EcoStruxure Machine Expert

DataLogging

Library Guide

EIO0000002938.02
11/2025

www.se.com

Legal Information

The information provided in this document contains general descriptions, technical characteristics and/or recommendations related to products/solutions.

This document is not intended as a substitute for a detailed study or operational and site-specific development or schematic plan. It is not to be used for determining suitability or reliability of the products/solutions for specific user applications. It is the duty of any such user to perform or have any professional expert of its choice (integrator, specifier or the like) perform the appropriate and comprehensive risk analysis, evaluation and testing of the products/solutions with respect to the relevant specific application or use thereof.

The Schneider Electric brand and any trademarks of Schneider Electric SE and its subsidiaries referred to in this document are the property of Schneider Electric SE or its subsidiaries. All other brands may be trademarks of their respective owner.

Schneider Electric does not grant any right or license for commercial use of the document or its content, except for a non-exclusive and personal license to consult it on an "as is" basis.

Schneider Electric reserves the right to make changes or updates with respect to or in the content of this document or the format thereof, at any time without notice.

To the extent permitted by applicable law, no responsibility or liability is assumed by Schneider Electric and its subsidiaries for any errors or omissions in the informational content of this document, as well as any non-intended use or misuse of the content thereof.

Table of Contents

Safety Information

Important Information

Read these instructions carefully, and look at the equipment to become familiar with the device before trying to install, operate, service, or maintain it. The following special messages may appear throughout this documentation or on the equipment to warn of potential hazards or to call attention to information that clarifies or simplifies a procedure.

The addition of this symbol to a “Danger” or “Warning” safety label indicates that an electrical hazard exists which will result in personal injury if the instructions are not followed.

This is the safety alert symbol. It is used to alert you to potential personal injury hazards. Obey all safety messages that follow this symbol to avoid possible injury or death.

⚠ DANGER

DANGER indicates a hazardous situation which, if not avoided, **will result in** death or serious injury.

⚠ WARNING

WARNING indicates a hazardous situation which, if not avoided, **could result in** death or serious injury.

⚠ CAUTION

CAUTION indicates a hazardous situation which, if not avoided, **could result** in minor or moderate injury.

NOTICE

NOTICE is used to address practices not related to physical injury.

Please Note

Electrical equipment should be installed, operated, serviced, and maintained only by qualified personnel. No responsibility is assumed by Schneider Electric for any consequences arising out of the use of this material.

A qualified person is one who has skills and knowledge related to the construction and operation of electrical equipment and its installation, and has received safety training to recognize and avoid the hazards involved.

Qualification of Personnel

A qualified person is one who has the following qualifications:

- Skills and knowledge related to the construction and operation of electrical equipment and the installation.
- Knowledge and experience in industrial control programming.
- Received safety-related training to recognize and avoid the hazards involved.

The qualified person must be able to detect possible hazards that may arise from parameterization, modifying parameter values and generally from mechanical, electrical, or electronic equipment. The qualified person must be familiar with the standards, provisions, and regulations for the prevention of industrial accidents, which they must observe when designing and implementing the system.

Proper Use

This product is a library to be used together with the control systems and long stator motor segments intended solely for the purposes as described in the present documentation as applied in the industrial sector.

Always observe the applicable safety-related instructions, the specified conditions, and the technical data.

Perform a risk evaluation concerning the specific use before using the product. Take protective measures according to the result.

Since the product is used as a part of an overall system, you must ensure the safety of the personnel by means of the design of this overall system (for example, machine design).

Any other use is not intended and may be hazardous.

Before You Begin

Do not use this product on machinery lacking effective point-of-operation guarding. Lack of effective point-of-operation guarding on a machine can result in serious injury to the operator of that machine.

⚠WARNING

UNGUARDED EQUIPMENT

- Do not use this software and related automation equipment on equipment which does not have point-of-operation protection.
- Do not reach into machinery during operation.

Failure to follow these instructions can result in death, serious injury, or equipment damage.

This automation equipment and related software is used to control a variety of industrial processes. The type or model of automation equipment suitable for each application will vary depending on factors such as the control function required, degree of protection required, production methods, unusual conditions, government regulations, etc. In some applications, more than one processor may be required, as when backup redundancy is needed.

Only you, the user, machine builder or system integrator can be aware of all the conditions and factors present during setup, operation, and maintenance of the machine and, therefore, can determine the automation equipment and the related safeties and interlocks which can be properly used. When selecting automation and control equipment and related software for a particular application, you should refer to the applicable local and national standards and regulations. The National Safety Council's Accident Prevention Manual (nationally recognized in the United States of America) also provides much useful information.

In some applications, such as packaging machinery, additional operator protection such as point-of-operation guarding must be provided. This is necessary if the operator's hands and other parts of the body are free to enter the pinch points or other hazardous areas and serious injury can occur. Software products alone cannot protect an operator from injury. For this reason the software cannot be substituted for or take the place of point-of-operation protection.

