



Programming manual

CP9031

ecomatDisplay SDK Utilities

GB

Contents

1	Preliminary note	5
1.1	Legal and copyright information	5
1.2	Purpose of the document	5
1.3	Symbols used	5
1.4	Warnings used	6
1.5	Change history	6
1.5.1	Changes 08 / 2022 – 11428978 / 00	6
1.5.2	Changes 10 / 2017 – 7391179 / 00	6
2	Safety instructions	7
2.1	Required background knowledge	7
3	Intended use	8
4	Function	9
4.1	Basic and R360 system	9
4.2	ecomatController	9
4.3	Overview	9
4.4	User interface	10
4.5	Communication	10
4.5.1	Communication of basic and R360 system	10
4.5.2	ecomatController communication	11
5	Installation	12
5.1	Requirements	12
5.1.1	Hardware	12
5.1.2	Software	12
5.1.3	Licensing	12
5.2	ifm package	12
5.2.1	Contents of the ifm package	12
5.2.2	CODESYS package	13
5.2.2.1	Installing the CODESYS package	13
5.2.2.2	Updating the CODESYS package	13
5.2.2.3	Uninstalling the CODESYS package	13
5.2.3	rpm package for the ecomatDisplay	14
5.2.3.1	Memory allocation	14
5.2.3.2	Preparing ecomatDisplay	14
5.2.3.3	Installing the rpm package on the ecomatDisplay	14
5.2.3.4	Transferring the software to other ecomatDisplays	15
5.3	Adding a function library to the project	16
5.4	Loading an example application to the ecomatDisplay	16
5.5	Preparing the ecomatController	16
6	Programming	17
6.1	Using library functions	17
7	ifm function libraries	19
7.1	ifmSDKutil_ecomatDisplay library	19
7.1.1	Behaviour model of the ifm function blocks	19
7.1.1.1	General	20
7.1.1.2	Behaviour model EXECUTE	20
7.1.2	Function blocks	21
7.1.2.1	PingDevice	21
7.1.2.2	GetSysInfo	23
7.1.2.3	Command	25
7.1.2.4	ReadFromController	27
7.1.2.5	GetSegmentListFromFile	29
7.1.2.6	SelectSegmentFromList	31
7.1.2.7	WriteToController	33
7.1.2.8	WriteInterfaceSettings	36
7.1.2.9	Authentication	39
7.1.2.10	GetSdkInterfaces	42
7.1.2.11	GetSysInfo_ecomatController	44

7.1.2.12	Command_ecomatController	46
7.1.2.13	ReadPackageFrom_ecomatController	49
7.1.2.14	UpdateFirmware_ecomatController	52
7.1.2.15	WritePackageTo_ecomatController	55
7.1.2.16	ReadIOSettings_ecomatController	58
7.1.2.17	WriteIOSettings_ecomatController	60
7.1.2.18	ReadInterfaceSettings_ecomatController	62
7.1.2.19	WriteInterfaceSettings_ecomatController	64
7.1.2.20	GetDataFileList_ecomatController	66
7.1.2.21	ReadDataFiles_ecomatController	68
7.1.2.22	WriteDataFiles_ecomatController	71
7.1.2.23	ReadDeviceLogs_ecomatController	74
7.1.2.24	ReadSegmentFrom_ecomatController	76
7.1.2.25	WriteSegmentTo_ecomatController	79
7.1.2.26	SetPasswords_ecomatController	83
7.1.3	ENUM	86
7.1.3.1	eCMD_LIST (ENUM)	86
7.1.3.2	eDEVICES (ENUM)	86
7.1.3.3	eSEGM_TYPE (ENUM)	87
7.1.3.4	eCMD_LIST_CR7nnS (ENUM)	87
7.1.3.5	ePACKAGE_TYPE_CR7nnS (ENUM)	87
7.1.3.6	eUSER_LEVEL (ENUM)	88
7.1.4	STRUCT	89
7.1.4.1	stSYS_INFO	89
7.1.4.2	stSEGM_LIST	89
7.1.4.3	stSEGM_NAME	90
7.1.4.4	stSEGM_TYPE_NAME	90
7.1.4.5	stSEGM_LIST_WRITE	90
7.1.4.6	stLAST_EVNT_RESP	91
7.1.4.7	stRD_WR_INFO_TYPE	91
7.1.4.8	stSEGM_READ_STATUS	91
7.1.4.9	stSEGM_WRITE_STATUS	91
7.1.4.10	stITF_SETTINGS	92
7.1.4.11	stSymbolCfgInfo	92
7.1.4.12	stSOFTWARE_INFO_CR7nnS	92
7.1.4.13	stDEVICE_INFO_CR7nnS	93
7.1.4.14	stNETWORK_INFO_CR7nnS	93
7.1.4.15	stSYS_INFO_CR7nnS	93
7.1.4.16	stREAD_INFO_TYPE_CR7nnS	93
7.1.4.17	stIO_SETTINGS_CR7nnS	94
7.1.4.18	stIO_INFO_CR7nnS	94
7.1.4.19	stIO_DATA_CR7nnS	94
7.1.4.20	stITF_SETTINGS_CR7nnS	94
7.1.4.21	stITF_SETTINGS_CAN_CR7nnS	95
7.1.4.22	stITF_SETTINGS_ETH_CR7nnS	95
7.1.4.23	stITF_SETTINGS_RS232_CR7nnS	95
7.1.4.24	stDATA_FILE_LIST_CR7nnS	96
7.1.4.25	stDEVICE_LOGS_CR7nnS	96
7.1.4.26	stDEVICE_LOG_INFO_CR7nnS	96
7.1.4.27	stSEGM_LIST_READ_CR7nnS	96
7.1.4.28	stSEGM_LIST_WRITE_CR7nnS	97
7.1.4.29	stPACKAGE_LIST_CR7nnS	97
7.1.4.30	stPACKAGE_READ_STATUS_CR7nnS	97
7.1.4.31	stPASSWORD_CR7nnS	98
7.1.4.32	stINTF	98
7.1.5	GVL	99
7.1.5.1	gv_EvntCallBack	99
7.1.6	GCL	100
7.1.6.1	gc_AllSegments	100
7.1.7	DUT	102
7.1.7.1	aSEGM_TYPE_NAME	102
7.1.7.2	aINTF_LIST	102

