NOTE

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.

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. 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, Windows Vista, Excel, and Visual Basic 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 reprinted with permission from Microsoft Corporation.
Introduction

Thank you for purchasing an NJ/NX-series CPU Unit.

This manual contains information that is necessary to use the function blocks in the Safety System Monitor Library. ("Function block" is sometimes abbreviated as “FB”.) Please read this manual and make sure you understand the functionality and performance of the NJ/NX-series CPU Unit 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.

Refer to the user’s manuals for all of the products in the application before you use any of the products.

Keep this manual in a safe place where it will be available for reference during operation.

Features of the Library

The Safety System Monitor Library is used to monitor safety system information.

You can use the Safety System Monitor Library to correctly manage information on the safety system during operation.

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, NY-series Industrial PC, and the Sysmac Studio that this library supports, refer to Sysmac Library Version Information in the SYSMAC-XRlightly Sysmac Library Catalog (Cat. No. P102). This catalog can be downloaded from the OMRON website (http://www.ia.omron.com/products/family/3459/download/catalog.html).
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 or 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 Sysmac Studio are given.

Note References are provided to more detailed or related information.
## CONTENTS

**Introduction** ......................................................................................................................... 1  
  Features of the Library...................................................................................................................... 1  
  Intended Audience............................................................................................................................ 1  
  Applicable Products.......................................................................................................................... 1  

**Manual Structure** .................................................................................................................... 2  
  Special Information ......................................................................................................................... 2  

**CONTENTS** ............................................................................................................................... 4  

**Terms and Conditions Agreement** ............................................................................................ 6  
  Warranty, Limitations of Liability.......................................................................................................... 6  
  Application Considerations.................................................................................................................. 7  
  Disclaimers......................................................................................................................................... 7  

**Safety Precautions** ................................................................................................................... 8  
  Definition of Precautionary Information............................................................................................... 8  
  Symbols............................................................................................................................................... 8  
  Cautions............................................................................................................................................. 9  

**Precautions for Correct Use** ...................................................................................................... 10  

**Related Manuals** ..................................................................................................................... 11  

**Revision History** ....................................................................................................................... 13  

**Procedure to Use Sysmac Libraries** ........................................................................................ 15  
  Procedure to Use Sysmac Libraries Installed Using the Installer...................................................... 16  
  Procedure to Use Sysmac Libraries Uploaded from a CPU Unit...................................................... 20  

**Common Specifications of Function Blocks** ......................................................................... 23  
  Common Variables............................................................................................................................. 24  
  Precautions....................................................................................................................................... 30  

**Individual Specifications of Function Blocks** ...................................................................... 31  
  GetSafetySignature............................................................................................................................ 32  

**Appendix** .................................................................................................................................. 39  
  Referring to Library Information........................................................................................................ 40  
  Referring to Function Block and Function Source Codes................................................................. 43
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 http://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.
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 an NJ/NX-series Controller.

The safety precautions that are provided are extremely important to safety. Always read and heed the information provided in all safety precautions.

The following notation is used.

| ![WARNING](warning.png) | Indicates a potentially hazardous situation which, if not avoided, could result in death or serious injury. Additionally, there may be severe property damage. |
| ![Caution](caution.png) | 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 prohibiting disassembly.

- 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.
Cautions

⚠️ 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.

The sample programming shows only the portion of a program that uses the function or function block from the library.

When using sample programming in actual devices, also program safety circuits, device interlocks, I/O with other devices, and other control procedures.
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.
• Do not change the file names and the contents of any object list.
• Do not turn OFF the power supply to the Controller or EtherCAT slave or stop EtherCAT communications until the processing for this function block ends normally or ends in an error.

Using Sample Programming

• Create a user program that will produce the intended device operation.
• Check the user program for proper execution before you use it for actual operation.
## Related Manuals

