can be changed.
- With this function block, the mode is switchable to all of the modes in which the permit mode switch flag is set. A function that allows a mode to be switched between specific modes is not supported.

Add required interlock logic outside of this function block if you need an application that allows a mode to be switched between specific modes in a specific state, and to control a mode to be switched between the specific modes as the following diagram shows.



<State Transition>

- In the transition command Cmd_StateTransition, more than one transition request flags can be set to TRUE. However, a transition request which is not executable for the current state is ignored. Even if no executable transition request is included, it is simply ignored without any error.

- The transition command Cmd_StateTransition and the transition command by a method execution are summarized in this function block using the PMLTransitionCmd_Summarize function. When more than one executable transition command is included, only one state transition is executed in the order of Cmd_Abort, Cmd_Stop, Sts_xxx_SC,Cmd_Hold, and Cmd_xxx.

<Mode Switch Function>
You can make the mode switch request with the following two methods.

1. From the OPC UA client, set a valid value (1 to 31) different from the current mode to RequestedMode, and call the SetUnitMode method.
2. Set the ModeNum input variable in this function block to a valid value (1 to 31) different from the current mode, and change the Cmd_ModeChange input variable to TRUE.

When the above two mode switch requests are made in the same period, the request by this function block shown in 2. is executed first.
When the mode switch request and a valid state transition command are given at the same time, the state transition is executed first.

● State Timer

The function block measures a dwell time (second) in each state and mode of the mode/state machine stipulated by PackML, and outputs to the following variables.
When Enable in this function block changes to TRUE, the time measurement starts.
The accumulated dwell time is reset by changing the Cmd_ResetAllDwellTimes input variable to TRUE.

Variable name	Description
Admin.ModeCurrentTime [mode number]	Dwell Time in Mode
Admin.ModeCumulativeTime [mode number]	Accumulated Dwell Time in Mode
Admin.StateCurrentTime [mode number, state number]	Dwell Time in State
Admin.StateCumulativeTime [mode number, state number]	Accumulated Dwell Time in State
Admin.AccTimeSinceReset	Elapsed Time After Last Reset

● OPC UA Node Exposing

The function block exposes the PackML information models to the OPC UA client. Refer to *2-1-3 Node Exposure* on page 2-6 for the information models that are exposed.
The following table shows the methods that can be called from the OPC UA client and their processing contents.

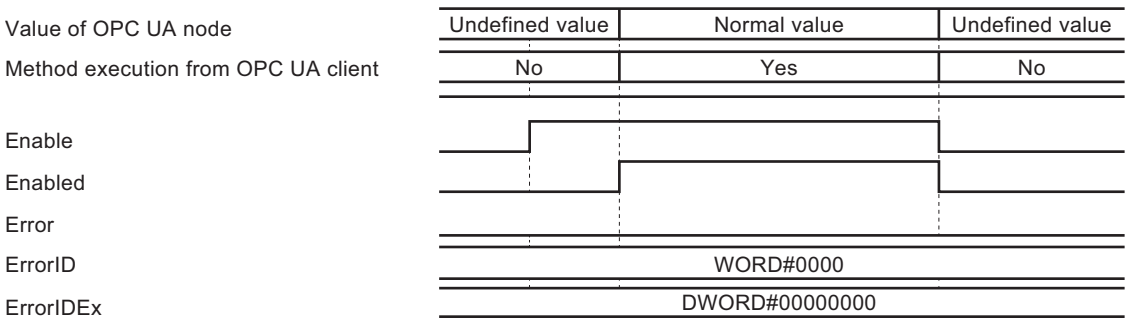
Method	Description of processing
SetUnitMode	Changes to the mode of input argument when the current state allows the mode switch.
SetMachSpeed	Sets the value of input argument to Status.MachSpeed.
SetProduct	Sets the value of input argument to Status.Product.
BaseStateMachine.Abort	Executes the transition to the Aborting state when the current state is Cleared.
BaseStateMachine.Clear	Executes the transition to the Clearing state when the current state is Aborted.
BaseStateMachine.MachineState.Stop	Executes the transition to the Stopping state when the current state is Running.

Method	Description of processing
BaseStateMachine.MachineState.Reset	Executes the transition to the Resetting state when the current state is Stopped.
BaseStateMachine.MachineState.ExecuteState.ToComplete	Executes the transition to the Completing state when the current state is Execute.
BaseStateMachine.MachineState.ExecuteState.Start	Executes the transition to the Starting state when the current state is Idle. Outputs the value passed with the input argument to the StartMethodParam output variable.
BaseStateMachine.MachineState.ExecuteState.Unhold	Executes the transition to the Unholding state when the current state is Held.
BaseStateMachine.MachineState,ExecuteState.Suspend	Executes the transition to the Suspending state when the current state is Execute.
BaseStateMachine.MachineState.ExecuteState.Unsuspend	Executes the transition to the Unsuspending state when the current state is Suspended.
BaseStateMachine.MachineState.ExecuteState.Hold	Executes the transition to the Holding state when the current state is Starting, Suspended, Execute, Unholding, Suspending, or Unsuspending.
BaseStateMachine.MahcineState.ExecuteState.Reset	Executes the transition to the Resetting state when the current state is Complete.
SetInterlock	Sets Status.MaterialInterlock according to the input argument. The argument range is 0 to 31.
SetParameter	Sets the value of input argument to Status.Parameter.

Timing Chart

The timing chart is shown below.

- When *Enable* changes to TRUE, the initialization is executed first. When the initialization is completed, *Enabled* changes to TRUE.
- When you assign this function block instance, the OPC UA node is exposed, but while *Enabled* is FALSE, the value of each node is undefined, and methods cannot be executed.



Precautions for Correct Use

- Assign an instance of this function block to a primary periodic task or periodic task. If you assign it to an event task, the NJ/NX-series CPU Unit may not correctly receive the execution results of the method.

- Assign an instance of this function block so that the instance is executed for each task period. Be careful that there are no periods in which the control does not run in the JMP instruction, etc. If there is a period in which the control does not run, OPC UA FW may not correctly receive the execution results of the method.
  Ensure that Enable, an instance of this function block, is TRUE from the start operation cycle. If Enable is FALSE at the start of operation, it may not be possible to read/write from the client to the OPC UA node.
- If there are multiple state transition requests for instances of this function block other than method execution from OPC UA clients, enter the one summarized using the PMLTransitionCmd_Summarize function or the PMLTransitionCmd_SummarizePackTagCtrlCmd function provided in this library to Cmd_StateTransition of this function block.
- While Enabled is FALSE, the value of each OPC UA node is undefined, and methods are not executed.

Troubleshooting

Error code	Expansion error code	Status	Description	Correction
16#0000	16#00000000	Normal end	---	---

This function block does not output an error. It ignores incorrect inputs, and always outputs Enabled to TRUE. While Enabled is TRUE, it outputs valid values to the other outputs and OPC UA node.

PMLCtrlCmd_**

The functions check which transition command is the number of Command.CntrlCmd stipulated by PackTag.

Function name	Name	FB/FUN	Graphic expression	ST expression
PMLCtrlCmd_**	Transition Command Display	FUN	\\OmronLib\PackML30\PMLCtrlCmd_** EN CtrlCmd "**" must be a transition command.	\\OmronLib \PackML30\PMLCtrlCmd_** (CtrlCmd); "**" must be a transition command.

Function Block and Function Information

Item	Function name	Description
Library file name	---	OmronLib_PackML_OPCUAxx_Vy_y.slr*1
Namespace	---	OmronLib\PackML30
Function block and function number	PMLCtlCmd_Reset	00103
	PMLCtlCmd_Start	00104
	PMLCtlCmd_Stop	00105
	PMLCtlCmd_Hold	00106
	PMLCtlCmd_Unhold	00107
	PMLCtlCmd_Suspend	00108
	PMLCtlCmd_Unsuspend	00109
	PMLCtlCmd_Abort	00110
	PMLCtlCmd_Clear	00111
Publish/Do not publish source code	---	Not published.
Function block and function version	---	1.00

*1. x indicates the version of the OPC UA for PackML specifications (example: in the case of 1.0.1, the upper two digits are used to indicate 10), and y indicates the version of the library.

Variables

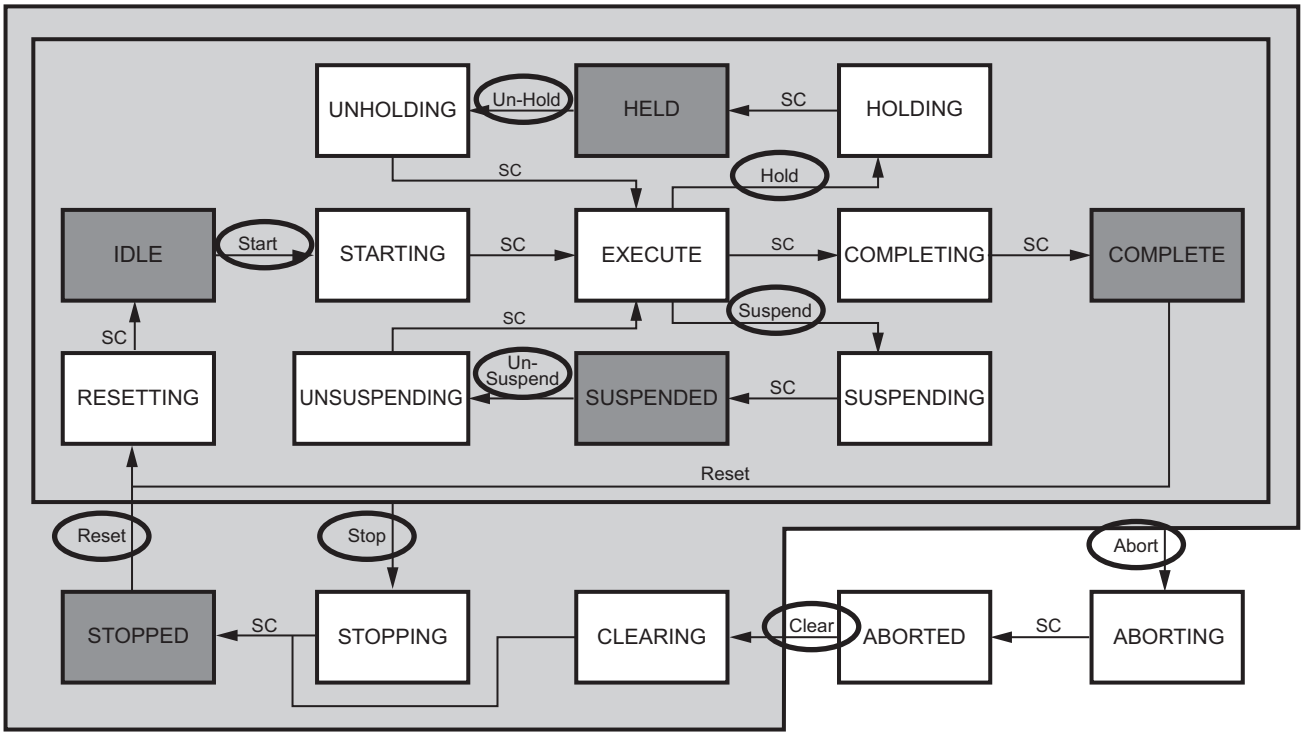
Name	I/O	Data type	Default	Description
EN	Input	BOOL	TRUE	Function Execution Control Flag When this variable is FALSE, internal logic does not operate even if it is called.
CtrlCmd	Input	DINT	0	Transition Command Number Specifies the value obtained from the Command.CntrlCmd tag of PackTag. The range of input value is from 1 to 9.