7.1.7.3	aIO_DATA_CR7nnS	102
7.1.7.4	aITF_SETTINGS_CAN_CR7nnS	102
7.1.7.5	aITF_SETTINGS_ETH_CR7nnS	102
7.1.7.6	aITF_SETTINGS_RS232_CR7nnS	102
7.1.7.7	aDEVICE_LOG_INFO_CR7nnS	102
7.1.7.8	aNETWORK_INFO_CR7nnS	103
7.1.7.9	aSOFTWARE_INFO_CR7nnS	103

1 Preliminary note

You will find instructions, technical data, approvals and further information using the QR code on the unit / packaging or at www.ifm.com.

1.1 Legal and copyright information

© All rights reserved by ifm electronic gmbh. No part of these instructions may be reproduced and used without the consent of ifm electronic gmbh.

All product names, pictures, companies or other brands used are the property of the respective rights owners.

- AS-i is the property of AS-International Association
- CAN is the property of Robert Bosch GmbH, Germany
- CAN is the property of CiA (CAN in Automation e.V.), Germany
- CODESYS™ is the property of CODESYS GmbH, Germany
- DeviceNet™ is the property of ODVA™ (Open DeviceNet Vendor Association), USA
- EtherNet/IP® is the property of ODVA™ (Open DeviceNet Vendor Association), USA
- EtherCAT® is a registered trademark and patented technology, licensed by Beckhoff Automation GmbH, Germany.
- ISOBUS is the property of AEF - Agricultural Industry Electronics Foundation e.V., Germany
- Modbus® is the property of Schneider Electric SE, France
- PROFINET® is the property of PROFIBUS Nutzerorganisation e.V., Germany
- PROFINET® is the property of PROFIBUS Nutzerorganisation e.V., Germany
- Microsoft® is the property of Microsoft Corporation, USA
- Windows® is the property of Microsoft Corporation, USA
- Linux® is the registered trademark of Linus Torvalds in the USA and in other countries.

1.2 Purpose of the document

The document describes the use of the ifm SDK ecomatDisplay for the maintenance of connected ifm controllers.

1.3 Symbols used

- ✓ Requirement
- Instructions
- ▷ Reaction, result
- [...] Designation of keys, buttons or indications
- Cross-reference
-  Important note
Non-compliance may result in malfunction or interference.
-  Information
Supplementary note

1.4 Warnings used

	ATTENTION Warning of damage to property
	CAUTION Warning of personal injury ▷ Slight reversible injuries may result.
	WARNING Warning of serious personal injury ▷ Death or serious irreversible injuries may result.

1.5 Change history

1.5.1 Changes 08 / 2022 – 11428978 / 00

- Update to SDK Utilities Package V3

1.5.2 Changes 10 / 2017 – 7391179 / 00

- New creation of the document

2 Safety instructions

- The described function blocks of the function library and the IEC applications created from them must not be used industrially without testing. ifm does not assume any guarantee for the functionality, in particular on machines and systems. It is not a (software) product covered by European directives.
Conformity with relevant directives and standards to meet the legal requirements regarding the placing on the market will only be determined during the verification and validation in the context of development at the system manufacturer. The system manufacturer shall not make the system available on the market for use by end users until that time.
- Read this document before setting up the product and keep it during the entire service life.
- The product must be suitable for the corresponding applications without any restrictions.
- Only use the product for its intended purpose (→ Intended use).
- If the operating instructions or the technical data are not adhered to, personal injury and/or damage to property may occur.
- The manufacturer assumes no liability or warranty for any consequences caused by tampering with the product or incorrect use by the operator.
- Improper or non-intended use may lead to malfunctions, undesirable effects in the application or the loss of warranty claims.
- Observe applicable documents.

2.1 Required background knowledge

This document is for specialists. Specialists are people who are qualified by their appropriate training and their experience to see risks and to avoid possible hazards that may be caused during operation or maintenance of the device.

For programming, they should also have knowledge of control technology experience in PLC programming to IEC 61131-3.

The document gives information concerning the correct handling of the product.

3 Intended use

With the help of the functions of the library "ifmSDKutil_ecomatDisplay", the user can access controls of the following ifm device families (= target devices) from the CODESYS application of an ecomatDisplay (= client) for maintenance purposes:

- ClassicController
- ExtendedController
- SmartController
- SafetyController
- BasicController
- ecomatController

The library offers the following functions:

- Reading device information
- Writing applications, firmware and data to the device (download)
- Reading applications, firmware and data from the device (upload)
- Controlling the device (start, stop, reset, reboot)
- Testing the connection to a device
- Help functions

4 Function

4.1 Basic and R360 system

The library provides functions for the following applications for one or more controllers of the Basic and R360 system:

- Retrieving system information.
- Controlling the operating mode.
- Reading and writing application, firmware, user data, etc.
- Updating a controller with new firmware and/or application etc. with the following file types:
 - resx files for all Basic and R360 controllers
 - h86 files for all R360 controllers
 - binary files for all Basic and R360 controllers
- Cloning of the complete software content of a controller in the field, e.g. in case of service.
- All or certain memory segments can be read from a programmed controller and written to another controller or to the same controller, e.g. in case of a failed update.