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

<table>
<thead>
<tr>
<th>Manual name</th>
<th>Cat. No.</th>
<th>Model numbers</th>
<th>Application</th>
<th>Description</th>
</tr>
</thead>
</table>
| NX-series CPU Unit Hardware User’s Manual | W535 | NX701-□□□□□ | Learning the basic specifications of the NX-series CPU Units, including introductory information, designing, installation, and maintenance. Mainly hardware information is provided. | An introduction to the entire NX-series 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  
Use this manual together with the **NJ/NX-series CPU Unit Software User’s Manual** (Cat. No. W501). |
| 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-series CPU Unit Hardware User’s Manual | W500 | NJ501-□□□□□  
NJ301-□□□□□  
NJ101-□□□□□ | Learning the basic specifications of the NJ-series CPU Units, including introductory information, designing, installation, and maintenance. Mainly hardware information is provided. | An introduction to the entire NJ-series 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  
Use this manual together with the **NJ/NX-series CPU Unit Software User’s Manual** (Cat. No. W501). |
| NY-series IPC Machine Controller Industrial Panel PC Hardware User’s Manual | W557 | NY532-□□□□□ | Learning the basic specifications of the NY-series Industrial Panel PCs, including introductory information, designing, installation, and maintenance. Mainly hardware information is provided. | An introduction to the entire NY-series system is provided along with the following information on the Industrial Panel PC.  
• Features and system configuration  
• Introduction  
• Part names and functions  
• General specifications  
• Installation and wiring  
• Maintenance and inspection |
| NY-series IPC Machine Controller Industrial Box PC Hardware User’s Manual | W556 | NY512-□□□□□ | Learning the basic specifications of the NY-series Industrial Box PCs, including introductory information, designing, installation, and maintenance. Mainly hardware information is provided. | An introduction to the entire NY-series system is provided along with the following information on the Industrial Box PC.  
• Features and system configuration  
• Introduction  
• Part names and functions  
• General specifications  
• Installation and wiring  
• Maintenance and inspection |
<table>
<thead>
<tr>
<th>Manual name</th>
<th>Cat. No.</th>
<th>Model numbers</th>
<th>Application</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>NJ/NX-series CPU Unit Software User’s Manual</td>
<td>W501</td>
<td>NX701-□□□□</td>
<td>Learning how to program and set up an NJ/NX-series CPU Unit. Mainly software information is provided.</td>
<td>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 Use this manual together with the NX-series CPU Unit Hardware User’s Manual (Cat. No. W535) or NJ-series CPU Unit Hardware User’s Manual (Cat. No. W500).</td>
</tr>
<tr>
<td>NY-series IPC Machine Controller</td>
<td>W558</td>
<td>NY532-□□□□</td>
<td>Learning how to program and set up the Controller functions of an NY-series Industrial PC.</td>
<td>The following information is provided on the NY-series Controller functions. • Controller operation • Controller features • Controller settings • Programming based on IEC 61131-3 language specifications</td>
</tr>
<tr>
<td>NY-series Instructions Reference Manual</td>
<td>W560</td>
<td>NY532-□□□□</td>
<td>Learning detailed specifications on the basic instructions of an NY-series Industrial PC.</td>
<td>The instructions in the instruction set (IEC 61131-3 specifications) are described.</td>
</tr>
<tr>
<td>NX-series EtherCAT® Coupler Unit User’s Manual</td>
<td>W519</td>
<td>NX-ECC20□</td>
<td>Learning how to use an NX-series EtherCAT Coupler Unit and EtherCAT Slave Terminals.</td>
<td>The following items are described: the overall system and configuration methods of an EtherCAT Slave Terminal (which consists of an NX-series EtherCAT Coupler Unit and NX Units), and information on hardware, setup, and functions to set up, control, and monitor NX Units through EtherCAT.</td>
</tr>
<tr>
<td>NX-series Safety Control Unit User’s Manual</td>
<td>Z930</td>
<td>NX-SL□□□□□□</td>
<td>Learning how to use NX-series Safety Control Units</td>
<td>Describes the hardware, setup methods, and functions of the NX-series Safety Control Units.</td>
</tr>
<tr>
<td>Sysmac Studio Version 1 Operation Manual</td>
<td>W504</td>
<td>SYSMAC-□□□□</td>
<td>Learning about the operating procedures and functions of the Sysmac Studio.</td>
<td>Describes the operating procedures of the Sysmac Studio.</td>
</tr>
<tr>
<td>CNC Operator Operation Manual</td>
<td>0032</td>
<td>SYSMAC-□□□□□</td>
<td>Learning an introduction of the CNC Operator and how to use it.</td>
<td>An introduction of the CNC Operator, installation procedures, basic operations, connection operations, and operating procedures for main functions are described.</td>
</tr>
</tbody>
</table>

12

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

**Cat. No. W582-E1-02**

<table>
<thead>
<tr>
<th>Revision code</th>
<th>Date</th>
<th>Revised content</th>
</tr>
</thead>
<tbody>
<tr>
<td>01</td>
<td>September 2016</td>
<td>Original production</td>
</tr>
<tr>
<td>02</td>
<td>January 2019</td>
<td>Added compatible models.</td>
</tr>
</tbody>
</table>
Procedure to Use Sysmac Libraries
Procedure to Use Sysmac Libraries Installed Using the Installer

This section describes the procedure to use Sysmac Libraries that you installed using the installer. There are two ways to use libraries.
- Using newly installed Sysmac Libraries
- Using upgraded Sysmac Libraries

**Version Information**

To use Sysmac Libraries, you need the Sysmac Studio version 1.14 or higher.