Name	I/O	Data type	Default	Description
<Function name>	Return value	BOOL	---	Return Value TRUE is returned only when the entered transition number represents the function name. (FALSE is returned when a value out of the range is entered.)

Function

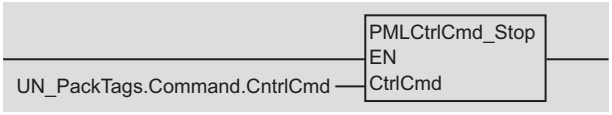
The functions check which transition command is the number of Command.CntrlCmd stipulated by PackTag. With these functions, the user no longer needs to see the specification to find out which transition number actually represents which transition.

The following diagram shows the transitions and their numbers specified by Command.CntrlCmd.



0	Undefined
1	Reset
2	Start
3	Stop
4	Hold
5	Unhold
6	Suspend
7	Unsuspend
8	Abort
9	Clear

The example of how to see which specified transition command is the Stop command.



Function List

Name	Description
PMLCtrlCmd_Reset	TRUE is returned when the entered transition number is 1.
PMLCtrlCmd_Start	TRUE is returned when the entered transition number is 2.
PMLCtrlCmd_Stop	TRUE is returned when the entered transition number is 3.
PMLCtrlCmd_Hold	TRUE is returned when the entered transition number is 4.
PMLCtrlCmd_Unhold	TRUE is returned when the entered transition number is 5.
PMLCtrlCmd_Suspend	TRUE is returned when the entered transition number is 6.
PMLCtrlCmd_Unsuspend	TRUE is returned when the entered transition number is 7.
PMLCtrlCmd_Abort	TRUE is returned when the entered transition number is 8.
PMLCtrlCmd_Clear	TRUE is returned when the entered transition number is 9.

PMLState_Is**

The functions check which state number stipulated by PackML represents which state.

Function name	Name	FB/FUN	Graphic expression	ST expression
PMLState_Is**	State Output	FUN	\\OmronLib\PackML30\PMLState_Is** EN StateNumber "**" must be a state.	\\OmronLib \PackML30\PMLState_Is** (StateNumber); "**" must be a state.

Function Block and Function Information

Item	Function name	Description
Library file name	---	OmronLib_PackML_OPCUAxx_Vy_y.slr*1
Namespace	---	OmronLib\PackML30
Function block and function number	PMLState_IsClearing	00112
	PMLState_IsStopped	00113
	PMLState_IsStarting	00114
	PMLState_IsIdle	00115
	PMLState_IsSuspended	00116
	PMLState_IsExecute	00117
	PMLState_IsStopping	00118
	PMLState_IsAborting	00119
	PMLState_IsAborted	00120
	PMLState_IsHolding	00121
	PMLState_IsHeld	00122
	PMLState_IsUnholding	00123
	PMLState_IsSuspending	00124
	PMLState_IsUnsuspending	00125
	PMLState_IsResetting	00126
	PMLState_IsCompleting	00127
	PMLState_IsComplete	00128
Publish/Do not publish source code	---	Not published.
Function block and function version	---	1.00

*1. x indicates the version of the OPC UA for PackML specifications (example: in the case of 1.0.1, the upper two digits are used to indicate 10), and y indicates the version of the library.

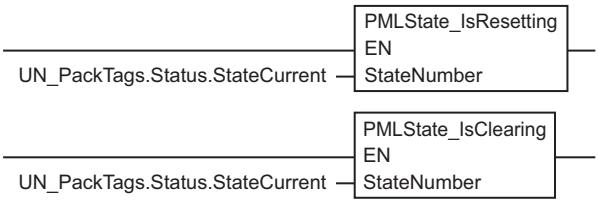
Variables

Name	I/O	Data type	Default	Description
EN	Input	BOOL	TRUE	Function Execution Control Flag When this variable is FALSE, internal logic does not operate even if it is called.

Name	I/O	Data type	Default	Description
StateNumber	Input	DINT	0	State Number Specifies the state number to check. The range of input value is from 1 to 17.
<the same with the function name>	Return value	BOOL	---	Return Value TRUE is returned only when the entered state number represents the function name. (FALSE is returned when a state number out of the range is entered.)

Function

The functions check which state number stipulated by PackML represents which state. With these functions, the user no longer needs to see the specification to find out which state number, which is output by the PackML mode/state control function block, represents which state.

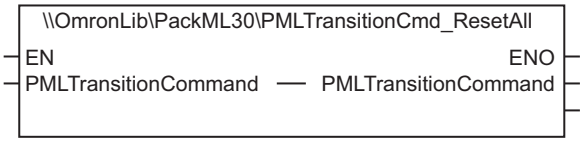


Function List

Name	Description
PMLState_IsClearing	TRUE is returned when the entered state number is 1.
PMLState_IsStopped	TRUE is returned when the entered state number is 2.
PMLState_IsStarting	TRUE is returned when the entered state number is 3.
PMLState_IsIdle	TRUE is returned when the entered state number is 4.
PMLState_IsSuspended	TRUE is returned when the entered state number is 5.
PMLState_IsExecute	TRUE is returned when the entered state number is 6.
PMLState_IsStopping	TRUE is returned when the entered state number is 7.
PMLState_IsAborting	TRUE is returned when the entered state number is 8.
PMLState_IsAborted	TRUE is returned when the entered state number is 9.
PMLState_IsHolding	TRUE is returned when the entered state number is 10.
PMLState_IsHeld	TRUE is returned when the entered state number is 11.
PMLState_IsUnholding	TRUE is returned when the entered state number is 12.
PMLState_IsSuspending	TRUE is returned when the entered state number is 13.
PMLState_IsUnsuspending	TRUE is returned when the entered state number is 14.
PMLState_IsResetting	TRUE is returned when the entered state number is 15.
PMLState_IsCompleting	TRUE is returned when the entered state number is 16.
PMLState_IsComplete	TRUE is returned when the entered state number is 17.

PMLTransitionCmd_ResetAll

For the state transition command sPACKML_TRANSITION_COMMAND structure variable, this function resets every BOOL member that indicates the state transition to FALSE.
This function is used for initializing the state transition request to the host module.

Function name	Name	FB/FUN	Graphic expression	ST expression
PMLTransitionCmd_ResetAll	Transition Command All Reset	FUN		\\OmronLib\PackML30\PMLTransitionCmd_ResetAll (PMLTransitionCommand);

Function Block and Function Information

Item	Description
Library file name	OmronLib_PackML_OPCUAxx_Vy_y.slr*1
Namespace	OmronLib\PackML30
Function block and function number	00129
Publish/Do not publish source code	Not published.
Function block and function version	1.00

*1. x indicates the version of the OPC UA for PackML specifications (example: in the case of 1.0.1, the upper two digits are used to indicate 10), and y indicates the version of the library.

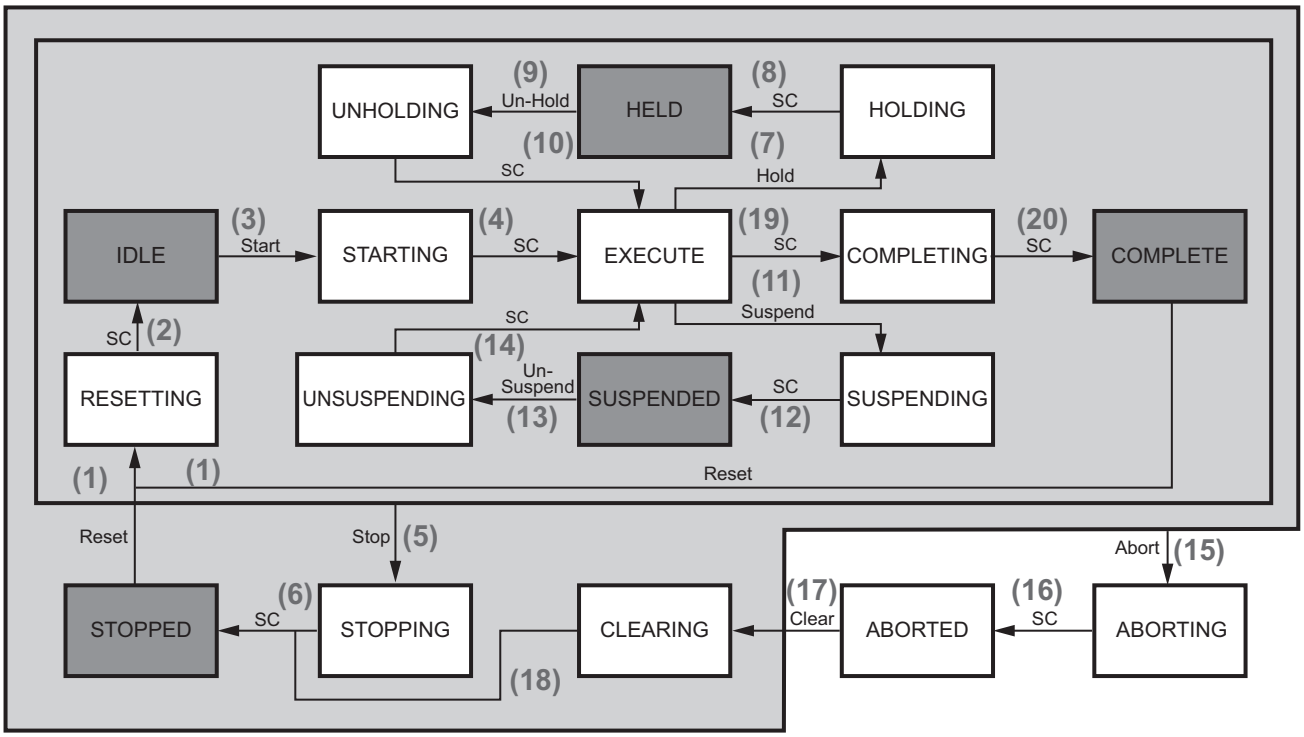
Variables

Name	I/O	Data type	Default	Description
EN	Input	BOOL	TRUE	Function Execution Control Flag When this variable is FALSE, internal logic does not operate even if it is called.
ENO	Output	BOOL	---	Function Execution Control Flag Output
PMLTransitionCommand	In-out	OmronLib\PackML30\sPACKML_TRANSITION_COMMAND	---	PackML State Transition Command