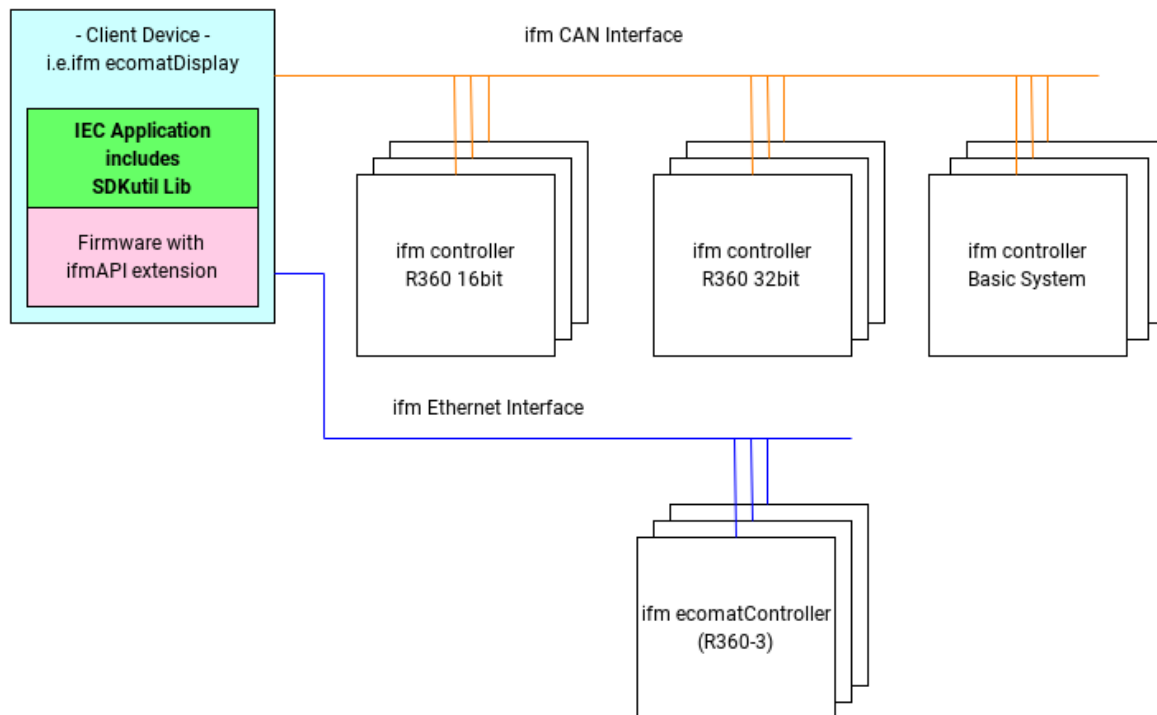
4.2 ecomatController

The library provides functions for the following use cases for one or more ecomatControllers:

- User authentication for access to the controller.
- Retrieving system information.
- Controlling the operating mode.
- Reading and writing the IEC application (standard/safety), the firmware and individual software segments.
- Reading and writing the memory configuration
- Reading and writing the I/O configuration.
- Reading and writing the communication interface settings of the controller.
- Reading user passwords for all permissible user levels.
- Reading and writing device event logs.
- Reading and writing user data files.
- Updating a controller with new firmware and/or application, etc., available as a zip file.
- Cloning of the complete software content of a controller in the field, e.g. in case of service.
- All or certain memory segments can be read from a programmed controller and written to another controller or to the same controller, e.g. in case of a failed update.

4.3 Overview

The overview shows the system structure and the communication paths between the components.



4.4 User interface

- The interface is provided as a CODESYS library with function blocks (FBs) and data structures.
- The function blocks of the library are divided into 2 groups.
This happens because of the differences in the available communication interfaces, data structures and functions of the various control types:
 - FB group 1: FBs for communication with R360 controllers and Basic System - The controller types Basic System, R360Classic Line, Extended Line, 16 bits, 32 bits, Safety and Standard controllers are supported by this group of FBs.
 - FB group 2: FBs for communication with ecomatController - All ecomatControllers (standard and safety) are supported by this group of FBs. FBs of this group are recognisable by the suffix "_ecomatController" in the FB name.

4.5 Communication

4.5.1 Communication of basic and R360 system

- The ecomatDisplay and R360 devices / basic devices communicate via CAN bus.
- All available CAN channels of the ecomatDisplay can be used. The number of available CAN channels is device-specific and can be selected from the associated `ifmDevice_ecomatDisplay-Bibliothek > ENUM - ifmDevice.CAN_CHANNEL`.
- The CAN baud rate used is the baud rate that has been configured within the application with blocks of the `ifmRawCAN` library or the baud rate configured in the device tree if the CANopen manager is used under the device communication node.



► For the supported baud rates through the APIs in the underlying SDK DLLs, refer to the documentation of ifm article no. CP9030.

- It is only possible to communicate with one controller at a time.
The communication commands to several controllers are called up one after the other in the programmed sequence in the CODESYS application. This ensures, for example, that updates of several controllers will be carried out automatically one after the other.