### Using Newly Installed Libraries

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

   ![New Project Image](image)

   **Precautions for Correct Use**

   If you create a new project, be sure to configure the settings as follows to enable the use of Sysmac Libraries. If you do not configure the following settings, you cannot proceed to the step 2 and later steps.
   - Set the project type to Standard Project or Library Project.
   - Set the device category to Controller.
   - Set the device version to 1.01 or later.
2 Select **Project** – **Library** – **Show References**.

![Image](image1.png)

**Precautions for Correct Use**

If you have more than one registered device in the project, make sure that the device selected currently is an NJ/NX-series CPU Unit. If you do not select an NJ/NX-series CPU Unit as the device, Library References does not appear in the above menu. When the device selected currently is an NJ/NX-series CPU Unit, the device icon ![Device Icon](device_icon.png) is displayed in the Multiview Explorer.

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

![Image](image2.png)

The Sysmac Library file is read into the project.

Now, when you select the Ladder Editor or ST Editor, the function blocks and functions included in a 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 or function in the Toolbox and drag and drop it onto the programming editor.

![Image](image3.png)
• Right-click the programming editor, select Insert Function Block in the menu, and enter the fully qualified name (\name of namespace\name of function block).

Using Upgraded Libraries

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

2 Select Project – Library – Show References.

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

4 Add the desired Sysmac Library to the list and click the OK Button.

Precautions for Correct Use

If you have more than one registered device in the project, make sure that the device selected currently is an NJ/NX-series CPU Unit. Otherwise, Library References does not appear in the above menu. When the device selected currently is an NJ/NX-series CPU Unit, the device icon is displayed in the Multiview Explorer.
Precautions for Correct Use

After you upgrade the Sysmac Studio, check all programs and make sure that there is no error of the program check results on the Build Tab Page.

Select Project – Check All Programs from the Main Menu.
Procedure to Use Sysmac Libraries Uploaded from a CPU Unit

You can use Sysmac Libraries uploaded from a CPU Unit to your computer if they are not installed.
The procedure to use uploaded Sysmac Libraries from a CPU Unit is as follows.

**Version Information**

To use Sysmac Libraries, you need the Sysmac Studio version 1.14 or higher.

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

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

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

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 or function in the Toolbox and drag and drop it onto the Ladder Editor.
• Right-click the programming editor, select **Insert Function Block** in the menu, and enter the fully qualified name (`\name of namespace\name of function block`).

---

**Precautions for Correct Use**

• The Sysmac Studio installs library files of the uploaded Sysmac Studio to the specified folder on the computer if they are not present. However, the Sysmac Studio does not install library files 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 Libraries from a CPU Unit does not install the manual and help files for the Sysmac Libraries, unlike the case where you install them using the installer. Please install the manual and help files using the installer if you need them.
Common Specifications of Function Blocks
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.

**Definition of Input Variables and Output Variables**

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

<table>
<thead>
<tr>
<th>Variable</th>
<th>I/O</th>
<th>Data type</th>
<th>Function/function block type to use</th>
<th>Meaning</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>EN</td>
<td>Input</td>
<td>BOOL</td>
<td>Execute-type Enable-type Function</td>
<td>Execute</td>
<td>The processing is executed while the variable is TRUE.</td>
</tr>
<tr>
<td>Execute</td>
<td>OK</td>
<td>Execute</td>
<td>OK</td>
<td>Execute</td>
<td>The processing is executed when the variable changes to TRUE.</td>
</tr>
<tr>
<td>Enable</td>
<td>OK</td>
<td>Run</td>
<td>OK</td>
<td>Run</td>
<td>The processing is executed while the variable is TRUE.</td>
</tr>
<tr>
<td>Abort</td>
<td>OK</td>
<td>Abort</td>
<td>OK</td>
<td>Abort</td>
<td>The processing is aborted. You can select the aborting method.</td>
</tr>
</tbody>
</table>
Common Variables