Ensure that appropriate safeties and mechanical/electrical interlocks related to point-of-operation protection have been installed and are operational before placing the equipment into service. All interlocks and safeties related to point-of-operation protection must be coordinated with the related automation equipment and software programming.

NOTE: Coordination of safeties and mechanical/electrical interlocks for point-of-operation protection is outside the scope of the Function Block Library, System User Guide, or other implementation referenced in this documentation.

Start-up and Test

Before using electrical control and automation equipment for regular operation after installation, the system should be given a start-up test by qualified personnel to verify correct operation of the equipment. It is important that arrangements for such a check are made and that enough time is allowed to perform complete and satisfactory testing.

⚠WARNING

EQUIPMENT OPERATION HAZARD

- Verify that all installation and set up procedures have been completed.
- Before operational tests are performed, remove all blocks or other temporary holding means used for shipment from all component devices.
- Remove tools, meters, and debris from equipment.

Failure to follow these instructions can result in death, serious injury, or equipment damage.

Follow all start-up tests recommended in the equipment documentation. Store all equipment documentation for future references.

Software testing must be done in both simulated and real environments.

Verify that the completed system is free from all short circuits and temporary grounds that are not installed according to local regulations (according to the National Electrical Code in the U.S.A, for instance). If high-potential voltage testing is necessary, follow recommendations in equipment documentation to prevent accidental equipment damage.

Before energizing equipment:

- Remove tools, meters, and debris from equipment.
- Close the equipment enclosure door.
- Remove all temporary grounds from incoming power lines.
- Perform all start-up tests recommended by the manufacturer.

Operation and Adjustments

The following precautions are from the NEMA Standards Publication ICS 7.1-1995:

(In case of divergence or contradiction between any translation and the English original, the original text in the English language will prevail.)

- Regardless of the care exercised in the design and manufacture of equipment or in the selection and ratings of components, there are hazards that can be encountered if such equipment is improperly operated.

- It is sometimes possible to misadjust the equipment and thus produce unsatisfactory or unsafe operation. Always use the manufacturer’s instructions as a guide for functional adjustments. Personnel who have access to these adjustments should be familiar with the equipment manufacturer’s instructions and the machinery used with the electrical equipment.
- Only those operational adjustments required by the operator should be accessible to the operator. Access to other controls should be restricted to prevent unauthorized changes in operating characteristics.

About the Document

Document Scope

This document explains the data logging functionality for controllers that support file management operations.

Validity Note

This document has been updated for the release of EcoStruxure™ Machine Expert V2.5.

Product Related Information

⚠WARNING

LOSS OF CONTROL

- Perform a Failure Mode and Effects Analysis (FMEA), or equivalent risk analysis, of your application, and apply preventive and detective controls before implementation.
- Provide a fallback state for undesired control events or sequences.
- Provide separate or redundant control paths wherever required.
- Supply appropriate parameters, particularly for limits.
- Review the implications of transmission delays and take actions to mitigate them.
- Review the implications of communication link interruptions and take actions to mitigate them.
- Provide independent paths for control functions (for example, emergency stop, over-limit conditions, and error conditions) according to your risk assessment, and applicable codes and regulations.
- Apply local accident prevention and safety regulations and guidelines.[1]
- Test each implementation of a system for proper operation before placing it into service.

Failure to follow these instructions can result in death, serious injury, or equipment damage.

[1] For additional information, refer to NEMA ICS 1.1 (latest edition), *Safety Guidelines for the Application, Installation, and Maintenance of Solid State Control* and to NEMA ICS 7.1 (latest edition), *Safety Standards for Construction and Guide for Selection, Installation and Operation of Adjustable-Speed Drive Systems* or their equivalent governing your particular location.

Before you attempt to provide a solution (machine or process) for a specific application using the POUs found in the library, you must consider, conduct and complete best practices. These practices include, but are not limited to, risk analysis, functional safety, component compatibility, testing and system validation as they relate to this library.

⚠WARNING

IMPROPER USE OF PROGRAM ORGANIZATION UNITS

- Perform a safety-related analysis for the application and the devices installed.
- Ensure that the Program Organization Units (POUs) are compatible with the devices in the system and have no unintended effects on the proper functioning of the system.
- Ensure that the axis is homed and that the homing is valid before usage of absolute movements or POUs using absolute movements.
- Use appropriate parameters, especially limit values, and observe machine wear and stop behavior.
- Verify that the sensors and actuators are compatible with the selected POUs.
- Thoroughly test all functions during verification and commissioning in all operation modes.
- Provide independent methods for critical control functions (emergency stop, conditions for limit values being exceeded, etc.) according to a safety-related analysis, respective rules, and regulations.

Failure to follow these instructions can result in death, serious injury, or equipment damage.

General Cybersecurity Information

In recent years, the growing number of networked machines and production plants has seen a corresponding increase in the potential for cyber threats, such as unauthorized access, data breaches, and operational disruptions. You must, therefore, consider all possible cybersecurity measures to help protect assets and systems against such threats.