4.5.2 ecomatController communication

- Communication between ecomatDisplay and ecomatController takes place via the Ethernet interface (TFTP/UDP).
- All available Ethernet interfaces of the ecomatDisplay can be used. The number of available Ethernet interfaces is device-specific and is selected from the associated `ifmDevice_ecomatDisplay-Bibliothek > ENUM - ifmDevice.eETHERNET_INTERFACES`.
- It is only possible to communicate with one controller at a time.
The communication commands to several controllers are called up one after the other in the programmed sequence in the CODESYS application. This ensures, for example, that updates of several controllers will be carried out automatically one after the other.

5 Installation

5.1 Requirements

5.1.1 Hardware

- A device of the ecomatDisplay product family as client (= the device from which the maintenance is carried out)
 - Runtime system: 2.1.0.15 or higher
- At least one device of the ecomatController, Basic Controller or R360 Controller product family as the target device (= the device that is being serviced)
- PC for the CODESYS programming system for the initial setup and project planning of the ecomatDisplay and the controls.
- The ecomatDisplay, the ecomatController(s) and the PC are connected via Ethernet and are located in the same subnet.
- The ecomat display and any Basic or R360 controllers are connected via the CAN interface.

5.1.2 Software

- CODESYS Development System
 - Version: 3.5 SP16 Patch 4 or higher
- Package "CODESYS for ifm ecomatDisplay"
 - Version: 2.3.0.0 or higher
- SSH client, e.g. Putty or WinSCP



The features and functions warranted in this manual can only be obtained by using the software components in the versions stated here.

The software components can be downloaded on the ifm electronic website: www.ifm.com

5.1.3 Licensing

With the purchase of the CP9031, the user simultaneously acquires a valid licence for the use of the ecomatDisplay SDK Utilities Library".



► Licence information → Installation routine or product page of the article on the ifm website.

5.2 ifm package



► Familiarise yourself with the following CODESYS functions:

Package Manager: → Online help > CODESYS Development System > Manage packages and licenses

5.2.1 Contents of the ifm package

The ifm package CP9031_VX.X.X.X.zip includes:

- the **CODESYS package** ifm_SDKUtilities_ecomatDisplay_Vx.x.x.x.package for installation on the PC with CODESYS.

- the **rpm package** "maintenance-sdk-x.x.xx.x-rx.aarch64.rpm" that must be installed on the ecomatDisplay.
- the pre-programmed CODESYS sample application "ifm Package Loader"
- the MaintenanceTool AddIn "Package Creator"

5.2.2 CODESYS package

5.2.2.1 Installing the CODESYS package

To install the ifm package "CODESYS für ifm SDK Extension":

- ✓ CODESYS3.5 SP16 Patch 4 has been correctly installed.
- ✓ The ifm package "ifm_SDKUtilities_ecomatDisplay_Vx.x.x.x.package" is stored on the PC.
- ▶ Start CODESYS with administrator rights.
 - ▷ CODESYS starts.
 - ▷ CODESYS user interface appears.
- ▶ Select [Tools] > [Package Manager] .
 - ▷ The [Package Manager] window appears.
- ▶ Click on [Install...]
- ▷ The file explorer is displayed.
- ▶ Select the "ifm_SDKUtilities_ecomatDisplay_Vx.x.x.x.package" file
- ▶ Click on [Open] and execute a complete installation.
 - ▷ "CODESYS für ifm SDK Extension" will be installed.
 - ▷ After successful installation: The [Package Manager] window shows the installed package.
 - ▷ The functions of the package can be used.
- ▶ Click on [Close] to quit the Package Manager.

5.2.2.2 Updating the CODESYS package

To update the ifm package "CODESYS für ifm SDK Extension":

- ▶ Uninstall the old version of the ifm package: Uninstalling the CODESYS package (→ □ 13)
- ▶ Install the new version of the ifm package: Installing the CODESYS package (→ □ 13)
- ▶ Open the project.
- ▶ In the device tree: Select the node of the device.
- ▶ Select [Project] > [Update Device...].
 - ▷ A dialogue window appears.
- ▶ Click on [Update Device] to start the update process.
 - ▷ CODESYS loads new device libraries.
 - ▷ Device tree view is updated.
- ▶ Click on [Close] to close the dialogue window.
- ▶ Save the project.

5.2.2.3 Uninstalling the CODESYS package

To install the ifm package "CODESYS für ifm SDK Extension":

- ▶ Start the Package Manager: Select [Tools] > [Package Manager].

- ▷ The [Package Manager] window shows the installed packages.
- ▶ Select the package to be uninstalled.
- ▶ Click on [Uninstall..].
 - ▷ The selected package will be uninstalled.
- ▶ Click on [Close] to quit the Package Manager.

5.2.3 rpm package for the ecomatDisplay

5.2.3.1 Memory allocation

The rpm package occupies the following memory on the ecomat display:

- RAM: approx. 80 MB
- FLASH: approx. 15 MB

5.2.3.2 Preparing ecomatDisplay

- ▶ Set the network settings of the ecomat display via the setup menu. → ecomatDisplay programming manual
- ▶ Connect the PC to the ecomatDisplay via the network.
- ▶ On the PC: Execute the command line command in Windows PowerShell to copy the rpm package to the directory `/home/root/` of the ecomatDisplay:
`scp -r .\maintenance-sdk-x.x.xx.x-rx.aarch64.rpm root@IP:/home/root`
IP: IP address of the ecomatDisplay

5.2.3.3 Installing the rpm package on the ecomatDisplay

ATTENTION

Installing the rpm package during operation

- ▷ The device may be destroyed!
 - ▶ Do not install the rpm package during operation.
 - ▶ Only install the rpm package during development.
 - ▶ Recommendation: Transfer the software on the device to other devices using the backup/restore function (software rollout).
-
- ▶ Log on to the ecomatDisplay with an SSH client (e.g. `putty`):
User `root`
Default password `pdm3`
 - ▶ Execute command line command in `putty` to change the directory:
`cd /home/root/`
 - ▶ Execute the command line command in `putty` to start the installation:
`dnf install maintenance-sdk-x.x.xx.x-rx.aarch64.rpm`
 - ▷ Information about the rpm package to be installed is displayed.
 - ▷ The query `Is this ok [y/N]` appears.
 - ▶ Confirm by entering `y` (= Yes).
 - ▷ The rpm package will be installed.
 - ▷ After successful installation, the following message will be displayed: `Complete!`
 - ▶ Restart the ecomatDisplay.