<table>
<thead>
<tr>
<th>Variable</th>
<th>I/O</th>
<th>Data type</th>
<th>Function/function block type to use</th>
<th>Meaning</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>ENO</td>
<td>Output</td>
<td>BOOL</td>
<td>Execute-type</td>
<td>OK</td>
<td>Done</td>
</tr>
<tr>
<td>Done</td>
<td></td>
<td>BOOL</td>
<td>OK</td>
<td></td>
<td>Done</td>
</tr>
<tr>
<td>Busy</td>
<td></td>
<td>BOOL</td>
<td>OK</td>
<td>OK</td>
<td>Executing</td>
</tr>
<tr>
<td>CalcRslt</td>
<td></td>
<td>LREAL</td>
<td>OK</td>
<td></td>
<td>Calculation Result</td>
</tr>
<tr>
<td>Enabled</td>
<td></td>
<td>BOOL</td>
<td>OK</td>
<td></td>
<td>Enabled</td>
</tr>
<tr>
<td>Command Aborted</td>
<td></td>
<td>BOOL</td>
<td>OK</td>
<td></td>
<td>Command Aborted</td>
</tr>
<tr>
<td>Error</td>
<td></td>
<td>BOOL</td>
<td>OK</td>
<td>OK</td>
<td>Error</td>
</tr>
<tr>
<td>ErrorID</td>
<td></td>
<td>WORD</td>
<td>OK</td>
<td>OK</td>
<td>Error Code</td>
</tr>
<tr>
<td>ErrorIDEx</td>
<td></td>
<td>DWORD</td>
<td>OK</td>
<td>OK</td>
<td>Expansion Error Code</td>
</tr>
</tbody>
</table>

**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, 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, aborted execution, and errors.

#### Normal End

<table>
<thead>
<tr>
<th>Execute</th>
<th>Done</th>
<th>Busy</th>
<th>CommandAborted</th>
<th>Error</th>
<th>ErrorID</th>
<th>ErrorIDEx</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>(16#0000)</td>
<td>(16#0000\ 0000)</td>
</tr>
</tbody>
</table>

#### Canceled Execution

<table>
<thead>
<tr>
<th>Execute</th>
<th>Abort</th>
<th>Busy</th>
<th>CommandAborted</th>
<th>Error</th>
<th>ErrorID</th>
<th>ErrorIDEx</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>(16#0000)</td>
<td>(16#0000\ 0000)</td>
</tr>
</tbody>
</table>
**Aborted Execution**

- Execute
- Done
- Busy
- CommandAborted
- Error
- ErrorID: 16#0000
- ErrorIDEx: 16#00000000

**Errors**

- Execute
- Done
- Busy
- CommandAborted
- Error
- ErrorID: 16#0000
- ErrorID: 16#0000
- ErrorIDEx: 16#00000000
- ErrorIDEx: 16#00000000
- ErrorIDEx: 16#00000000
- ErrorIDEx: 16#00000000
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, 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

![Timing Chart](image)
**Errors**

- Enable
- Enabled
- CalcRslt
- Busy
- Error
- ErrorID
- ErrorIDEx

<table>
<thead>
<tr>
<th>Enable</th>
<th>Enabled</th>
<th>CalcRslt</th>
<th>Busy</th>
<th>Error</th>
<th>ErrorID</th>
<th>ErrorIDEx</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>16#0000</td>
<td>16#00000000</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- Retained
- 16#0000
- 16#00000000
- 16#00000000
Precautions

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

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).

Instruction Options

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

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).
Individual Specifications of Function Blocks

<table>
<thead>
<tr>
<th>Function block name</th>
<th>Name</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>GetSafetySignature</td>
<td>Read Safety Signature</td>
<td>P.32</td>
</tr>
</tbody>
</table>
The GetSafetySignature function block reads the safety signature from the Safety CPU Unit specified on the EtherCAT Network.

<table>
<thead>
<tr>
<th>Function block name</th>
<th>Name</th>
<th>FB/FUN</th>
<th>Graphic expression</th>
<th>ST expression</th>
</tr>
</thead>
<tbody>
<tr>
<td>GetSafetySignature</td>
<td>Read Safety Signature</td>
<td>FB</td>
<td>GetSafetySignature_Instance()</td>
<td>GetSafetySignature_Instance(Execute, NodeAddress, NXUnitNumber, Done, SignatureCode, SignatureLastModified, Busy, Error, ErrorID, ErrorIDEx);</td>
</tr>
</tbody>
</table>

### Library Information

<table>
<thead>
<tr>
<th>Item</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Library file name</td>
<td>OmronLib_SF_Monitor_V1_0.slr</td>
</tr>
<tr>
<td>Namespace</td>
<td>OmronLib\SF_Monitor</td>
</tr>
<tr>
<td>Function block and function number</td>
<td>00165</td>
</tr>
<tr>
<td>Source code published/not published</td>
<td>Not published</td>
</tr>
<tr>
<td>Function block and function version</td>
<td>1.00</td>
</tr>
</tbody>
</table>