Structures

● sPACKML_TRANSITION_COMMAND

Name	Data type	Description
sPACKML_TRANSITION_COMMAND	STRUCT	The structure that indicates a transition for the PackML state machine.
Cmd_Reset	BOOL	The command to execute the state transition from Stopped or Complete to Resetting. (1)
Sts_Resetting_SC	BOOL	The request to execute the state transition from Resetting to Idle. (2)
Cmd_Start	BOOL	The command to execute the state transition from Idle to Starting. (3)
Sts_Starting_SC	BOOL	The request to execute the state transition from Starting to Execute. (4)
Cmd_Stop	BOOL	The command to execute the state transition from Idle, Resetting, Starting, Execute, Completing, Complete, Holding, Held, Unholding, Suspending, Suspended, or Unsuspending to Stopping. (5)
Sts_Stopping_SC	BOOL	The request to execute the state transition from Stopping to Stopped. (6)
Cmd_Hold	BOOL	The command to execute the state transition from Execute to Holding. (7)
Sts_Holding_SC	BOOL	The request to execute the state transition from Holding to Held. (8)
Cmd_UnHold	BOOL	The command to execute the state transition from Held to UnHolding. (9)
Sts_UnHolding_SC	BOOL	The request to execute the state transition from UnHolding to Execute. (10)
Cmd_Suspend	BOOL	The command to execute the state transition to Execute to Suspending. (11)
Sts_Suspending_SC	BOOL	The request to execute the state transition from Suspending to Suspended. (12)
Cmd_UnSuspend	BOOL	The command to execute the state transition from Suspended to UnSuspending. (13)
Sts_UnSuspending_SC	BOOL	The request to execute the state transition from UnSuspending to Execute. (14)
Cmd_Abort	BOOL	The command to execute the state transition from the state except Aborting and Aborted state, to Aborting. (15)
Sts_Aborting_SC	BOOL	The request to execute the state transition from Aborting to Aborted. (16)
Cmd_Clear	BOOL	The command to execute the state transition from Aborted to Clearing. (17)
Sts_Clearing_SC	BOOL	The request to execute the state transition from Clearing to Stopped. (18)
Sts_Execute_SC	BOOL	The request to execute the state transition from Execute to Completing. (19)
Sts_Completing_SC	BOOL	The request to execute the state transition from Completing to Complete. (20)



Function

For the state transition command sPACKML_TRANSITION_COMMAND structure variable, this function resets every BOOL member that indicates the state transition to FALSE.
This function is used for initializing the state transition request to the host module.

PMLTransitionCmd_ResetAllCmd-SetAllSC

For the state transition command sPACKML_TRANSITION_COMMAND structure variable, this function resets all the state transition commands (Cmd_<state name>) in the BOOL type members which indicates state transition to FALSE, and sets all the Wait state completion notifications (STs_<state name>_SC) to TRUE.
This function is used for initializing the state transition request to the host module.

Function name	Name	FB/FUN	Graphic expression	ST expression
PMLTransitionCmd_ResetAllCmd-SetAllSC	Transition Command Reset State Set	FUN	\\OmronLib\PackML30 \PMLTransitionCmd_ResetAllCmdSetAllSC EN — ENO PMLTransitionCommand — PMLTransitionCommand	\\OmronLib\PackML30\PMLTransitionCmd_ResetAllCmdSetAllSC (PMLTransitionCommand);

Function Block and Function Information

Item	Description
Library file name	OmronLib_PackML_OPCUAxx_Vy_y.slr*1
Namespace	OmronLib\PackML30
Function block and function number	00130
Publish/Do not publish source code	Not published.
Function block and function version	1.00

*1. x indicates the version of the OPC UA for PackML specifications (example: in the case of 1.0.1, the upper two digits are used to indicate 10), and y indicates the version of the library.

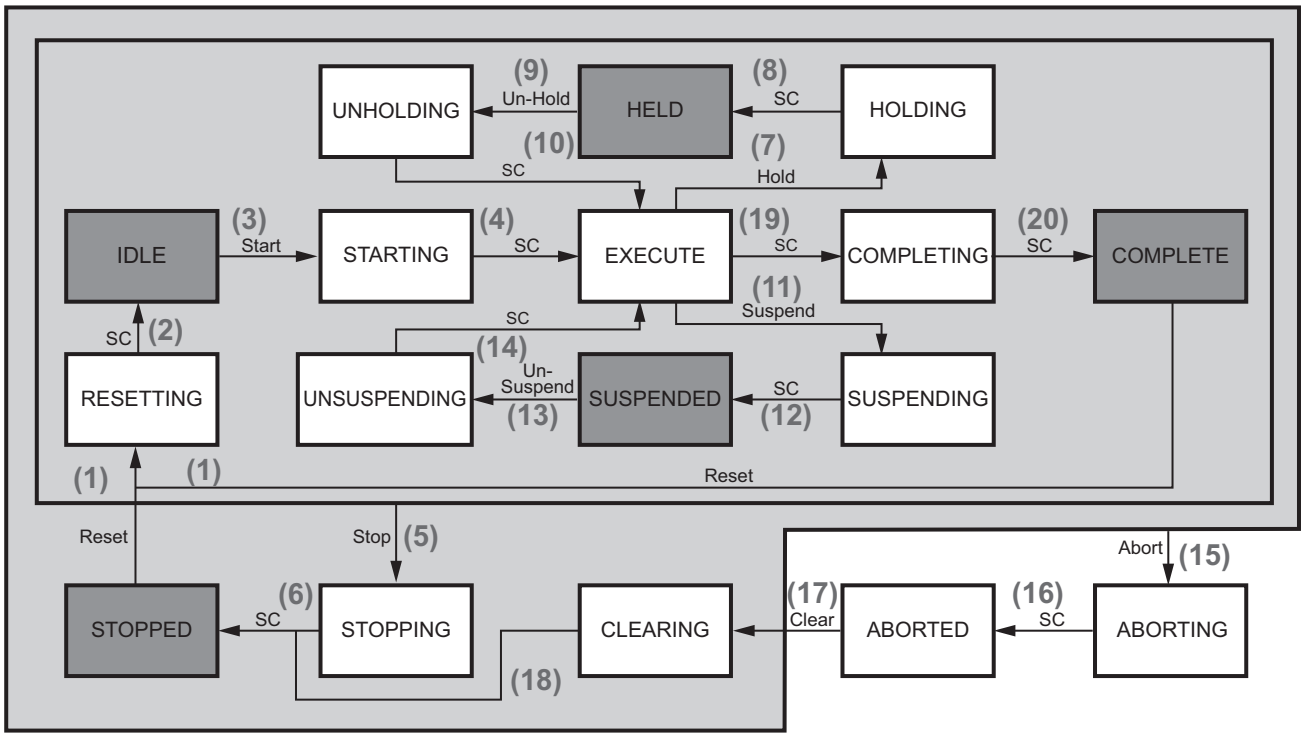
Variables

Name	I/O	Data type	Default	Description
EN	Input	BOOL	TRUE	Function Execution Control Flag When this variable is FALSE, internal logic does not operate even if it is called.
ENO	Output	BOOL	---	Function Execution Control Flag Output
PMLTransitionCommand	In-out	OmronLib\PackML30\sPACKML_TRANSITION_COMMAND	---	PackML State Transition Command

Structures

● sPACKML_TRANSITION_COMMAND

Name	Data type	Description
sPACKML_TRANSITION_COMMAND	STRUCT	The structure that indicates a transition for the PackML state machine.
Cmd_Reset	BOOL	The command to execute the state transition from Stopped or Complete to Resetting. (1)
Sts_Resetting_SC	BOOL	The request to execute the state transition from Resetting to Idle. (2)
Cmd_Start	BOOL	The command to execute the state transition from Idle to Starting. (3)
Sts_Starting_SC	BOOL	The request to execute the state transition from Starting to Execute. (4)
Cmd_Stop	BOOL	The command to execute the state transition from Idle, Resetting, Starting, Execute, Completing, Complete, Holding, Held, Unholding, Suspending, Suspended, or Unsuspending to Stopping. (5)
Sts_Stopping_SC	BOOL	The request to execute the state transition from Stopping to Stopped. (6)
Cmd_Hold	BOOL	The command to execute the state transition from Execute to Holding. (7)
Sts_Holding_SC	BOOL	The request to execute the state transition from Holding to Held. (8)
Cmd_UnHold	BOOL	The command to execute the state transition from Held to UnHolding. (9)
Sts_UnHolding_SC	BOOL	The request to execute the state transition from UnHolding to Execute. (10)
Cmd_Suspend	BOOL	The command to execute the state transition to Execute to Suspending. (11)
Sts_Suspending_SC	BOOL	The request to execute the state transition from Suspending to Suspended. (12)
Cmd_UnSuspend	BOOL	The command to execute the state transition from Suspended to UnSuspending. (13)
Sts_UnSuspending_SC	BOOL	The request to execute the state transition from UnSuspending to Execute. (14)
Cmd_Abort	BOOL	The command to execute the state transition from the state except Aborting and Aborted state, to Aborting. (15)
Sts_Aborting_SC	BOOL	The request to execute the state transition from Aborting to Aborted. (16)
Cmd_Clear	BOOL	The command to execute the state transition from Aborted to Clearing. (17)
Sts_Clearing_SC	BOOL	The request to execute the state transition from Clearing to Stopped. (18)
Sts_Execute_SC	BOOL	The request to execute the state transition from Execute to Completing. (19)
Sts_Completing_SC	BOOL	The request to execute the state transition from Completing to Complete. (20)



Function

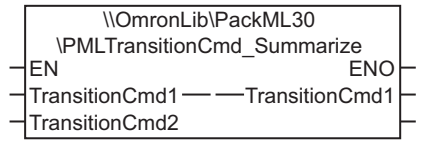
For the state transition command sPACKML_TRANSITION_COMMAND structure variable, this function resets all the state transition commands (Cmd_<state name>) in the BOOL type members which indicate state transition to FALSE, and sets all the Waite state completion notifications (Sts_<state name>_SC) to TRUE.