To help keep your Schneider Electric products secure and protected, it is in your best interest to implement the cybersecurity best practices as described in the Cybersecurity Best Practices document.

Schneider Electric provides additional information and assistance:

- Subscribe to the Schneider Electric security newsletter.
- Visit the Cybersecurity Support Portal web page to:
  - Find Security Notifications.
  - Report vulnerabilities and incidents.
- Visit the Schneider Electric Cybersecurity and Data Protection Posture web page to:
  - Access the cybersecurity posture.
  - Learn more about cybersecurity in the cybersecurity academy.
  - Explore the cybersecurity services from Schneider Electric.

Product Related Cybersecurity Information

Refer to the Cybersecurity Guidelines for EcoStruxure Machine Expert, Modicon and PacDrive Controllers and Associated Equipment User Guide.

NOTE: Schneider Electric adheres to industry best practices in the development and implementation of control systems. This includes a "Defense-in-Depth" approach to secure an Industrial Control System. This approach places the controllers behind one or more firewalls to restrict access to authorized personnel and protocols only.

⚠WARNING

UNAUTHENTICATED ACCESS AND SUBSEQUENT UNAUTHORIZED MACHINE OPERATION

- Evaluate whether your application environments are connected to your critical infrastructure and, if so, take appropriate steps in terms of prevention, based on Defense-in-Depth, before connecting the automation system to any network.
- Limit the number of devices connected to a network to the minimum necessary.
- Isolate your industrial network from other networks inside your company.
- Protect any network against unintended access by using firewalls, VPN, or other, proven security measures such as an Intrusion Prevention System or Intrusion Detection System.
- Monitor activities within your systems.
- Prevent subject devices from direct access or direct link by unauthorized parties or unauthenticated actions.
- Install certificates that are issued by publicly known Trusted Certificate Authorities.
- Keep your systems up to date and rely only on legitimate sources.
- Prepare a recovery plan including backup of your system and process information.

Failure to follow these instructions can result in death, serious injury, or equipment damage.

For more information on organizational measures and rules covering access to infrastructures, refer to ISO/IEC 27000 series, Common Criteria for Information Technology Security Evaluation, ISO/IEC 15408, IEC 62351, ISA/IEC 62443, NIST Cybersecurity Framework, Information Security Forum - Standard of Good Practice for Information Security and refer to Cybersecurity Guidelines for EcoStruxure Machine Expert, Modicon and PacDrive Controllers and Associated Equipment.

Related Documents

Document title	Reference
Cybersecurity Best Practices	CS-Best-Practices-2019-340
Cybersecurity Guidelines for EcoStruxure Machine Expert, Modicon and PacDrive Controllers and Associated Equipment User Guide	EIO0000004242

To find documents online, visit the Schneider Electric download center (www.se.com/ww/en/download/).

Information on Non-Inclusive or Insensitive Terminology

As a responsible, inclusive company, Schneider Electric is constantly updating its communications and products that contain non-inclusive or insensitive terminology. However, despite these efforts, our content may still contain terms that are deemed inappropriate by some customers.

Terminology Derived from Standards

The technical terms, terminology, symbols and the corresponding descriptions in the information contained herein, or that appear in or on the products themselves, are generally derived from the terms or definitions of international standards.

In the area of functional safety systems, drives and general automation, this may include, but is not limited to, terms such as *safety*, *safety function*, *safe state*, *fault*, *fault reset*, *malfunction*, *failure*, *error*, *error message*, *dangerous*, etc.

Among others, these standards include:

Standard	Description
IEC 61131-2:2007	Programmable controllers, part 2: Equipment requirements and tests.
ISO 13849-1:2023	Safety of machinery: Safety related parts of control systems. General principles for design.
EN 61496-1:2020	Safety of machinery: Electro-sensitive protective equipment. Part 1: General requirements and tests.
ISO 12100:2010	Safety of machinery - General principles for design - Risk assessment and risk reduction
EN 60204-1:2006	Safety of machinery - Electrical equipment of machines - Part 1: General requirements
ISO 14119:2013	Safety of machinery - Interlocking devices associated with guards - Principles for design and selection
ISO 13850:2015	Safety of machinery - Emergency stop - Principles for design
IEC 62061:2021	Safety of machinery - Functional safety of safety-related electrical, electronic, and electronic programmable control systems
IEC 61508-1:2010	Functional safety of electrical/electronic/programmable electronic safety-related systems: General requirements.
IEC 61508-2:2010	Functional safety of electrical/electronic/programmable electronic safety-related systems: Requirements for electrical/electronic/programmable electronic safety-related systems.
IEC 61508-3:2010	Functional safety of electrical/electronic/programmable electronic safety-related systems: Software requirements.
IEC 61784-3:2021	Industrial communication networks - Profiles - Part 3: Functional safety fieldbuses - General rules and profile definitions.
2006/42/EC	Machinery Directive
2014/30/EU	Electromagnetic Compatibility Directive
2014/35/EU	Low Voltage Directive

In addition, terms used in the present document may tangentially be used as they are derived from other standards such as:

Standard	Description
IEC 60034 series	Rotating electrical machines
IEC 61800 series	Adjustable speed electrical power drive systems
IEC 61158 series	Digital data communications for measurement and control – Fieldbus for use in industrial control systems

Finally, the term *zone of operation* may be used in conjunction with the description of specific hazards, and is defined as it is for a *hazard zone* or *danger zone* in the *Machinery Directive* (*2006/42/EC*) and *ISO 12100:2010*.