### Compatible Models

<table>
<thead>
<tr>
<th>Item</th>
<th>Name</th>
<th>Model numbers</th>
<th>Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>Automation Software</td>
<td>Sysmac Studio</td>
<td>SYSMAC-SE</td>
<td>Version 1.14 or higher</td>
</tr>
<tr>
<td>Device</td>
<td>CPU Unit</td>
<td>NX701</td>
<td>Version 1.10 or later</td>
</tr>
<tr>
<td></td>
<td></td>
<td>NJ101</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>NJ501</td>
<td>Version 1.07 or later</td>
</tr>
<tr>
<td></td>
<td></td>
<td>NJ301</td>
<td></td>
</tr>
<tr>
<td>IPC Motion Controller</td>
<td>NY532-1</td>
<td></td>
<td>Version 1.12 or later</td>
</tr>
<tr>
<td></td>
<td></td>
<td>NY512</td>
<td></td>
</tr>
<tr>
<td>Safety CPU Unit</td>
<td>NX-SL3</td>
<td></td>
<td>Version 1.00 or later</td>
</tr>
<tr>
<td>EtherCAT Coupler Unit</td>
<td>NX-ECC201</td>
<td></td>
<td>Version 1.2.1 or later</td>
</tr>
<tr>
<td></td>
<td>NX-ECC202</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>NX-ECC203</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Variables

- GetSafetySignature_Instance
- Execute
- NodeAddress
- NXUnitNumber
- Signature
- SignatureLastModified
- Busy
- Error
- ErrorID
- ErrorIDEx
- NodeAddress
- NXUnitNumber
GetSafetySignature

### Name | Meaning | I/O | Description | Valid range | Unit | Default
--- | --- | --- | --- | --- | --- | ---
**Execute** | Execute | Input | Execute <br>TRUE: Executes the function block. FALSE: Does not execute the function block. | Depends on data type. | --- | FALSE
**NodeAddress** | Slave Node Address | Input | Specifies the node address of the EtherCAT Coupler Unit to which the Safety CPU Unit is mounted. | 1 to 512<sup>1</sup> | --- | 1
**NXUnitNumber** | NX Unit Number | Input | Specifies the NX Unit number of the Safety CPU Unit. | 1 to 63 | --- | 1
**Done** | Done | Output | Done <br>TRUE: Normal end FALSE: Error end, execution in progress, or execution condition not met | Depends on data type. | --- | ---
**Signature** | Safety Signature | Output | Safety signature <br>16#0000: Safety validation has not been performed. Other than 16#0000: Safety signature | Depends on data type. | --- | ---
**LastModified** | Safety Signature Date Created | Output | Specifies the date the safety signature was created in UTC. | Depends on data type. | --- | ---
**Busy** | Executing | Output | Executing <br>TRUE: Execution processing is in progress. FALSE: Execution processing is not in progress. | Depends on data type. | --- | ---
**Error** | Error | Output | Error <br>TRUE: Error end FALSE: Normal end, execution in progress, or execution condition not met | Depends on data type. | --- | ---
**ErrorID** | Error Code | Output | This is the error ID for an error end. The value is WORD#16#0 for a normal end. | *2 | --- | ---
**ErrorIDE** | Expansion Error Code | Output | This is the error ID for an Expansion Unit Hardware Error. The value is DWORD#16#0 for a normal end. | *2 | --- | ---

---

*1. The valid range is 1 to 192 for NJ-series CPU Units.

*2. For details, refer to Troubleshooting on page 34.

### Function

- The GetSafetySignature function block reads the safety signature and safety signature date created for the Safety CPU Unit specified with **NodeAddress** and **NXUnitNumber**.
- For **NodeAddress**, specify the node address of the NX-ECC20 EtherCAT Communications Coupler Unit to which the Safety CPU Unit is mounted.
**Related System-defined Variables**

<table>
<thead>
<tr>
<th>Name</th>
<th>Meaning</th>
<th>Data type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>_EC_MBXSavTbl</td>
<td>Message Communications Enabled Slave Table</td>
<td>Array[1..512(^1)] OF BOOL</td>
<td>This table indicates the slaves that can perform message communications. Slaves are given in the table in the order of slave node addresses. TRUE: Message communications are possible. FALSE: Message communications are not possible.</td>
</tr>
</tbody>
</table>

\(^1\) The valid range is 1 to 192 for NJ-series CPU Units.