This function is used for initializing the state transition request to the host module.

PMLTransitionCmd_Summarize

This function processes the state transition requests sPACKML_TRANSITION_COMMAND structure variable which are output by each lower module as follows, and outputs them as the state transition requests for the host module.

- Execute OR evaluation on State transition commands (Cmd_<state name>)
- Execute AND evaluation on Wait state completion notifications (Sts_<state name>_SC)

Function name	Name	FB/FUN	Graphic expression	ST expression
PMLTransitionCmd_Summarize	Transition Command Summarize	FUN		\\OmronLib \PackML30\PMLTransitionCmd_Summarize (TransitionCmd1, TransitionCmd2);

Function Block and Function Information

Item	Description
Library file name	OmronLib_PackML_OPCUAxx_Vy_y.slr*1
Namespace	OmronLib\PackML30
Function block and function number	00131
Publish/Do not publish source code	Not published.
Function block and function version	1.00

*1. x indicates the version of the OPC UA for PackML specifications (example: in the case of 1.0.1, the upper two digits are used to indicate 10), and y indicates the version of the library.

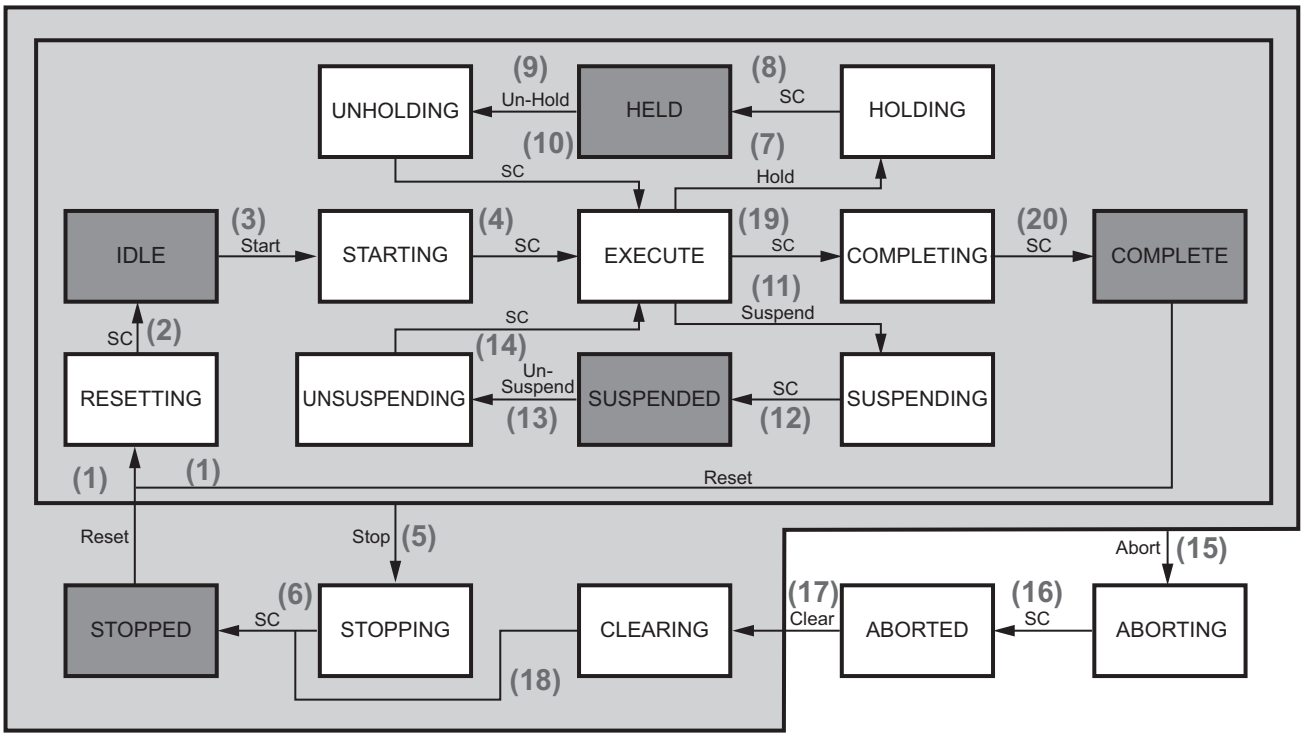
Variables

Name	I/O	Data type	Default	Description
EN	Input	BOOL	TRUE	Function Execution Control Flag When this variable is FALSE, internal logic does not operate even if it is called.
ENO	Output	BOOL	---	Function Execution Control Flag Output
TransitionCmd1	In-out	OmronLib \PackML30\sPACKML_TRANSITION_COMMAND	---	Summarize Destination State Transition Request Specifies state transition requests to be updated by merging TransitionCmd2.
TransitionCmd2	Input	OmronLib \PackML30\sPACKML_TRANSITION_COMMAND	FALSE(All member)	Specifies a state transition request to be merged into TransitionCmd1.

Structures

● sPACKML_TRANSITION_COMMAND

Name	Data type	Description
sPACKML_TRANSITION_COMMAND	STRUCT	The structure that indicates a transition for the PackML state machine.
Cmd_Reset	BOOL	The command to execute the state transition from Stopped or Complete to Resetting. (1)
Sts_Resetting_SC	BOOL	The request to execute the state transition from Resetting to Idle. (2)
Cmd_Start	BOOL	The command to execute the state transition from Idle to Starting. (3)
Sts_Starting_SC	BOOL	The request to execute the state transition from Starting to Execute. (4)
Cmd_Stop	BOOL	The command to execute the state transition from Idle, Resetting, Starting, Execute, Completing, Complete, Holding, Held, Unholding, Suspending, Suspended, or Unsuspending to Stopping. (5)
Sts_Stopping_SC	BOOL	The request to execute the state transition from Stopping to Stopped. (6)
Cmd_Hold	BOOL	The command to execute the state transition from Execute to Holding. (7)
Sts_Holding_SC	BOOL	The request to execute the state transition from Holding to Held. (8)
Cmd_UnHold	BOOL	The command to execute the state transition from Held to UnHolding. (9)
Sts_UnHolding_SC	BOOL	The request to execute the state transition from UnHolding to Execute. (10)
Cmd_Suspend	BOOL	The command to execute the state transition to Execute to Suspending. (11)
Sts_Suspending_SC	BOOL	The request to execute the state transition from Suspending to Suspended. (12)
Cmd_UnSuspend	BOOL	The command to execute the state transition from Suspended to UnSuspending. (13)
Sts_UnSuspending_SC	BOOL	The request to execute the state transition from UnSuspending to Execute. (14)
Cmd_Abort	BOOL	The command to execute the state transition from the state except Aborting and Aborted state, to Aborting. (15)
Sts_Aborting_SC	BOOL	The request to execute the state transition from Aborting to Aborted. (16)
Cmd_Clear	BOOL	The command to execute the state transition from Aborted to Clearing. (17)
Sts_Clearing_SC	BOOL	The request to execute the state transition from Clearing to Stopped. (18)
Sts_Execute_SC	BOOL	The request to execute the state transition from Execute to Completing. (19)
Sts_Completing_SC	BOOL	The request to execute the state transition from Completing to Complete. (20)



Function

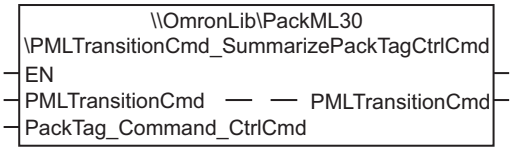
State transition requests are merged for the host module by processing the state transition requests (sPACKML_TRANSITION_COMMAND structure variable) arisen from the lower modules as described below:

- Execute OR evaluation on State transition commands (Cmd_<state name>)
- Execute AND evaluation on Wait state completion notifications (Sts_<state name>_SC)

This function is used for merging each of state transition requests of CM below EM into the state transitions of EM, and for merging each of state transition requests of EM into the state transition requests of UN.

PMLTransitionCmd_Summarize-PackTagCtrlCmd

This function executes OR evaluation on the state transition request commands coming from outside of the machine through the Command.CtrlCmd tag of PackTag, and reflects it to the state transition requests of the summarizing destination.

Function name	Name	FB/FUN	Graphic expression	ST expression
PMLTransitionCmd_SummarizePack-TagCtrlCmd	Pack Tag Transition Command	FUN		\\OmronLib \PackML30\PMLTransitionCmd_SummarizePack-TagCtrlCmd (PMLTransitionCmd, PackTag_Command_CtrlCmd);

Function Block and Function Information

Item	Description
Library file name	OmronLib_PackML_OPCUAxx_Vy_y.slr*1
Namespace	OmronLib\PackML30
Function block and function number	00132
Publish/Do not publish source code	Not published.
Function block and function version	1.00

*1. x indicates the version of the OPC UA for PackML specifications (example: in the case of 1.0.1, the upper two digits are used to indicate 10), and y indicates the version of the library.