NOTE: The aforementioned standards may or may not apply to the specific products cited in the present documentation. For more information concerning the individual standards applicable to the products described herein, see the characteristics tables for those product references.

General Information

What’s in This Part

Presentation of the Library

What's in This Chapter

General Information

Library Overview

The DataLogging library performs data logging management of controllers that support file management operations. A data log file is created that is a text file of user-defined strings containing application data information for a process or machine. The data log file is stored on the controller. You can upload the file and open it with a text editor. The information includes embedded variable values and their associated text. Time and date stamping are additional options.

Characteristics of the Library

The table indicates the characteristics of the library:

Characteristic	**Value**
Library title	DataLogging
Company	Schneider Electric
Category	**Controller**
Component	**Core Repository**
Default namespace	**SEDL**
Forward compatible library	Yes (FCL)

Controller Platforms

To identify the controllers that are compatible with the library, refer to the List of Controllers Compatible with your Libraries in the Libraries Overview online help.

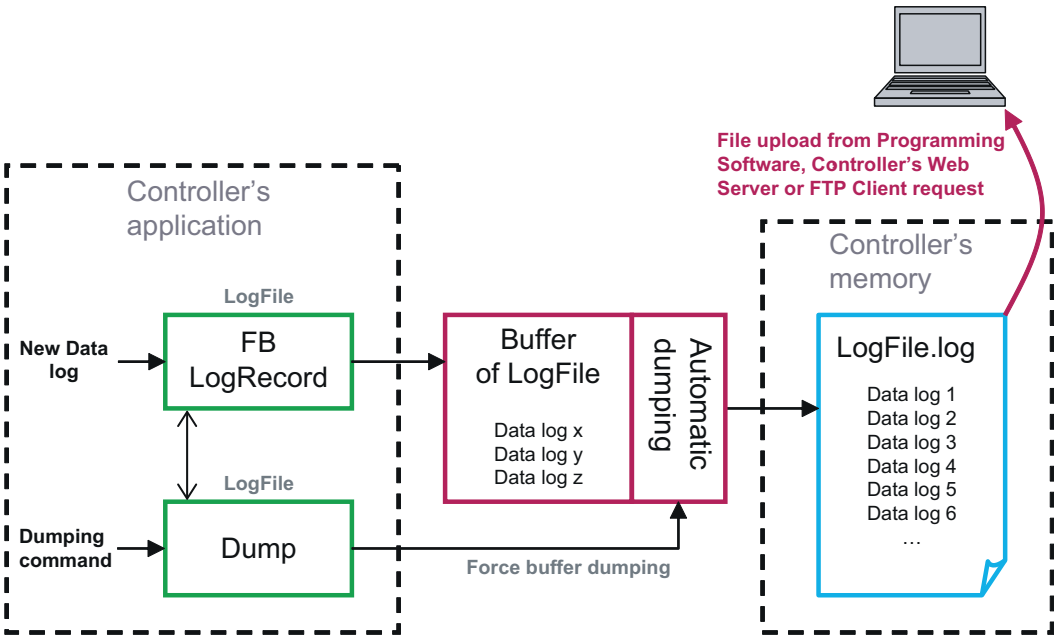
General Considerations

Since file operations require longer time to complete, execute the function blocks in a low-priority, cyclic task. Adapt the watchdog function for this task to allow sufficient time for the connection.

Introduction to Data Logging

Overview

You can monitor and analyze application data by examining the data log file (.log).



The figure illustrates an application that includes the *LogRecord* function block and the *Dump* method. The *LogRecord* function block writes data to a buffer, which then empties into the data log file (.log) located in the controller memory. Buffer dumping occurs automatically when the buffer is 80% full. Alternatively, you can manually force this process by executing the *Dump* method. As an FTP client, a PC can access this data log file when the controller functions as an FTP server. You can also upload the file using EcoStruxure Machine Expert, or the web server of the controller.

NOTE: Data logging is exclusively supported by controllers with file management functionality. Consult the *Programming Guide* specific to your controller to see if it supports file management. The software itself does not evaluate your controller for compatibility with data logging activities.