**Precautions for Correct Use**

- Execution of this function block will be continued until processing is ended even if the value of *Execute* changes to FALSE or the execution time exceeds the task period. The value of *Done* changes to TRUE when processing is ended. Use this to confirm normal ending of processing.
- This function block executes one EC_CoESDOWrite instruction or one EC_CoESDORead instruction within the part. Refer to the EC_CoESDOWrite instruction in the *NJ/NX-series Instructions Reference Manual* (Cat. No. W502) for the maximum number of instructions that can be executed at the same time.
- Do not turn OFF the power supply to the Controller or EtherCAT slave or stop EtherCAT communications until the processing for this function block ends normally or ends in an error.
- An error will occur in the following cases. *Error* will change to TRUE.
  a) The EtherCAT master is not in a state that allows message communications.
  b) The slave specified with *NodeAddress* does not exist.
  c) The slave specified with *NodeAddress* is not in a state that allows communications.
  d) The number of EtherCAT communications instructions exceeded the maximum number of instructions that can be executed at the same time.
- If safety validation has not been performed for the Safety CPU Unit specified by this function block, the value of Signature will be 16#0000.
- The date created for the safety signature that was read with this function block is the UTC value. Change it to the local time as required.

**Troubleshooting**

<table>
<thead>
<tr>
<th>Error code</th>
<th>Expansion error code</th>
<th>Status</th>
<th>Description</th>
<th>Correction</th>
</tr>
</thead>
<tbody>
<tr>
<td>16#0000</td>
<td>16#00000000</td>
<td>Normal End</td>
<td>---</td>
<td>---</td>
</tr>
<tr>
<td>16#0400</td>
<td>16#00000000</td>
<td>Input Value Out of Range</td>
<td>The value of <em>NodeAddress</em> is out of range.</td>
<td>Set the value of <em>NodeAddress</em> correctly.</td>
</tr>
<tr>
<td>16#1800</td>
<td>16#00000000</td>
<td>EtherCAT Communications Error</td>
<td>The EtherCAT network is not in a usable status.</td>
<td>Check the operation status of the EtherCAT network by checking the status of the EtherCAT master. Use this information to correct the cause of the problem.</td>
</tr>
</tbody>
</table>
### Sample Programming

This sample programming reads the safety signature for the specified Safety CPU Unit.

#### Conditions

- The Safety CPU Unit is mounted at the following node address and NX Unit number.

  Node address  | 10  
  NX Unit number | 1  

- Safety validation has been performed for the Safety CPU Unit.

#### Processing

1. Make sure that the target slave is in a state that allows communications and that the same instance is not being executed.

---

<table>
<thead>
<tr>
<th>Error code</th>
<th>Expansion error code</th>
<th>Status</th>
<th>Description</th>
<th>Correction</th>
</tr>
</thead>
<tbody>
<tr>
<td>16#1801</td>
<td>16#00000000</td>
<td>EtherCAT Slave Does Not Respond</td>
<td>The target slave does not exist.</td>
<td>Specify an existing node address.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>The target slave is not in a usable status.</td>
<td>Check the status of the target EtherCAT slave. Make sure that the target slave is in a usable status.</td>
</tr>
<tr>
<td>16#1802</td>
<td>16#00000000</td>
<td>EtherCAT Time-out</td>
<td>Communications with the target slave timed out.</td>
<td>Check the operating status of the target slave and correct the cause of the problem.</td>
</tr>
<tr>
<td>16#1803</td>
<td>16#00000000</td>
<td>Reception Buffer Overflow</td>
<td>The receive data from an EtherCAT slave overflowed the receive buffer.</td>
<td>Set the size of the reception buffer to a value larger than the size of the receive data from the slave.</td>
</tr>
<tr>
<td>16#1804</td>
<td>SDO Abort Code*1</td>
<td>SDO Abort Error</td>
<td>The cause of this error depends on the specifications of the EtherCAT slave.</td>
<td>Refer to the manual for the EtherCAT slave and correct the problem.</td>
</tr>
<tr>
<td>16#1808</td>
<td>16#00000000</td>
<td>Communications Resource Overflow</td>
<td>More than 32 EtherCAT communications instructions were executed at the same time.</td>
<td>Correct the user program so that no more than 32 EtherCAT communications instructions are executed at the same time.</td>
</tr>
<tr>
<td>16#3CE8</td>
<td>16#00000001</td>
<td>Unsupported Slave</td>
<td>The communications coupler does not support this function block.</td>
<td>Check that the specified node address is for an NX-ECC20 EtherCAT Communications Coupler.</td>
</tr>
<tr>
<td>16#00000002</td>
<td>Unsupported Version</td>
<td>The version of the communications coupler does not support this function block.</td>
<td>Confirm that the specified communications coupler version is listed in Compatible Models on page 32.</td>
<td></td>
</tr>
<tr>
<td>16#00000003</td>
<td>NX Unit Does Not Exist</td>
<td>The target NX Unit does not exist.</td>
<td>Check that the Unit configuration is correct.</td>
<td></td>
</tr>
<tr>
<td>16#00000004</td>
<td>Unsupported NX Unit</td>
<td>The device does not support this function block.</td>
<td>Check that the device with the specified node address and NX Unit Number is an NX-SL3 Safety CPU Unit.</td>
<td></td>
</tr>
</tbody>
</table>