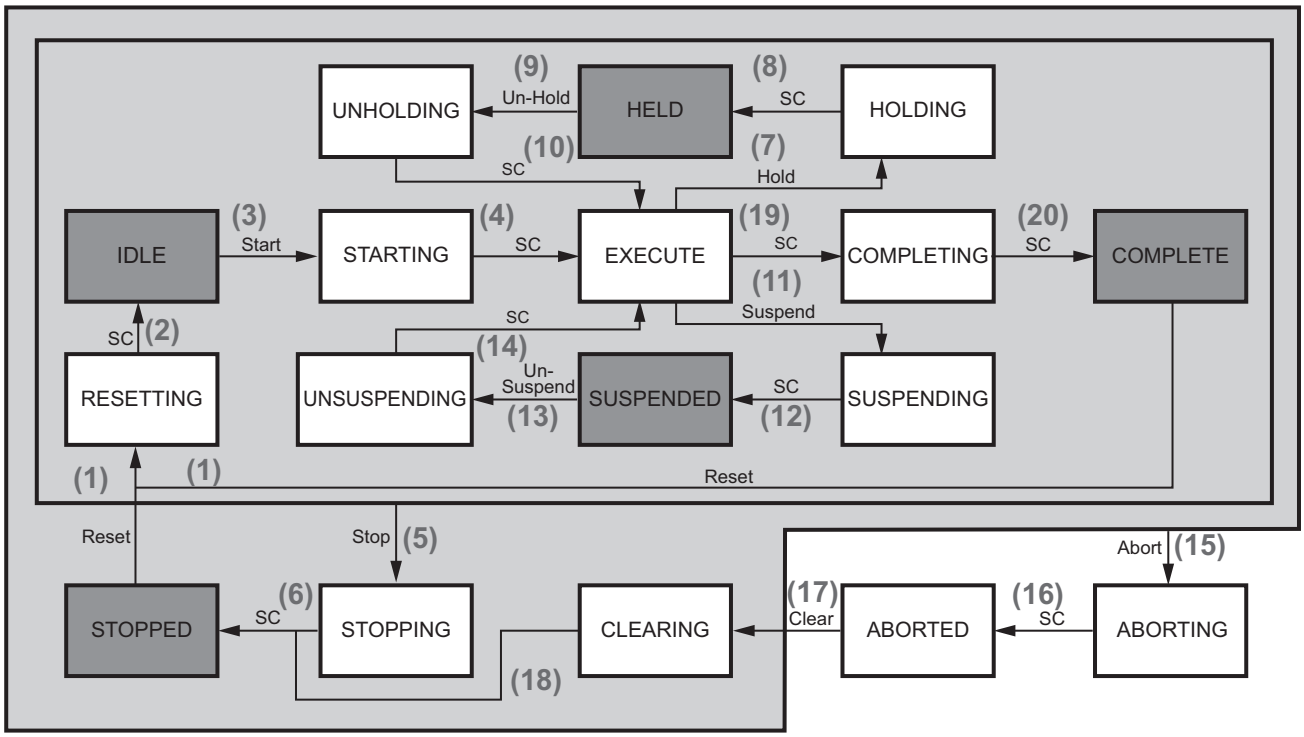
Variables

Name	I/O	Data type	Default	Description
EN	Input	BOOL	TRUE	Function Execution Control Flag When this variable is FALSE, internal logic does not operate even if it is called.
PMLTransitionCommand	In-out	OmronLib \PackML30\sPACKML_TRANSITION_COMMAND	---	Summarize Destination State Transition Request State transition requests from outside of the machine are to be merged.
PackTag_Command_CtrlCmd	Input	DINT	0	The state transition requests gained by PackTag-Command from outside of the machine. These gained state transition requests should be merged into PMLTransitionCommand.

Structures

● sPACKML_TRANSITION_COMMAND

Name	Data type	Description
sPACKML_TRANSITION_COMMAND	STRUCT	The structure that indicates a transition for the PackML state machine.
Cmd_Reset	BOOL	The command to execute the state transition from Stopped or Complete to Resetting. (1)
Sts_Resetting_SC	BOOL	The request to execute the state transition from Resetting to Idle. (2)
Cmd_Start	BOOL	The command to execute the state transition from Idle to Starting. (3)
Sts_Starting_SC	BOOL	The request to execute the state transition from Starting to Execute. (4)
Cmd_Stop	BOOL	The command to execute the state transition from Idle, Resetting, Starting, Execute, Completing, Complete, Holding, Held, Unholding, Suspending, Suspended, or Unsuspending to Stopping. (5)
Sts_Stopping_SC	BOOL	The request to execute the state transition from Stopping to Stopped. (6)
Cmd_Hold	BOOL	The command to execute the state transition from Execute to Holding. (7)
Sts_Holding_SC	BOOL	The request to execute the state transition from Holding to Held. (8)
Cmd_UnHold	BOOL	The command to execute the state transition from Held to UnHolding. (9)
Sts_UnHolding_SC	BOOL	The request to execute the state transition from UnHolding to Execute. (10)
Cmd_Suspend	BOOL	The command to execute the state transition to Execute to Suspending. (11)
Sts_Suspending_SC	BOOL	The request to execute the state transition from Suspending to Suspended. (12)
Cmd_UnSuspend	BOOL	The command to execute the state transition from Suspended to UnSuspending. (13)
Sts_UnSuspending_SC	BOOL	The request to execute the state transition from UnSuspending to Execute. (14)
Cmd_Abort	BOOL	The command to execute the state transition from the state except Aborting and Aborted state, to Aborting. (15)
Sts_Aborting_SC	BOOL	The request to execute the state transition from Aborting to Aborted. (16)
Cmd_Clear	BOOL	The command to execute the state transition from Aborted to Clearing. (17)
Sts_Clearing_SC	BOOL	The request to execute the state transition from Clearing to Stopped. (18)
Sts_Execute_SC	BOOL	The request to execute the state transition from Execute to Completing. (19)
Sts_Completing_SC	BOOL	The request to execute the state transition from Completing to Complete. (20)

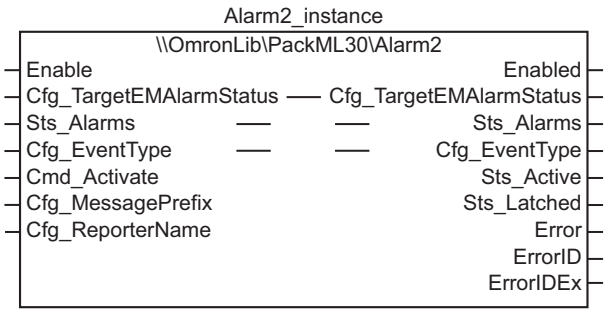


Function

The function merges the state transition requests from outside of the machine through the Command. CntrlCmd tag of PackTag, and the state transition requests gained in the machine by merging the state transition requests from EM and CM below UN.

Alarm2

This function block defines “Alarm” to support events and reports the state of the defined Alarm to the *sALARM_STATUS2* structure variable under the host module control. *Sts_Alarms* is a variable-length array.

FB/FUN name	Name	FB/FUN	Graphic expression	ST expression
Alarm2	Alarm 2	FB		Alarm2_instance(Enable, Cfg_TargetEMA- larmStatus, Sts_Alarms, Cfg_EventType, Cmd_Activate, Cfg_MessagePrefix, Cfg_ReporterName, Enabled, Sts_Active, Sts_Latched, Error, ErrorID, ErrorIDEx);

Function Block and Function Information

Item	Description
Library file name	OmronLib_PackML_OPCUAxx_Vy_y.slr*1
Namespace	OmronLib\PackML30
Function block and function number	00219
Publish/Do not publish source code	Not published.

*1. x indicates the version of the OPC UA for PackML specifications (example: in the case of 1.0.1, the upper two digits are used to indicate 10), and y indicates the version of the library.

Variables

Input Variables

Name	Data type	Description	Valid range	Unit	Default
Enable	BOOL	FB-enabled Flag Enables this function block. When this variable is FALSE, nothing executes.	TRUE or FALSE	---	FALSE

Name	Data type	Description	Valid range	Unit	Default
Cmd_Activate	BOOL	Alarm Activation Flag Input Sets TRUE when Alarm is activated after the Event occurs. To reset, sets FALSE.	TRUE or FALSE	---	FALSE
Cfg_MessagePrefix	STRING[10]	Alarm Message Prefix When reporting Alarm, specifies a prefix that should be attached to the message specified by Cfg_EventType.	10 bytes max. (9 single-byte alphanumeric characters plus the final NULL character)	---	''
Cfg_ReporterName	STRING[100]	Report Source Name Specifies the necessary name in order to identify the Alarm report source.	100 bytes max. (99 single-byte alphanumeric characters plus the final NULL character)	---	''

Output Variables

Name	Data type	Description	Valid range	Unit	Default
Enabled	BOOL	FB-enabled Flag Output It becomes TRUE when Enable becomes TRUE and this function block is operating normally.	TRUE or FALSE	---	---
Sts_Active	BOOL	Alarm Activation Flag Output It becomes TRUE when this Alarm is activated.	TRUE or FALSE	---	---
Sts_Latched	BOOL	Alarm Latch Flag Output When this Alarm is activated, it becomes TRUE. Even after being reset, it retains TRUE. When it is reset by the AlarmStatus_Update2 function, it goes back to FALSE.	TRUE or FALSE	---	---
Error	BOOL	Error Output	TRUE or FALSE	---	---
ErrorID	WORD	The value is 16#3CC8 while there is an error. The value is 16#0000 for a normal end.	16#0000 or 16#3CC8	---	---
ErrorIDEx	DWORD	The value is an expansion error code for an error end. The value is 16#00000000 for a normal end.	*1	---	---

*1. Refer to *Function* on page 4-41 for details.

In-Out Variables

Name	Data type	Description	Valid range	Unit	Default
Cfg_TargetEMAlarmStatus	OmronLib \PackML30\sALARM_STATUS2	Report Destination Alarm Status Specifies the sALARM_STATUS2 variable to which this Alarm status is reported. * Do not change the value while Enable is TRUE.	Depends on data type.	---	---
Sts_Alarms[]*1	ARRAY[*] OF OmronLib \PackML30\sALARM	Array of Alarms collected by the equipment module.	Depends on data type.	---	---
Cfg_EventType	OmronLib \PackML30\sEVENT_CFG	Event Type Specifies the event type to be supported as Alarm. * Do not change the value while Enable is TRUE.	Depends on data type.	---	---

*1. The maximum number of array elements is 500. The first number of array element should be 0.