Sample Data Log File (.log)

```
Entries in File: 8; Last Entry: 8;
18/06/2009;14:12:33;cycle: 1182;
18/06/2009;14:12:35;cycle: 1292;
18/06/2009;14:12:38;cycle: 1450;
18/06/2009;14:12:40;cycle: 1514;
18/06/2009;14:12:41;cycle: 1585;
18/06/2009;14:12:43;cycle: 1656;
18/06/2009;14:14:20;cycle: 6346;
18/06/2009;14:14:26;cycle: 6636;
```

Implementation Procedure

First declare and configure the data log files in your application before starting to write your program.

Creating and Configuring the Data Log

Adding a Data Log Manager

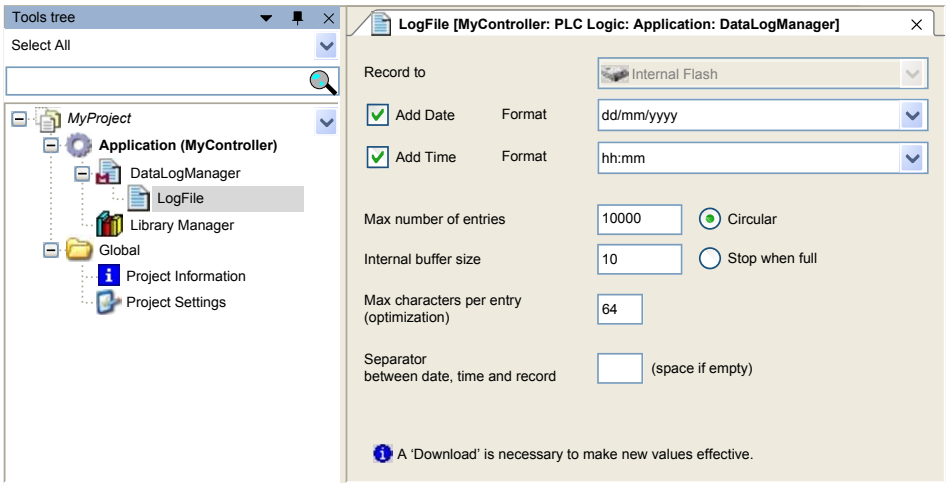
Add a data log manager to your application before configuring data logging:

Step	Action
1	In the **Tools tree**, select the **Application** node, click the green plus sign and select **Add other objects > DataLogManager...**. **Result:** The **Add Data Log Manager** dialog box is displayed.
2	In the **Add Data Log Manager** dialog box, click **Add**. **Result:** A **DataLogManager** node is displayed below the **Application** node.
3	Select the **DataLogManager** node, click the green plus sign, and select **DataLog...**. **Result:** The **Add DataLog** dialog box is displayed.
4	In the **Data Logging File Name** text box, enter a name for your data log file, and click **Add**. **Result:** The data log file with the given name is displayed below the **DataLogManager** node, and the configuration editor opens in the editor view in the middle of the Logic Builder screen. **Note:** The data log file name cannot be modified.
5	Set the data log file parameters, page 19.
6	Repeat steps 3 to 5 to create additional data log files.

NOTE: For each data log file added to the **DataLogManager**, an instance of the function block *LogRecord* is created. A maximum number of 20 instances of the *DataLog* function block can be managed in an application.

Configuration Editor

The configuration editor is displayed after you have added a data log file to your configuration:



Configuration Parameters of the Data Log File

Parameter	Description	
Add Date **Add Time**	These options print the respective date or time for each record. For example, an instance associated with 10 June 2009 at 2:30 p.m. could be represented as `10/06/2009` or `06/10/2009` or `20090610` ... at `14:30` or `02:30:00 pm`, etc.	
Max number of entries	This option sets the maximum number of records contained in the data log file. Value range: 10...65536 Default value: 10000	
Mode	**Circular** (default)	When the **Max number of entries** is reached, new records overwrite old records. The first line of the data log file indicates the present position of the last record and the rank of other records. The following cases are based on a file with a 10-record maximum: • **Case 1:** `Entries in File: 8; Last Entry: 8` **Meaning:** The number of entries in the file equals the value of `Last Entry`. The records are ranked from N°1 (the oldest) to N°8 (the most recent). • **Case 2:** `Entries in File: 10; Last Entry: 5` **Meaning:** The number of entries in the file is greater than the value of `Last Entry`. The file is full, and a new record replaces the oldest one. The 10 records are ranked in this order (oldest to newest): 6, 7, 8, 9, 10, 1, 2, 3, 4, 5
	Stop when full	When the **Max number of entries** is reached, a new record attempt returns an error message.
Internal buffer size	Configure the size of the RAM buffer that stores the added records. Value range: 1...value configured with **Max number of entries** Default value: 10	
Max characters per entry (optimization)	Set the maximum length of each entry. Value range: 10...255 Default value: 64 The maximum length includes separators and optional date and time. Additional spaces are added at the end of the line to accommodate the number of defined characters.	
Separator between date, time and record	Define the character to be inserted between different fields in the data log. An additional separator is added after the record.	

NOTE: For each configured data log file, there is a *LogRecord* function block instance with the same name as this data log file is created. It handles the internal data and allows data logging management.