*1. Refer to SDO Abort Codes in the NJ/NX-series Instructions Reference Manual (Cat. No. W502) for a list of SDO abort codes.*
2. Execute GetSafetySignature_instance.

```LD

- **Internal Variables**

<table>
<thead>
<tr>
<th>Name</th>
<th>Data type</th>
<th>Default</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>GetSafetySignature_instance</td>
<td>OmronLib\SF_Monitor\GetSafetySignature</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Init_Trigger</td>
<td>BOOL</td>
<td></td>
<td>GetSafetySignature execution condition</td>
</tr>
<tr>
<td>GetSafetySignature_Trigger</td>
<td>BOOL</td>
<td></td>
<td>GetSafetySignature execution condition</td>
</tr>
<tr>
<td>GetSafetySignature_Busy</td>
<td>BOOL</td>
<td></td>
<td>GetSafetySignature execution condition</td>
</tr>
<tr>
<td>NodeAddress</td>
<td>UINT</td>
<td>10</td>
<td>Target Node Address</td>
</tr>
<tr>
<td>NXUnitNumber</td>
<td>USINT</td>
<td>1</td>
<td>Target NX Unit Address</td>
</tr>
<tr>
<td>SignatureCode</td>
<td>WORD</td>
<td></td>
<td>Safety Signature: CRC data</td>
</tr>
<tr>
<td>SignatureLastModified</td>
<td>DATE_AND_TIME</td>
<td></td>
<td>Safety Signature: Last Modified</td>
</tr>
</tbody>
</table>

- **External Variables**

<table>
<thead>
<tr>
<th>Name</th>
<th>Data type</th>
<th>Constant</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>_EC_MBXSlavTbl</td>
<td>ARRAY[1..N *1] OF BOOL</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>


```ST

- **Internal Variables**

<table>
<thead>
<tr>
<th>Name</th>
<th>Data type</th>
<th>Default</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>GetSafetySignature_instance</td>
<td>OmronLib\SF_Monitor\GetSafetySignature</td>
<td></td>
<td></td>
</tr>
<tr>
<td>GetSafetySignature_Trigger</td>
<td>BOOL</td>
<td></td>
<td>GetSafetySignature execution condition</td>
</tr>
<tr>
<td>GetSafetySignature_Busy</td>
<td>BOOL</td>
<td></td>
<td>GetSafetySignature execution condition</td>
</tr>
<tr>
<td>NodeAddress</td>
<td>UINT</td>
<td>10</td>
<td>Target Node Address</td>
</tr>
<tr>
<td>NXUnitNumber</td>
<td>USINT</td>
<td>1</td>
<td>Target NX Unit Address</td>
</tr>
</tbody>
</table>
GetSafetySignature

External Variables

<table>
<thead>
<tr>
<th>Name</th>
<th>Data type</th>
<th>Default</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>SignatureCode</td>
<td>WORD</td>
<td></td>
<td>Safety Signature: CRC data</td>
</tr>
<tr>
<td>SignatureLastModified</td>
<td>DATE_AND_TIME</td>
<td></td>
<td>Safety Signature: Last Modified</td>
</tr>
<tr>
<td>LastTrigger</td>
<td>BOOL</td>
<td></td>
<td></td>
</tr>
<tr>
<td>OperatingStart</td>
<td>BOOL</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Operating</td>
<td>BOOL</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>


```plaintext
// Detect when Trigger changes to TRUE.
IF (GetSafetySignature_Trigger=TRUE) AND (LastTrigger=FALSE) AND
(_EC_MBXSlavTbl[NodeAddress]=TRUE) AND (GetSafetySignature_Busy=FALSE) THEN
  NodeAddress:=UINT#10;
  NXUnitNumber:=USINT#1;
  OperatingStart:=TRUE;
END_IF;
LastTrigger:=GetSafetySignature_Trigger;

// Initialize GetSafetySignature instruction.
IF (OperatingStart=TRUE) THEN
  Operating:=TRUE;
  OperatingStart:=FALSE;
END_IF;

// Execute GetSafetySignature instruction.
GetSafetySignature_instance(Execute:=Operating,
  NodeAddress:=NodeAddress,
  NXUnitNumber:=NXUnitNumber,
  Signature:=SignatureCode,
  LastModified:=SignatureLastModified,
  Busy:=GetSafetySignature_Busy);
IF (Operating=TRUE) AND (GetSafetySignature_instance.Done=TRUE) THEN
  // Processing after normal end.
  Operating:=FALSE;
ELSIF (Operating=TRUE) AND (GetSafetySignature_instance.Error=TRUE) THEN
  // Processing after error end.
  Operating:=FALSE;
END_IF;
```
Appendix
Referring to Library Information

When you make an inquiry to OMRON about the library, you can refer to the library information to identify the library to ask about.