Structures

● OmronLib\PackML30\sALARM_STATUS2

This is the structure that merges the states of Alarm collected per EM (equipment module).

Member name	Data type	Description
Sts_FirstOutAlarm	OmronLib\PackML30\sALARM	The snapshot of the first active Alarm.
Sts_FirstOutAlarmByCategory	ARRAY[0..9] OF OmronLib \PackML30\sALARM	The snapshot of the first active Alarm in each category.
Sts_NumOfAlarms	UINT	Number of data stored in Alarm array.
Sts_CategoryActiveFlag	ARRAY[0..9] OF BOOL	The array of the flag that shows whether each category includes active Alarm or not. The array element number represents the category number.
Sts_CategoryLatchedFlag	ARRAY[0..9] OF BOOL	Array of flag indicating whether each category includes any Latched Alarms (Alarms with evidence of having been active). The array element number represents the category number.
NeedToBeUpdated	BOOL	The flag that shows the necessity of updating data by the AlarmStatus_Update2 function because the state of Sts_Alarms[] is updated.
NeedToBeSummarized	BOOL	The flag that shows the necessity of updating AlarmSummation.

● OmronLib\PackML30\sALARM

This is the structure that represents a single Alarm.

Member name	Data type	Description
EventType	OmronLib\PackML30\sEVENT_CFG	Event type supported by this Alarm.
OccuredTime	DATE_AND_TIME	Event occurrence time.

Member name	Data type	Description
Active	BOOL	The flag that indicates whether this Alarm is active or not (not acknowledged yet).
Latched	BOOL	Flag indicating this Alarm has been active (i.e. a related event has occurred) since the last reset.
AcknowledgedTime	DATE_AND_TIME	Time when this Alarm became inactive.
ReporterName	STRING[256]	Information, which shows a source of Alarm, for debugging.

● OmronLib\PackML30\sEVENT_CFG

This is the structure that retains detail information of events that Alarm supports.

Member name	Data type	Description
ID	DINT	Identifier of event type
Value	DINT	Additional information of event type
Message	STRING[80]	Message to be indicated for event
Description	STRING[256]	Detailed description of event type
Category	USINT	Event category number. Range (0 to 9)

Function

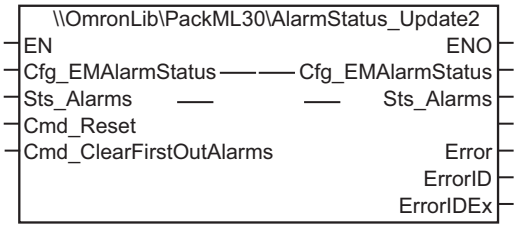
This function block defines "Alarm" to support events and reports the state of the defined Alarm to the *sALARM_STATUS2* structure variable under the host module control.

Operation Specification

- At first execution, this function block checks *Sts_NumOfAlarms* of *Cfg_TargetEMAlarmStatus*, retains the unused element numbers of *Sts_Alarms[]* as an internal variable; and increments *Sts_NumOfAlarms*.
- Afterwards, the function block writes content based on other input variables to the *sALARM* structure variable of *Sts_Alarms[]*, of the relevant array element number. It also changes *Cfg_TargetEMAlarmStatus.NeedToBeUpdated* to TRUE.
- The function block outputs to its output variable the corresponding member of the *sALARM* structure with its array element number.
- In the following cases, the function block changes the *Error* output variable to TRUE and outputs 16#3CC8 to *ErrorID*.
  a) A number other than 0 is assigned to the first number of *Sts_Alarms[]* array element, or the number of *Sts_Alarms[]* array elements exceeds 500 (ErrorIDEx=16#00000001).
  b) An instance of this function block has been executed more times than the number of *Sts_Alarms[]* array elements (ErrorIDEx=16#00000002).

AlarmStatus_Update2

This function checks *Cfg_EMAlarmStatus*, which indicates the status of Alarms collected to EM as in-out variables, to see whether the status of each Alarm has changed. The function then updates each member of *Cfg_EMAlarmStatus*.
Also, the function resets *Cfg_EMAlarmStatus* based on instructions given as input variables.
Sts_Alarms is a variable-length array.

FB/FUN name	Name	FB/FUN	Graphic expression	ST expression
AlarmStatus_Update2	EM Alarm Status Update 2	FUN		\\OmronLib\PackML30\AlarmStatus_Update2(Cfg_EMAlarmStatus, Sts_Alarms, Cmd_Reset, Cmd_ClearFirstOutAlarms, Error, ErrorID, ErrorIDEx);

Function Block and Function Information

Item	Description
Library file name	OmronLib_PackML_OPCUAxx_Vy_y.slr*1
Namespace	OmronLib\PackML30
Function block and function number	00220
Publish/Do not publish source code	Not published.

*1. x indicates the version of the OPC UA for PackML specifications (example: in the case of 1.0.1, the upper two digits are used to indicate 10), and y indicates the version of the library.

Variables

Input Variables

Name	Data type	Description	Valid range	Unit	Default
EN	BOOL	Function Execution Control Flag When this variable is TRUE, the internal code in this function is executed. When it is FALSE, nothing executes.	TRUE or FALSE	---	FALSE
Cmd_Reset	BOOL	Reset Command When this variable is TRUE, all information except Sts_FirstOutAlarm of the target Alarm status is reset.*1	TRUE or FALSE	---	FALSE

Name	Data type	Description	Valid range	Unit	Default
Cmd_ClearFirst-OutAlarms	BOOL	First Alarm Clear Command When this variable is TRUE, Sts_First-OutAlarm of the target Alarm status is cleared.	TRUE or FALSE	---	FALSE

*1. The following data is reset by *Cmd_Reset*.
- Sts_Alarms[] (except EventType)
- Cfg_EMAlarmStatus.Sts_CategoryActiveFlag
- Cfg_EMAlarmStatus.Sts_CategoryLatchedFlag

Output Variables

Name	Data type	Description	Valid range	Unit	Default
ENO	BOOL	Function Execution Control Flag Output EN is reflected as it is.	TRUE or FALSE	---	---
Error	BOOL	Error Output	TRUE or FALSE	---	---
ErrorID	WORD	The value is 16#3CC9 while there is an error. The value is 16#0000 for a normal end.	16#0000 or 16#3CC9	---	---
ErrorIDEx	DWORD	The value is an expansion error code for an error end. The value is 16#00000000 for a normal end.	*1	---	---

*1. Refer to *Function* on page 4-44 for details.

In-Out Variables

Name	Data type	Description	Valid range	Unit	Default
Cfg_EMAlarmStatus	OmronLib \PackML30\sALARM_STATUS2	Update Alarm Status Specifies the sALARM_STATUS2 structure variable to be updated by this function.	Depends on data type.	---	---
Sts_Alarms[]*1	ARRAY[*] OF OmronLib \PackML30\sALARM	Array of Alarms collected by the equipment module.	Depends on data type.	---	---

*1. The maximum number of array elements is 500. The first number of array element should be 0.

Structures

● OmronLib\PackML30\sALARM_STATUS2

This is the structure that merges the states of Alarm collected per EM (equipment module).

Member name	Data type	Description
Sts_FirstOutAlarm	OmronLib\PackML30\sALARM	The snapshot of the first active Alarm.
Sts_FirstOutAlarmByCategory	ARRAY[0..9] OF OmronLib \PackML30\sALARM	The snapshot of the first active Alarm in each category.
Sts_NumOfAlarms	UINT	Number of data stored in Alarm array.

Member name	Data type	Description
Sts_CategoryActiveFlag	ARRAY[0..9] OF BOOL	The array of the flag that shows whether each category includes active Alarm or not. The array element number represents the category number.
Sts_CategoryLatchedFlag	ARRAY[0..9] OF BOOL	Array of flag indicating whether each category includes any Latched Alarms (Alarms with evidence of having been active). The array element number represents the category number.
NeedToBeUpdated	BOOL	The flag that shows the necessity of updating data by the AlarmStatus_Update2 function because the state of Sts_Alarms[] is updated.
NeedToBeSummarized	BOOL	The flag that shows the necessity of updating AlarmSummation.

● OmronLib\PackML30\sALARM

This is the structure that represents a single Alarm.

Member name	Data type	Description
EventType	OmronLib\PackML30\sEVENT_CFG	Event type supported by this Alarm.
OccuredTime	DATE_AND_TIME	Event occurrence time.
Active	BOOL	The flag that indicates whether this Alarm is active or not (not acknowledged yet).
Latched	BOOL	Flag indicating this Alarm has been active (i.e. a related event has occurred) since the last reset.
AcknowledgedTime	DATE_AND_TIME	Time when this Alarm became inactive.
ReporterName	STRING[256]	Information, which shows a source of Alarm, for debugging.

● OmronLib\PackML30\sEVENT_CFG

This is the structure that retains detail information of events that Alarm supports.

Member name	Data type	Description
ID	DINT	Identifier of event type
Value	DINT	Additional information of event type
Message	STRING[80]	Message to be indicated for event
Description	STRING[256]	Detailed description of event type
Category	USINT	Event category number. Range (0 to 9)

Function

This function checks *Cfg_EMAlarmStatus*, which indicates the status of Alarms collected to EM as in-out variables, to see whether the status of each Alarm has changed. The function then updates each member of *Cfg_EMAlarmStatus*.

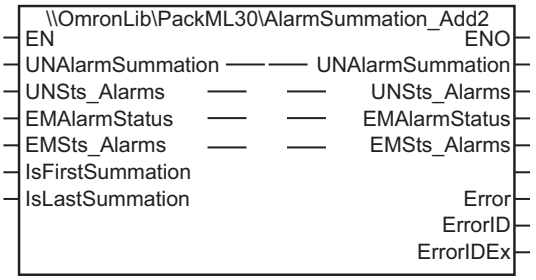
Also, the function resets *Cfg_EMAlarmStatus* based on instructions given as input variables.