▷ The installation is completed.

Upgrading the rpm package

- ▶ Log on to the ecomatDisplay with an SSH client (e.g. `putty`):
User `root`
Default password `pdm3`
- ▶ Execute command line command in `putty` to change the directory:
`cd /home/root/`
- ▶ Execute the command line command in `putty` to start the installation:
`dnf upgrade maintenance-sdk-x.x.xx.x-rx.aarch64.rpm`

Deleting the rpm package

- ▶ Log on to the ecomatDisplay with an SSH client (e.g. `putty`):
User `root`
Default password `pdm3`
- ▶ Execute command line command in `putty` to change the directory:
`cd /home/root/`
- ▶ Execute the command line command in `putty` to delete the rpm package:
`dnf remove maintenance-sdk`



- ▶ Specify package without version and architecture.

Checking the installed rpm package

- ▶ Log on to the ecomatDisplay with an SSH client (e.g. `putty`):
User `root`
Default password `pdm3`
- ▶ Execute command line command in `putty` to change the directory:
`cd /home/root/`
- ▶ Execute command line command in `putty` to check the package installed on the device (installation and version):
`dnf list grep maintenance-sdk`



- ▶ Specify package without version and architecture.

▷ The rpm package is displayed with the version.

OR

- ▶ Execute the command line command in `putty` to check all packages installed on the device by name (installation and version):
`dnf list maintenance-sdk`



- ▶ Specify package without version and architecture.

▷ All rpm packages with the name are displayed with the version.

5.2.3.4 Transferring the software to other ecomatDisplays



To transfer the software of a fully installed and configured ecomatDisplay:

- ▶ Transfer the software on the device to other devices using the backup/restore function (software rollout)
 - ecomatDisplay programming manual, chapter Backup: Create a data backup
 - ecomatDisplay_BackupRestoreGuide_vx.x.pdf from the ecomatDisplay software package.

5.3 Adding a function library to the project

In order to be able to access the functions of the package "CODESYS for ifm SDK Extension" in a CODESYS project, the user must integrate the corresponding function library into the application.

✓ The "CODESYS for ifm SDK Extension" package has been installed correctly.

► Create a new project with ecomatDisplay.

OR:

► Open an existing project with ecomatDisplay.

► In the device tree: Double-click on [Application] > [Library Manager]

▷ The editor window shows the library manager of the application.

▷ The library manager shows the installed libraries.

► In the library manager: Click on [Add library].

▷ A dialogue window appears.

► In the tree view: Select [Target] > [ifm_CRxxxx] > [ifmSDKutil].

► Click on [OK] to confirm the selection.

▷ CODESYS integrates the selected library in the application.

▷ Functions of the library can be used for the programming.

5.4 Loading an example application to the ecomatDisplay

► Open and compile the sample application or template project in CODESYS.

► Load the application to the ecomatDisplay.

► Start the application on the ecomatDisplay.

5.5 Preparing the ecomatController

► Observe the programming manual and the original operating instructions of the ecomat controller.

► User settings and the assigned user levels and the associated passwords of the ecomatController must be known.

► Set the network settings of the ecomat controller. → ecomatController programming manual

6 Programming

6.1 Using library functions

The following function elements are available:

Function block	Description	Reference
PingDevice	Test the connection between the dialogue device and a controller	PingDevice (→ 21)
GetSysInfo	Read the system information of a device	GetSysInfo (→ 23)
Command	Control the operating mode of the device	Command (→ 25)
ReadFromController	Read firmware, application and data of a device and save them in a file	ReadFromController (→ 27)
GetSegmentListFromFile	Read the list of available software segments from a file	GetSegmentListFromFile (→ 29)
SelectSegmentFromList	Select a software segment from a read list	SelectSegmentFromList (→ 31)
WriteToController	Read firmware, application and data from a file and save them on the device	WriteToController (→ 33)
WriteInterfaceSettings	Change the node ID and CAN baud rate of the target device (Basic, R360).	WriteInterfaceSettings (→ 36)
Authentication	Logging in and out of the controller with the set user data	Authentication (→ 39)
GetSdkInterfaces	Searches for interfaces on the client system (ecomatDisplay) and retrieves a list of available communication interfaces	GetSdkInterfaces (→ 42)
GetSysInfo_ecomatController	Reading out system information from the ecomatController.	GetSysInfo_ecomatController (→ 44)
Command_ecomatController	Controlling the operating mode of the ecomatController	Command_ecomatController (→ 46)
ReadPackageFrom_ecomatController	Reading out the package from the ecomatController	ReadPackageFrom_ecomatController (→ 49)
UpdateFirmware_ecomatController	Installing a firmware update on the ecomatController	UpdateFirmware_ecomatController (→ 52)
WritePackageTo_ecomatController	Writing a package to the ecomatController	WritePackageTo_ecomatController (→ 55)
ReadIOSettings_ecomatController	Reading out the IO configuration from the ecomatController.	ReadIOSettings_ecomatController (→ 58)
WriteIOSettings_ecomatController	Writing the IO configuration to the ecomatController	WriteIOSettings_ecomatController (→ 60)
ReadInterfaceSettings_ecomatController	Reading out the interface configuration from the ecomatController	ReadInterfaceSettings_ecomatController (→ 62)
WriteInterfaceSettings_ecomatController	Writing the interface configuration to the ecomatController	WriteInterfaceSettings_ecomatController (→ 64)
GetDataFileList_ecomatController	Returns the list of all files located in the /data folder of the ecomatController.	GetDataFileList_ecomatController (→ 66)
ReadDataFiles_ecomatController	Reads one or more user files located in the /data folder of the ecomatController and saves them in a target path as a zip file	ReadDataFiles_ecomatController (→ 68)
WriteDataFiles_ecomatController	Writes one or more user files from a zip file to the /data folder of the ecomatController	WriteDataFiles_ecomatController (→ 71)
ReadDeviceLogs_ecomatController	Reads logging data from the ecomatController.	ReadDeviceLogs_ecomatController (→ 74)