Diagnostic Concept

What's in This Chapter

Diagnostic Concept

Overview

EcoStruxure Machine Expert provides a three-layer diagnostic concept for the libraries. This concept is applicable to the Technology/Module libraries (for example, the library CommonToolbox) and uses enumerations for diagnostic coding.

The diagnostic information has the following layers:

1. General information on the exception: Provides basic details about the exception without requiring specific knowledge of the POU functionality.
2. POU-specific diagnostic and status messages (part 1): Offers information on the source triggering the diagnostic or status messages.
3. POU-specific diagnostic and status messages (part 2): Provides detailed and dynamic information on the source triggering the diagnostic or status messages.

   This information changes at runtime (for example, information about the condition of the input parameters). This diagnostic output is optional for the POUs.

The diagnostic concept offers the following advantages:

- Diagnostic messages can be viewed online
- Detailed information on diagnostic events, including diagnostic codes
- Overview of the status or exceptional condition of a POU
- Pertinent suggested solutions to address the causes for exceptional conditions
- Enumerated diagnostic messages to facilitate multi-language support for HMI displays

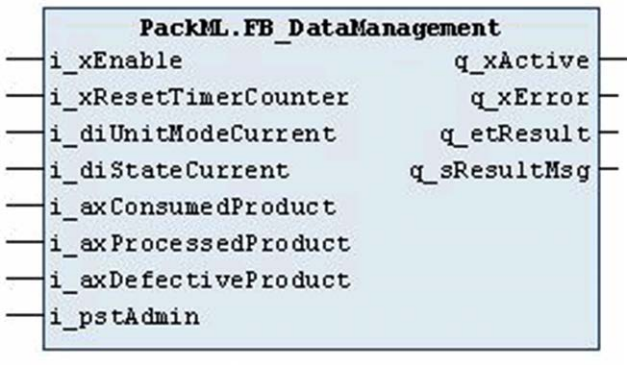
Diagnostic Outputs

Function blocks and functions or methods can have the three diagnostic outputs *q_xError*, *q_etResult*, and *q_sResultMsg*. The outputs are defined in the POU one after another.

Output	Data type	Meaning
q_xError	*BOOL*	The output *q_xError* is set to TRUE when an error has been detected during execution of the function block. The outputs *q_etResult* and *q_sResultMsg* provide the corresponding error code and a plain text information.
q_etResult	*ET_Result*(1)	Provides diagnostic and status information as a numeric value. *If q_xError* = FALSE, *q_etResult* provides status information. *If q_xError* = TRUE, *q_etResult* provides diagnostic/error information. The enumeration *ET_Result* contains the possible values of the POU operation results.

Output	Data type	Meaning
q_sResultMsg	STRING	Provides additional information about the present status of the POU. If the POU is in error state (*q_xError* = TRUE), the message provides additional information for the error cause and, if meaningful, a possible solution. If the POU is busy the present process or state is issued at this output.
(1) Each library provides its own, specific enumeration *ET_Result* which contains the possible *q_etResult* outputs for that library.		

Diagnostic information example:



Common Inputs and Outputs

What's in This Chapter

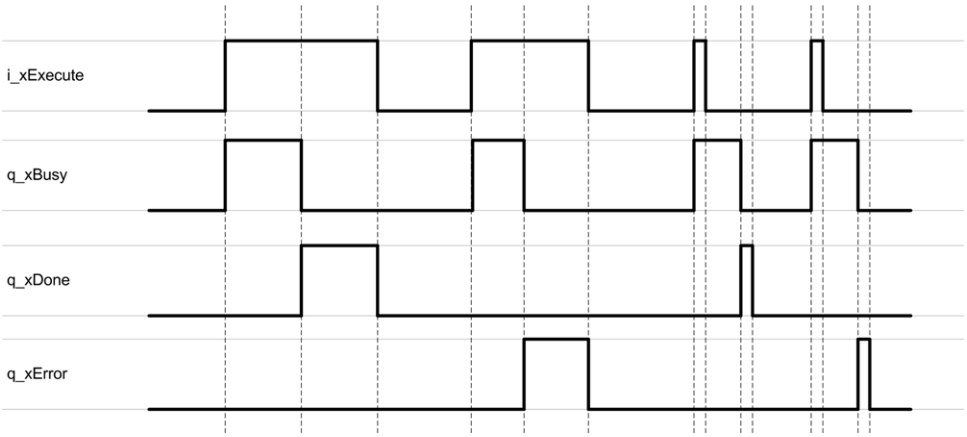
Behavior of Function Blocks with the Input *i_xExecute*

General Information

A rising edge of the input *i_xExecute* starts the execution of the function block. The function block continues execution and the output *q_xBusy* is set to TRUE. Additional rising edges at the input *i_xExecute* are ignored while the function block is being executed.

Once the execution is finished, the outputs *q_xDone* or *q_xError* remain TRUE until the input *i_xExecute* is set to FALSE. If the input is reset before the execution is finished, the function block continues to execute until it has finished its processing, and then the outputs *q_xDone* or *q_xError* are set to TRUE for one cycle.