Operation Specification

- When *Cmd_Reset* is TRUE, the function changes *Active* and *Latched* of each *Cfg_EMAlarmStatus.Sts_Alarms* element to FALSE.
- When *Cmd_ClearFirstOutAlarm* is TRUE, the function clears *Cfg_EMAlarmStatus.Sts_FirstOutAlarm* and *Cfg_EMAlarmStatus.Sts_FirstOutAlarmByCategory*.
- When *Cfg_EMAlarmStatus.NeedToBeUpdated* is TRUE, the function block updates each *Cfg_EMAlarmStatus* member.
- In the following cases, the function changes the *Error* output variable to TRUE and outputs 16#3CC9 to *ErrorID*.
  a) A number other than 0 is assigned to the first number of *Sts_Alarms[]* array element, or the number of *Sts_Alarms[]* array elements exceeds 500, or the number of *Sts_Alarms[]* array elements is less than *Cfg_EMAlarmStatus.Sts_NumOfAlarms* (ErrorIDEx=16#00000001).

AlarmSummation_Add2

This function adds the specific EM Alarm status given by the *EMAlarmStatus* in-out variable to the *UNAlarmSummation* in-out variable retaining the Alarm statuses merged to UN (unit/machine). *UNSts_Alarms* and *EMSts_Alarms* are variable-length arrays.

FB/FUN name	Name	FB/FUN	Graphic expression	ST expression
AlarmSummation_Add2	UN Alarm Status Add 2	FUN		\\OmronLib \PackML30\AlarmSummation_Add2(UNAlarmSummation, UNSts_Alarms, EMAlarmStatus, EMSts_Alarms, IsFirstSummation, IsLastSummation, Error, ErrorID, ErrorIDEx);

Function Block and Function Information

Item	Description
Library file name	OmronLib_PackML_OPCUAxx_Vy_y.slr*1
Namespace	OmronLib\PackML30
Function block and function number	00221
Publish/Do not publish source code	Not published.

*1. x indicates the version of the OPC UA for PackML specifications (example: in the case of 1.0.1, the upper two digits are used to indicate 10), and y indicates the version of the library.

Variables

Input Variables

Name	Data type	Description	Valid range	Unit	Default
EN	BOOL	Function Execution Control Flag When this variable is TRUE, the internal code in this function is executed. When it is FALSE, nothing executes.	TRUE or FALSE	---	FALSE

Name	Data type	Description	Valid range	Unit	Default
IsFirstSummation	BOOL	Set to TRUE when the first EM-level status is added to the machine-level alarm status. When this input variable is TRUE, UNAlarmSummation is cleared and then EMAlarmStatus is added on the top. When this input variable is FALSE, it is added to the end of the existing valid array size.	TRUE or FALSE	---	FALSE
IsLastSummation	BOOL	Set to TRUE when the last EM-level status is added to the machine-level alarm status. When this input variable is TRUE, UNAlarmSummation members are updated as necessary after EMAlarmStatus is added to UNAlarmSummation. While multiple EMAlarmStatus are added, setting this input variable to FALSE allows you to speed up the process by skipping unnecessary update procedures.	TRUE or FALSE	---	FALSE

Output Variables

Name	Data type	Description	Valid range	Unit	Default
ENO	BOOL	Function Execution Control Flag Output EN is reflected as it is.	TRUE or FALSE	---	---
Error	BOOL	Error Output	TRUE or FALSE	---	---
ErrorID	WORD	The value is 16#3CCA while there is an error. The value is 16#0000 for a normal end.	16#0000 or 16#3CCA	---	---
ErrorIDEx	DWORD	The value is an expansion error code for an error end. The value is 16#00000000 for a normal end.	*1	---	---

*1. Refer to *Function* on page 4-49 for details.

In-Out Variables

Name	Data type	Description	Valid range	Unit	Default
UNAlarmSummation	OmronLib \PackML30\sALARM_SUMMATION2	Update Alarm Status Specifies the sALARM_SUMMATION2 structure variable to be updated by this function.	Depends on data type.	---	---
UNSts_Alarms[]*1	ARRAY[*] OF OmronLib \PackML30\sALARM	Array of update machine-level alarm status.	Depends on data type.	---	---

Name	Data type	Description	Valid range	Unit	Default
EMAlarmStatus	OmronLib \PackML30\sALARM_STATUS2	The EM-level alarm status to add to the machine-level alarm status.	Depends on data type.	---	---
EMSts_Alarms[]*1	ARRAY[*] OF OmronLib \PackML30\sALARM	Array of EM alarm status to add.	Depends on data type.	---	---

*1. The maximum number of array elements is 500. The first number of array element should be 0.

Structures

● OmronLib\PackML30\sALARM_SUMMATION2

The structure that merges Alarms collected from all EM below UN (unit/machine).

Member name	Data type	Description
Sts_FirstOutAlarm	OmronLib\PackML30\sALARM	The first active Alarm.
Sts_NumOfAlarms	UINT	The number of data stored in the array of update machine-level alarm status.
ActiveOneExists	BOOL	Flag indicating whether each category includes an active Alarm.
LatchedOneExists	BOOL	Flag indicating whether each category includes any Latched Alarms (Alarms with evidence of having been active).
Sts_CategoryActiveFlag	ARRAY[0..9] OF BOOL	The array of the flag that shows whether each category includes active Alarm or not. The array element number represents the category number.
Sts_CategoryLatchedFlag	ARRAY[0..9] OF BOOL	Array of flag indicating whether each category includes any Latched Alarms (Alarms with evidence of having been active). The array element number represents the category number.

● OmronLib\PackML30\sALARM_STATUS2

This is the structure that merges the states of Alarm collected per EM (equipment module).

Member name	Data type	Description
Sts_FirstOutAlarm	OmronLib\PackML30\sALARM	The snapshot of the first active Alarm.
Sts_FirstOutAlarmByCategory	ARRAY[0..9] OF OmronLib \PackML30\sALARM	The snapshot of the first active Alarm in each category.
Sts_NumOfAlarms	UINT	Number of data stored in Alarm array.
Sts_CategoryActiveFlag	ARRAY[0..9] OF BOOL	The array of the flag that shows whether each category includes active Alarm or not. The array element number represents the category number.
Sts_CategoryLatchedFlag	ARRAY[0..9] OF BOOL	Array of flag indicating whether each category includes any Latched Alarms (Alarms with evidence of having been active). The array element number represents the category number.
NeedToBeUpdated	BOOL	The flag that shows the necessity of updating data by the AlarmStatus_Update2 function because the state of Sts_Alarms[] is updated.

Member name	Data type	Description
NeedToBeSummarized	BOOL	The flag that shows the necessity of updating Alarm-Summation.

● OmronLib\PackML30\sALARM

This is the structure that represents a single Alarm.

Member name	Data type	Description
EventType	OmronLib\PackML3\sEV-ENT_CFG	Event type supported by this Alarm.
OccuredTime	DATE_AND_TIME	Event occurrence time.
Active	BOOL	The flag that indicates whether this Alarm is active or not (not acknowledged yet).
Latched	BOOL	Flag indicating this Alarm has been active (i.e. a related event has occurred) since the last reset.
AcknowledgedTime	DATE_AND_TIME	Time when this Alarm became inactive.
ReporterName	STRING[256]	Information, which shows a source of Alarm, for debugging.

● OmronLib\PackML30\sEVENT_CFG

This is the structure that retains detail information of events that Alarm supports.

Member name	Data type	Description
ID	DINT	Identifier of event type
Value	DINT	Additional information of event type
Message	STRING[80]	Message to be indicated for event
Description	STRING[256]	Detailed description of event type
Category	USINT	Event category number. Range (0 to 9)

Function

This function adds the specific EM Alarm status given by the *EMAlarmStatus* in-out variable to the *UNAlarmSummation* in-out variable retaining the Alarm statuses merged to UN (unit/machine).

Operation Specification

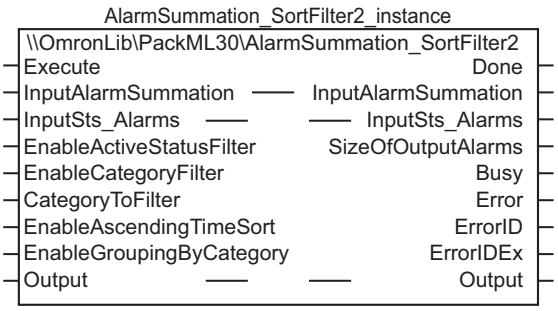
This function performs following operations when EN is TRUE.

- When *IsFirstSummation* is TRUE, the function clears each *UNAlarmSummation* value and stores Alarms retained by *EMAlarmStatus* to *UNSts_Alarms[]*, beginning with its first element.
  When *IsFirstSummation* is FALSE, the function does not clear *UNAlarmSummation* and stores Alarms retained by *EMAlarmStatus* to *UNSts_Alarms[]*, beginning with its last element.
- When *IsLastSummation* is TRUE, the function adds *EMAlarmStatus* to *UNAlarmSummation*, and then updates the value of each *UNAlarmSummation* member.
- In the following cases, the function changes the *Error* output variable to TRUE and outputs 16#3CCA to *ErrorID*.
  a) A number other than 0 is assigned to the first number of *UNSts_Alarms[]* array element, or the number of *UNSts_Alarms[]* array elements exceeds 500 (ErrorIDEx=16#00000001).

b) A number other than 0 is assigned to the first number of *EMSts_Alarms[]* array element, the number of *EMSts_Alarms[]* array elements exceeds 500, or the number of *EMSts_Alarms[]* array elements is less than *Cfg_EMAlarmStatus.Sts_NumOfAlarms* (ErrorIDEx=16#00000002).

AlarmSummation_SortFilter2

This function block reflects the results of filtering and sorting that are conducted with the conditions specified by the *InputAlarmSummation* in-out variable that retains the Alarm statuses merged into UN (unit/machine), to the sALARM array variable *Output*. *InputSts_Alarms* is a variable-length array.