Function block	Description	Reference
ReadSegmentFrom_ecomatController	Reads one or more software segments and user files from the ecomatController and saves them in a zip file	ReadSegmentFrom_ecomatController (→ 76)
WriteSegmentTo_ecomatController	Writes one or more software segments and user files from a zip file to the ecomatController	WriteSegmentTo_ecomatController (→ 79)
SetPasswords_ecomatController	Writes the password file into the ecomatController.	SetPasswords_ecomatController (→ 83)

7 ifm function libraries

This chapter contains the detailed description of the function libraries provided by ifm electronic for programming the device under CODESYS 3.5.

7.1 ifmSDKutil_ecomatDisplay library

This library contains all functions for the maintenance of an ifm controller with the help of an application on an ecomatDisplay.

Properties of the library

Field	Content
Company:	ifm electronic
Title:	ifmSDKutil_ecomatDisplay
Version:	According to ifm versioning guidelines, e.g.: 1.2.3.4
Categories:	Target ifm_CRnn MaintenanceSDK
Namespace:	ifmSDKutil
Author:	ifm electronic
Placeholder:	ifmSDKutil

7.1.1 Behaviour model of the ifm function blocks

This chapter describes the behaviour models of the function blocks from the ifmSDKutil_ecomatDisplay library.

7.1.1.1 General

ifm function blocks feature the following outputs to return status and error information:

Output	Description	
xError	TRUE	An error has occurred.
	FALSE	No error has occurred.
eDiagInfo	Diagnostic/error information Messages / diagnostic codes of the function blocks (→ 20)	

All inputs and outputs in the function block that belong to the ifm behaviour model are featured at the top.

Messages / diagnostic codes of the function blocks

Status/diagnostic/error messages of the function blocs are defined in the global Enum DIAG_INFO.

They have one of the following prefixes depending on the type of message:

Prefix	Type of message	Description
STAT	Signalled state:	Status messages contain information about the condition of the function block during the normal procedure.
DIAG	Diagnostic message	Diagnostic messages contain information about a failure event. They reset themselves after the failure event has disappeared and can optionally be evaluated by the application.
ERR	Error message	Error messages contain information about a failure event. They must be reset in the application after the failure event has disappeared.

Examples for messages / diagnostic codes:

- STAT_INACTIVE
- DIAG_OPEN_CIRCUIT
- ERR_OVERVOLTAGE



Lists of diagnostic codes are part of the function block descriptions.ifm function libraries (→ 19)

7.1.1.2 Behaviour model EXECUTE

Function blocks that have the EXECUTE behaviour model are processed once after a rising edge at the xExecute input.

The signal on xExecute must remain set to TRUE until xDone = TRUE or xError = TRUE. If the signal to xExecute becomes FALSE beforehand, the edit process of the function block is aborted without result.

New input values will only be adopted for the rising edge calculation when the currently executed action has been completed or cancelled.

If the function block has executed its function successfully, the output xDone will be set to TRUE for one cycle.

Response to errors

In case of an error, xError is set to TRUE and eDiagInfo indicates the error status as long as xExecute is = TRUE.

The output xDone is set to FALSE since the execution could not be finished successfully.

7.1.2 Function blocks

7.1.2.1 PingDevice

Function block type: Function block (FB)
Behaviour model: EXECUTE (→ Behaviour model EXECUTE □ 20)
Library: ifmSDKutil_ecomatDisplay

Symbol in CODESYS:



Description

The FB checks whether a connection can be established between the client device with SDK and the target device in the CAN network.

The FB sends a PING signal to a target device with a specific node ID (`usiDownloadID`) and a specific item number (`eDevice`) in the CAN network (`eChannel`).

If a target device with the corresponding node ID is active in the CAN network, it responds with the associated item number.



To receive a successful PING response:

- For the parameters `eDevice` and `usiDownloadID`, the correct values of the target device must be provided.