Example



Program Organization Units (POU)

What's in This Part

LogRecord Function Block

What's in This Chapter

General Information

Overview

Type:	Function block
Available as of:	V1.0.0.0

Task

This function block is used for logging a UNICODE string in a specific data log file.

Description

The *LogRecord* function block writes text string entries to the data log file. It stores the input string in an internal buffer. When 80% of the capacity of the buffer is full, the data is moved to the controller. To start the saving process programmatically, execute the *Dump* method.

When power is removed, you may lose the data in the internal buffer or increase the cycle time until the buffer is emptied.

NOTICE

LOSS OF DATA

- Do not remove power from the controller until all internal buffer information has been moved to the actual file system.
- If the data being recorded is important to your application, configure the Internal buffer size parameter to 1.

Failure to follow these instructions can result in equipment damage.

If the buffered data is important to your application, and you choose to have an internal buffer size greater than 1, implement an external means, such as a controller input, to signal the execution of the *Dump* method.

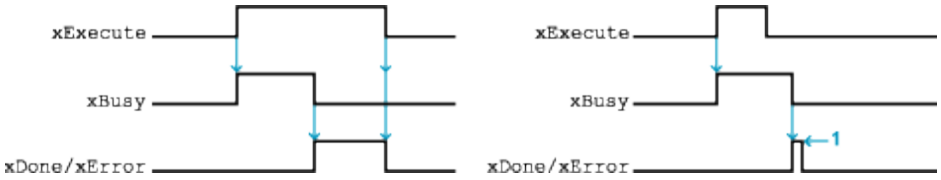
Interface

Input	Data type	Description
xExecute	BOOL	The function is executed on the rising edge of this input. **NOTE:** When *xExecute* is set to TRUE at the first task cycle in RUN after a cold or warm reset, the rising edge is not detected.
wsRecord	WSTRING	This user-specified UNICODE text string is written to the data log file. **NOTE:** The WSTRING, page 27 type is available in the Standard64.lib library that is automatically inserted when the data log manager is added to the application.

Output	Data type	Description
xDone	BOOL	This output is set to TRUE when the record is successfully saved to the internal buffer without error messages.
xBusy	BOOL	This output remains TRUE while *LogRecord* is busy (until the transfer to the buffer is complete).
xError	BOOL	This output is set to TRUE when an error is detected (for example, when the internal buffer is full).
eError	ERROR	This output contains the error code when *xError* is TRUE: • NO_ERROR • INIT_ERROR • DUMP_ERROR • BUFFER_FULL_ERROR • FILE_FULL_ERROR • DUMP_INCOMPLETED • INPUT_ERROR • FILE_OPEN_ERROR • FILE_SETPOINTER_ERROR • FILE_WRITE_ERROR • FILE_CLOSE_ERROR

NOTE: If the record exceeds the configured length, it is truncated.

The *xDone* and *xError* outputs remain TRUE as long as *xExecute* is TRUE. When *xExecute* is set to FALSE before *xDone* or *xError* is set to TRUE (*xBusy* is still TRUE), one of each is set to TRUE when the function block is completed during one controller cycle so that the application detects this end:



1 One cycle as `xExecute` is FALSE.

For each data log file, an instance of the *LogRecord* function block is created.

NOTE: Do not explicitly declare an instance of the function block because the instance is declared automatically. If you explicitly declare an instance of the function block, the *2 time variable declaration* message is displayed () and the function block is rendered inoperable.

Add the function block to your POU and specify the appropriate data log file instance with **Input Assistant** (refer to Adding the *LogRecord* Function Block, page 26).

Properties of the Data Log

You can access the data log properties after you have configured the function block.

The *LogRecord* properties are additional variables (read only) automatically attached to the *LogRecord* instance that provide information about the data log file status:

Variables	Data type	Description
<Data Log File name>.NumberOfRecords	UDINT	Number of records in the data log file
<Data Log File name>. NumberOfBufferedRecords	UINT	Number of records in the buffer
<Data Log File name>.FileStatus	FILE_STATUS	Status information about the data log file (FILE_STATUS type): • 0: OK • 1: FILE_FULL • 2: NO_WRITE_ACCESS • 3: FILE_NOT_EXISTS
<Data Log File name>.DumpInProgress	BOOL	TRUE when the buffered records are being saved in the data log file.

Adding the *LogRecord* Function Block

Add the *LogRecord* function block to your project:

Step	Action
1	To select a *LogRecord* function block in your POU, use the **Input Assistant** or directly type `LogRecord`: In the **Input Assistant** dialog box, select the following options: • **Categories**: **Function Blocks** • **Items**: **{} SEDL > LogRecord** (With the option **Structured view** selected.)
2	Click **OK** or press **ENTER**. **Result**: The *LogRecord* function block is saved to your project.
3	Select the appropriate data log file as *LogRecord* instance name from the **Input Assistant** or directly enter the data log file name.
4	Configure the inputs and outputs of the function block. Refer to the description of parameters, page 24.