The library information is useful in identifying the target library among the libraries provided by OMRON or created by the user.

The library information consists of the attributes of the library and the attributes of function blocks and functions contained in the library.

- Attributes of libraries
  Information for identifying the library itself
- Attributes of function blocks and functions
  Information for identifying the function block and function contained in the library

Use the Sysmac Studio to access the library information.

Attributes of Libraries, Function Blocks and Functions

The following attributes of libraries, function blocks and functions are provided as the library information.

- **Attributes of Libraries**

<table>
<thead>
<tr>
<th>No.*1</th>
<th>Attribute</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>(1)</td>
<td>Library file name</td>
<td>The name of the library file</td>
</tr>
<tr>
<td>(2)</td>
<td>Library version</td>
<td>The version of the library</td>
</tr>
<tr>
<td>(3)</td>
<td>Author</td>
<td>The name of creator of the library</td>
</tr>
<tr>
<td>(4)</td>
<td>Comment</td>
<td>The description of the library*2</td>
</tr>
</tbody>
</table>

*1. These numbers correspond to the numbers shown on the screen images in the next section, Referring to Attributes of Libraries, Function Blocks and Functions on page 41.

*2. It is provided in English and Japanese.

- **Attributes of Function Blocks and Functions**

<table>
<thead>
<tr>
<th>No.*1</th>
<th>Attribute</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>(5)</td>
<td>FB/FUN name</td>
<td>The name of the function block or function</td>
</tr>
<tr>
<td>(6)</td>
<td>Name space</td>
<td>The name of name space for the function block or function</td>
</tr>
<tr>
<td>(7)</td>
<td>FB/FUN version</td>
<td>The version of the function block or function</td>
</tr>
<tr>
<td>(8)</td>
<td>Author</td>
<td>The name of creator of the function block or function</td>
</tr>
<tr>
<td>(9)</td>
<td>FB/FUN number</td>
<td>The function block number or function number</td>
</tr>
<tr>
<td>(10)</td>
<td>Comment</td>
<td>The description of the function block or function*2</td>
</tr>
</tbody>
</table>

*1. These numbers correspond to the numbers shown on the screen images in the next section, Referring to Attributes of Libraries, Function Blocks and Functions on page 41.

*2. It is provided in English and Japanese.
Referring to Attributes of Libraries, Function Blocks and Functions

You can refer to the attributes of libraries, function blocks and functions of the library information at the following locations on the Sysmac Studio.

- Library Reference Dialog Box
- Toolbox Pane
- Ladder Editor

(a) Library Reference Dialog Box

When you refer to the libraries, the library information is displayed at the locations shown below.

1. Library file name
2. Library version
3. Library author
4. Library comment
5. FB/FUN name
6. Name space
7. FB/FUN version
8. FB/FUN author
9. FB/FUN comment
(b) Toolbox Pane
Select a function block and function to display its library information at the bottom of the Toolbox Pane.
The text “by OMRON” which is shown on the right of the library name (1) indicates that this library was provided by OMRON.

(c) Ladder Editor
Place the mouse on a function block and function to display the library information in a tooltip.
Referring to Function Block and Function Source Codes

You can refer to the source codes of function blocks and functions provided by OMRON to customize them to suit the user’s environment.

User function blocks and user functions can be created based on the copies of these source codes. The following are the examples of items that you may need to customize.

- Customizing the size of arrays to suit the memory capacity of the user’s Controller
- Customizing the data types to suit the user-defined data types

Note that you can access only function blocks and functions whose Source code published/not published is set to Published in the library information shown in their individual specifications.

Use the following procedure to refer to the source codes of function blocks and functions.

1. Select a function block or function in the program.

2. Double-click or right-click and select To Lower Layer from the menu. The source code is displayed.

Precautions for Correct Use

For function blocks and functions whose source codes are not published, the following dialog box is displayed in the above step 2. Click the Cancel button.
Authorized Distributor:

In the interest of product improvement, specifications are subject to change without notice.

Cat. No. W582-E1-02
© OMRON Corporation 2016-2019 All Rights Reserved. In the interest of product improvement, specifications are subject to change without notice.