Input parameters

Name	Data type	Description	Value range
xExecute	BOOL	Control of the FB	- FALSE ⇒ TRUE: Execute the FB once. The values at the inputs will be adopted. Execute the FB (xExecute = TRUE) until the function block execution is successfully terminated (xDone = TRUE). If xDone = TRUE, then reset the input xExecute to FALSE. - TRUE ⇒ FALSE: Finish the FB processing.
eChannel	ifmDevice.CAN_CHANNEL	Identifier of the CAN Interface	Defined in: ifmDevice.CAN_CHANNEL (ENUM)
usiDownloadId	USINT	CAN node ID of the target device	1...127
eDevice	eDEVICES	Article number of the target device	Defined in eDEVICES (ENUM) (→ □ 86)

Output parameters

Name	Data type	Description	Value range
xDone	BOOL	Processing status	- FALSE: FB / FUN is inactive or being executed - TRUE: - FB / FUN was successfully executed - FB / FUN can be called again

Name	Data type	Description	Value range
xError	BOOL	Error indication	- FALSE: no error occurred or FB / FUN is still being executed - TRUE: - an error has occurred - The action could not be executed - Observe the diagnostic information
eDiagInfo	eDIAG_INFO	Diagnostic information	→ List below (diagnostic codes)
sArticle	STRING	Ping response. (Current item number of the connected device)	e.g. CR0403

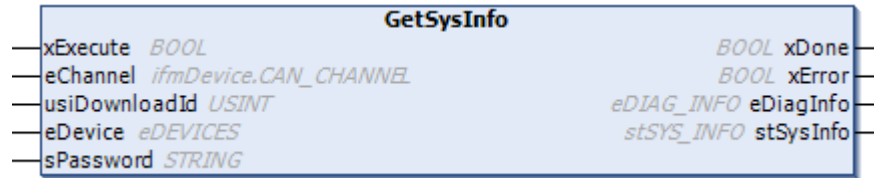
Diagnostic codes

• STAT_INACTIVE	Status: FB / function is inactive
• STAT_BUSY	Status: The FB / function is being executed
• STAT_DONE	Status: FB / function successfully processed; there are valid results on the outputs.
• STAT_ABORT	Status: Execution of the FB / the function is cancelled
• STAT_EVNT_OPEN_CHAN	Status: SDK-AP even <code>OpenChan</code> is opened.
• STAT_EVNT_OPEN_CLOS_CHAN	Status: SDK-API event <code>ClosChan</code> is opened.
• STAT_EVNT_OPEN_PING_DEVI	Status: Open Event ID: Ping
• STAT_EVNT_OPEN_DONE	Status: SDK-API event open.
• STAT_EVNT_REGI_OPEN_CHAN	Status: SDK-API event <code>OpenChan</code> registered.
• STAT_EVNT_REGI_OPEN_CHAN	Status: SDK-API event <code>ClosChan</code> registered.
• STAT_EVNT_REGI_DONE	Status: SDK-API event registered.
• STAT_EVNT_UNREGI	Status: SDK-API event will be unregistered.
• STAT_EVNT_UNREGI_DONE	Status: SDK-API event unregistered.
• STAT_EVNT_CLOS	Status: SDK-API event is closed.
• STAT_EVNT_CLOS_DONE	Status: SDK-API event closed.
• ERR_EVNT_OPEN	Error: occurred when an SDK-API event was opened
• ERR_EVNT_CLOS	Error: occurred when an event was closed
• ERR_EVNT_REGI	Error: occurred when the call-back function was registered
• ERR_EVNT_UNREGI	Error: occurred when the call-back function was unregistered.
• ERR_DYNAMIC_MEMORY	Error: Memory reservation failed during creation of the internal FB instance.
• ERR_CAN_INIT	Error: CAN channel is not initialised.
• ERR_INVALID_BAUD	Error: Invalid CAN baud rate.
• ERR_INVALID_DEVICE	Error: Wrong or unknown device
• ERR_INVALID_TARGET_ID	Error: Invalid target ID (e.g. CAN ID, IP address etc.)
• ERR_PROTOCOL	Error: Error in the maintenance command protocol ► Repeat the execution of the command.
• ERR_COMPATIBILITY	Error: Incompatible version of the hardware, file, runtime system, software or application
• ERR_UNKNOWN	Error: Unknown error
• ERR_INTERNAL	Error: Internal system error; contact the ifm Service Center!
• ERR_INVALID_VALUE	Error: at least 1 invalid input parameter or invalid combination of input parameters; processing was cancelled

7.1.2.2 GetSysInfo

Function block type: Function block (FB)
Behaviour model: EXECUTE (→ Behaviour model EXECUTE □ 20)
Library: ifmSDKutil_ecomatDisplay

Symbol in CODESYS:



Description

The FB reads the system information of the device (eChannel, usiDownloadId, eDevice).

As an option, the user can hand over the password for password-protected IEC applications (sPassword).

The read information is stored in a structure variable of the type stSYS_INFO (STRUCT).

Input parameters

Name	Data type	Description	Value range
xExecute	BOOL	Control of the FB	- FALSE ⇒ TRUE: Execute the FB once. The values at the inputs will be adopted. Execute the FB (xExecute = TRUE) until the function block execution is successfully terminated (xDone = TRUE). If xDone = TRUE, then reset the input xExecute to FALSE. - TRUE ⇒ FALSE: Finish the FB processing.
eChannel	ifmDevice.CAN_CHANNEL	Identifier of the CAN Interface	Defined in: ifmDevice.CAN_CHANNEL (ENUM)
usiDownloadId	USINT	CAN node ID of the target device	1...127
eDevice	eDEVICES	Article number of the target device	Defined in eDEVICES (ENUM) (→ □ 86)
ePassword	STRING	Optional: password for protected IEC applications	

Output parameters

Name	Data type	Description	Value range
xDone	BOOL	Processing status	- FALSE: FB / FUN is inactive or being executed - TRUE: - FB / FUN was successfully executed - FB / FUN can be called again
xError	BOOL	Error indication	- FALSE: no error occurred or FB / FUN is still being executed - TRUE: - an error has occurred - The action could not be executed - Observe the diagnostic information
eDiagInfo	eDIAG_INFO	Diagnostic information	→ List below (diagnostic codes)