FB/FUN name	Name	FB/FUN	Graphic expression	ST expression
Alarm-Summa-tion_Sort Filter2	Alarm Sort and Filter 2	FB		AlarmSummation_SortFil-ter2_instance(Execute, InputAlarmSummation, InputSts_Alarms, EnableActiveStatusFilter, EnableCategoryFilter, CategoryToFilter, EnableAscendingTime-Sort, EnableGroupingByCate-gory, Output, Done, SizeOfOutputAlarms, Busy, Error, ErrorID, ErrorIDEx);

Function Block and Function Information

Item	Description
Library file name	OmronLib_PackML_OPCUAxx_Vy_y.slr*1
Namespace	OmronLib\PackML30
Function block and function number	00222
Publish/Do not publish source code	Not published.

*1. x indicates the version of the OPC UA for PackML specifications (example: in the case of 1.0.1, the upper two digits are used to indicate 10), and y indicates the version of the library.

Variables

Input Variables

Name	Data type	Description	Valid range	Unit	Default
Execute	BOOL	FB Execution Flag TRUE: The processing is executed. FALSE: The processing is not executed.	TRUE or FALSE	---	FALSE
EnableActiveStatusFilter	BOOL	Active Alarm Filter Enabled Flag When this variable is TRUE, only the Alarms whose Active are TRUE are output to Output. To reset, sets FALSE.	TRUE or FALSE	---	FALSE
EnableCategoryFilter	BOOL	Category Filter Enabled Flag When this variable is TRUE, only Alarms of categories specified by CategoryToFilter input variable are output to Output.	TRUE or FALSE	---	FALSE
CategoryToFilter	USINT	Category Number for Filter Specifies the category number to be output by the category filter.	0 to 9	---	0
EnableAscendingTimeSort	BOOL	Time Ascending Flag When this variable is TRUE, sorted Alarm occurrence time in ascending order is output to Output.	TRUE or FALSE	---	FALSE
EnableGroupingByCategory	BOOL	Categorized Grouping Flag When this variable is TRUE, sorted group numbers in ascending order are output to Output.	TRUE or FALSE	---	FALSE

Output Variables

Name	Data type	Description	Valid range	Unit	Default
Done	BOOL	Processing Completed Flag After processing is complete, the value remains TRUE while Execute is TRUE.	TRUE or FALSE	---	---
SizeOfOutputAlarms	INT	Output the number of alarm that is stored to Output[].	0 to 499	---	---
Busy	BOOL	Processing Busy Flag The value is TRUE from the time Execute changes to TRUE until processing is complete.	TRUE or FALSE	---	---
Error	BOOL	Error Output	TRUE or FALSE	---	---
ErrorID	WORD	The value is 16#3CCB while there is an error. The value is 16#0000 for a normal end.	16#0000 or 16#3CCB	---	---
ErrorIDEx	DWORD	The value is an expansion error code for an error end. The value is 16#00000000 for a normal end.	*1	---	---

*1. Refer to *Function* on page 4-54 for details.

In-Out Variables

Name	Data type	Description	Valid range	Unit	Default
InputAlarmSummation	OmronLib \PackML30\sALARM_SUMMATION2	Source Alarm Information The machine-level sALARM_SUMMATION2 structure variable.	Depends on data type.	---	---
InputSts_Alarms[]*1	ARRAY[*] OF OmronLib \PackML30\sALARM	Array of source alarm information of filtering and sorting.	Depends on data type.	---	---
Output[]*1	ARRAY[*] OF OmronLib \PackML30\sALARM	Alarm array to which the sorted/filtered results are output.	Depends on data type.	---	---

*1. The maximum number of array elements is 500. The first number of array element should be 0.

Structures

● OmronLib\PackML30\sALARM_SUMMATION2

The structure that merges Alarms collected from all EM below UN (unit/machine).

Member name	Data type	Description
Sts_FirstOutAlarm	OmronLib\PackML30\sALARM	The first active Alarm.
Sts_NumOfAlarms	UINT	Flag indicating the number of data records stored in the array of update machine-level alarm status.
ActiveOneExists	BOOL	Flag indicating whether each category includes an active Alarm.
LatchedOneExists	BOOL	Flag indicating whether each category includes any Latched Alarms (Alarms with evidence of having been active).
Sts_CategoryActiveFlag	ARRAY[0..9] OF BOOL	The array of the flag that shows whether each category includes active Alarm or not. The array element number represents the category number.
Sts_CategoryLatchedFlag	ARRAY[0..9] OF BOOL	Array of flag indicating whether each category includes any Latched Alarms (Alarms with evidence of having been active). The array element number represents the category number.

● OmronLib\PackML30\sALARM

This is the structure that represents a single Alarm.

Member name	Data type	Description
EventType	OmronLib\PackML30\sEVENT_CFG	Event type supported by this Alarm.
OccuredTime	DATE_AND_TIME	Event occurrence time.
Active	BOOL	The flag that indicates whether this Alarm is active or not (not acknowledged yet).
Latched	BOOL	Flag indicating this Alarm has been active (i.e. a related event has occurred) since the last reset.

Member name	Data type	Description
AcknowledgedTime	DATE_AND_TIME	Time when this Alarm became inactive.
ReporterName	STRING[256]	Information, which shows a source of Alarm, for debugging.

- **OmronLib\PackML30\sEVENT_CFG**

  This is the structure that retains detail information of events that Alarm supports.

Member name	Data type	Description
ID	DINT	Identifier of event type
Value	DINT	Additional information of event type
Message	STRING[80]	Message to be indicated for event
Description	STRING[256]	Detailed description of event type
Category	USINT	Event category number. Range (0 to 9)

Function

This function block reflects the results of filtering and sorting that are conducted with the conditions specified by the *InputAlarmSummation* in-out variable that retains the Alarm statuses merged into UN (unit/machine), to the sALARM array variable *Output[]*.

Filtering

When *Execute* changes to TRUE, the function block executes as follows:

- When *EnableActiveStatusFilter* is TRUE, only Alarms of *InputSts_Alarms[]* whose *Active* are TRUE are stored to *Output[]*.
- When *EnableCategoryFilter* is TRUE, only Alarms of *InputSts_Alarms[]* whose categories are specified by *CategoryToFilter* are stored to the *Output[]* array.
- When both *EnableActiveStatusFilter* and *EnableCategoryFilter* are TRUE, Alarms whose *Active* are TRUE and whose categories are specified by *CategoryToFilter* are stored to *Output[]*.

Sorting

When *Execute* changes to TRUE, the function block executes as follows:

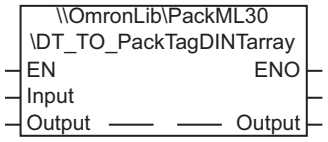
- When *EnableAscendingTimeSort* is TRUE, *InputSts_Alarms[]* is sorted in ascending order of their times of occurrence and output to *Output[]*.
- When *EnableGroupingByCategory* is TRUE, *InputSts_Alarms[]* is sorted in ascending order of their category numbers and output to *Output[]*.
- When both *EnableAscendingTimeSort* and *EnableGroupingByCategory* are TRUE, *InputSts_Alarms[]* is first sorted in ascending order of their times of occurrence, and then sorted in ascending order of their category numbers to output to *Output[]*.

In the following cases, the function block changes the *Error* output variable to TRUE and outputs 16#3CCB to *ErrorID*.

- A number other than 0 is assigned to the first number of *InputSts_Alarms[]* array element, the number of *InputSts_Alarms[]* array elements exceeds 500, or the number of *InputSts_Alarms[]* array elements is less than the number of *InputAlarmSummation.Sts_NumOfAlarms* (ErrorIDEx=16#00000001).
- A number other than 0 is assigned to the first number of *Output[]* array element, or the number of *Output[]* elements exceeds 500 (ErrorIDEx=16#00000002).
- The number of *InputSts_Alarms[]* array elements exceeds the number of *Output[]* array elements (ErrorIDEx=16#00000003).

DT_TO_PackTagDINTarray

This function converts the DATE_AND_TIME input variable into the array variable specified by PackTags.

Function name	Name	FB/FUN	Graphic expression	ST expression
DT_TO_PackTagDINTarray	DATE_AND_TIME Pack Tag Array Conversion	FUN		\\OmronLib \PackML30\DT_TO_PackTagDINTarray (Input, Output);

Function Block and Function Information

Item	Description
Library file name	OmronLib_PackML_OPCUAxx_Vy_y.slr*1
Namespace	OmronLib\PackML30
Function block and function number	00137
Publish/Do not publish source code	Not published.
Function block and function version	1.00

*1. x indicates the version of the OPC UA for PackML specifications (example: in the case of 1.0.1, the upper two digits are used to indicate 10), and y indicates the version of the library.

Variables

Name	I/O	Data type	Default	Description
EN	Input	BOOL	TRUE	Execution Start Flag When this variable is TRUE, convert *Input* data to the DINT array variable and output it to *Output*.
Input	Input	DATE_AND_TIME	0	Conversion Source Data
Output	In-out	ARRAY[0..6] OF DINT	---	Conversion Destination Data Convert *Input* in the following format. Array element 0 = Year Array element 1 = Month Array element 2 = Day Array element 3 = Hour (24hr format) Array element 4 = Min Array element 5 = Sec Array element 6 = USec (1/1,000,000 sec)

Function

This function converts the DATE_AND_TIME input variable into the array variable specified by PackTags.

I

Index

Index

A

D

P

OMRON Corporation **Industrial Automation Company**

Kyoto, JAPAN **Contact : www.ia.omron.com**

Regional Headquarters

OMRON EUROPE B.V.
Wegalaan 67-69, 2132 JD Hoofddorp
The Netherlands
Tel: (31) 2356-81-300 Fax: (31) 2356-81-388

OMRON ELECTRONICS LLC
2895 Greenspoint Parkway, Suite 200
Hoffman Estates, IL 60169 U.S.A.
Tel: (1) 847-843-7900 Fax: (1) 847-843-7787

OMRON ASIA PACIFIC PTE. LTD.
438B Alexandra Road, #08-01/02 Alexandra
Technopark, Singapore 119968
Tel: (65) 6835-3011 Fax: (65) 6835-3011

OMRON (CHINA) CO., LTD.
Room 2211, Bank of China Tower,
200 Yin Cheng Zhong Road,
PuDong New Area, Shanghai, 200120, China
Tel: (86) 21-6023-0333 Fax: (86) 21-5037-2388

Authorized Distributor:

Cat. No. W638-E1-02 1023