NOTE: For the **DataLogManager**, a maximum number of 20 files is allowed in the *usr/log* directory. If the number of files is exceeded, the 21st instance of the *LogRecord()* function block is not executed and the error message INIT_ ERROR is displayed.

Using *LogRecord* in Task with Short Interval

The *LogRecord* function block needs more than 15 interval cycles after the activation (with *xExecute*) to save a record to the logfile. Therefore, it is a good practice to use this function block within a task with a short interval:

Task Type	Interval (ms)	Minimum Time Needed to Save Record
cyclic	20	300 ms
cyclic	1	15 ms - preferred solution
event	-	15 events - inefficient solution

Building a Wide String (WSTRING)

The *wsRecord* input of the *LogRecord* function block is of the type WSTRING (wide string). To build the log string, add the Standard64 library to your application and use wide string functions.

This figure indicates the creation of a sample WSTRING that includes a variable value:

```
PROGRAM POU_1
VAR
    cycle: INT;
    str: WSTRING;
END_VAR
```

```
cycle := cycle + 1;
str := wconcat("cycle: ", INT_TO_WSTRING(cycle));
```

Dump (Method)

Overview

Type:	Method
Available as of:	V1.0.0.0

Task

Activates the save mechanism of the *LogRecord* function block.

Description

You can use the *Dump* method to move stored records in the internal buffer to the file in the file system of the controller.

Interface

Input	Data type	Description
xExecute	BOOL	The save function is executed on the rising edge of this input. **NOTE:** When *xExecute* is set to 1 at first controller task cycle, the rising edge is not detected.

Output	Data type	Description
xDone	BOOL	This output is set to TRUE when the records are successfully saved without error messages.
xBusy	BOOL	This output remains TRUE while *Dump* is busy (until the file is completely written).
xError	BOOL	This output is set to TRUE when an error is detected (for example, when the data log file is full).
eError	ERROR	This output contains the error code when *xError* is TRUE: • NO_ERROR • INIT_ERROR • DUMP_ERROR • BUFFER_FULL_ERROR • FILE_FULL_ERROR • DUMP_INCOMPLETED • INPUT_ERROR • FILE_OPEN_ERROR • FILE_SETPOINTER_ERROR • FILE_WRITE_ERROR • FILE_CLOSE_ERROR

For the *Dump* method implementation, refer to Adding the *Dump* Method, page 28.

Adding the *Dump* Method

Add the *Dump* method to your project:

Step	Action
1	To add a *Dump* method to your POU, use the **Input Assistant** or directly type `LogRecord.Dump`: In the **Input Assistant** dialog box, select the following options: • **Categories**: **Function Blocks** • **Items**: **SEDL > LogRecord > Dump** (With the option **Structured view** selected.)
2	Click **OK** or press **ENTER**. **Result**: The *LogRecord.Dump* method is saved to your project.
3	Select the appropriate data log file as *LogRecord* instance name from the **Input Assistant** or directly enter the data log file name.
4	Configure the inputs and outputs as you would for other functions. Refer to Interface, page 28.

Automatic Save to Data Log File

When the *LogRecord* function block is used, the system automatically moves the data from the internal buffer to the data log file when the buffer is 80% full. The *Dump* method allows you to activate this move before reaching the 80% threshold. This 80% threshold allows the write process to begin before the buffer is full and while a new record is added.

The table indicates the number of records stored in the file for a given configured buffer size (80% of buffer size rounded down to the nearest integer). Refer to the **Internal buffer size** parameter, page 19.

Value of the Internal buffer size parameter	80% Threshold	Comment
1	1	The save to the data log file is activated as soon as a record is added, with an additional slot available for a new record that arrives during this save.
2	1	The save to the data log file is activated when the buffer reaches 80%, with an additional slot available for a new record that arrives during this save.
3	2	
4	3	
5	4	The save to the data log file is activated when the buffer reaches 80%, with additional slots available for new records that arrive during this save.
6	4	
7	5	
8	6	
...		

In the case of either an explicit (*Dump*) or automatic (80%) save, the data log file closes after each record (or record group) to help prevent data loss in case of a subsequent external power outage.

When power is removed, you may lose the data in the internal buffer or increase the cycle time until the buffer is emptied.

NOTICE

LOSS OF DATA

- Do not remove power from the controller until all internal buffer information has been moved to the actual file system.
- If the data being recorded is important to your application, configure the Internal buffer size parameter to 1.

Failure to follow these instructions can result in equipment damage.

If the buffered data is important to your application, and you choose to have an internal buffer size greater than 1, implement an external means, such as a controller input, to signal the execution of the *Dump* method.

Index

Schneider Electric
35 rue Joseph Monier
92500 Rueil Malmaison
France

+ 33 (0) 1 41 29 70 00

www.se.com

EIO0000002938.02