Name	Data type	Description	Value range
stSysInfo	stSYS_INFO	Structure variable in which the read information is stored	Defined in: stSYS_INFO (→ 89)

Diagnostic codes

- | | |
|---------------------------------|---|
| • STAT_ABORT | Status: Execution of the FB / the function is cancelled |
| • STAT_BUSY | Status: The FB / function is being executed |
| • STAT_DONE | Status: FB / function successfully processed; there are valid results on the outputs. |
| • STAT_INACTIVE | Status: FB / function is inactive |
| • STAT_EVNT_OPEN_CLOS_CHAN | Status: SDK-API event <code>ClosChan</code> is opened. |
| • STAT_EVNT_OPEN_CHAN | Status: SDK-API event <code>OpenChan</code> is opened. |
| • STAT_EVNT_OPEN_READ_DEVI_INFO | Status: SDK-API event <code>ReadDeviInfo</code> is opened. |
| • STAT_EVNT_OPEN_READ_APP_INFO | Status: SDK-API event <code>ReadAppInfo</code> is opened. |
| • STAT_EVNT_OPEN_READ_CRC_INFO | Status: SDK-API event <code>ReadCrcInfo</code> is opened. |
| • STAT_EVNT_OPEN_DONE | Status: SDK-API event open. |
| • STAT_EVNT_REGI_OPEN_CHAN | Status: SDK-API event <code>OpenChan</code> registered. |
| • STAT_EVNT_REGI_OPEN_CHAN | Status: SDK-API event <code>ClosChan</code> registered. |

7.1.2.3 Command

Function block type: Function block (FB)
Behaviour model: EXECUTE (→ Behaviour model EXECUTE □ 20)
Library: ifmSDKutil_ecomatDisplay

Symbol in CODESYS:



Description

The FB is used to control the operating status of the application of the connected target device. The target device is selected by the inputs eChannel, usiDownloadId, eDevice. The command is set at the eCommand input.

Input parameters

Name	Data type	Description	Value range
xExecute	BOOL	Control of the FB	- FALSE ⇒ TRUE: Execute the FB once. The values at the inputs will be adopted. Execute the FB (xExecute = TRUE) until the function block execution is successfully terminated (xDone = TRUE). If xDone = TRUE, then reset the input xExecute to FALSE. - TRUE ⇒ FALSE: Finish the FB processing.
eChannel	ifmDevice.CAN_CHANNEL	Identifier of the CAN Interface	Defined in: ifmDevice.CAN_CHANNEL (ENUM)
usiDownloadId	USINT	CAN node ID of the target device	1...127
eDevice	eDEVICES	Article number of the target device	Defined in eDEVICES (ENUM) (→ □ 86)
eCommand	eCMD_LIST	Command to be executed.	Defined in: eCMD_LIST (ENUM) (→ □ 86)

Output parameters

Name	Data type	Description	Value range
xDone	BOOL	Processing status	- FALSE: FB / FUN is inactive or being executed - TRUE: - FB / FUN was successfully executed - FB / FUN can be called again
xError	BOOL	Error indication	- FALSE: no error occurred or FB / FUN is still being executed - TRUE: - an error has occurred - The action could not be executed - Observe the diagnostic information
eDiagInfo	eDIAG_INFO	Diagnostic information	→ List below (diagnostic codes)

Diagnostic codes

• STAT_ABORT	Status: Execution of the FB / the function is cancelled
• STAT_BUSY	Status: The FB / function is being executed
• STAT_DONE	Status: FB / function successfully processed; there are valid results on the outputs.
• STAT_EVT_OPEN_CLOS_CHAN	Status: SDK-API event <code>ClosChan</code> is opened.
• STAT_EVT_OPEN_CHAN	Status: SDK-API event <code>OpenChan</code> is opened.
• STAT_INACTIVE	Status: FB / function is inactive
• STAT_EVT_OPEN_STRT_STOP_APP	Status: SDK-API event <code>StrtStopApp</code> is opened.
• STAT_EVT_OPEN_RST_DEVI	Status: SDK-API event <code>RstDevi</code> is opened.
• STAT_EVT_OPEN_DONE	Status: SDK-API event open.
• STAT_EVT_REGI_DONE	Status: SDK-API event registered.
• STAT_EVT_UNREGI	Status: SDK-API event will be unregistered.
• STAT_EVT_UNREGI_DONE	Status: SDK-API event unregistered.
• STAT_EVT_CLOS	Status: SDK-API event is closed.
• STAT_EVT_CLOS_DONE	Status: SDK-API event closed.
• ERR_CAN_INIT	Error: CAN channel is not initialised.
• ERR_DYNAMIC_MEMORY	Error: Memory reservation failed during creation of the internal FB instance.
• ERR_EVT_CLOS	Error: occurred when an event was closed
• ERR_EVT_OPEN	Error: occurred when an SDK-API event was opened
• ERR_EVT_REGI	Error: occurred when the call-back function was registered
• ERR_EVT_UNREGI	Error: occurred when the call-back function was unregistered.
• ERR_INVALID_BAUD	Error: Invalid CAN baud rate.
• ERR_INVALID_DEVICE	Error: Wrong or unknown device
• ERR_INVALID_TARGET_ID	Error: Invalid target ID (e.g. CAN ID, IP address etc.)
• ERR_INVALID_VALUE	Error: at least 1 invalid input parameter or invalid combination of input parameters; processing was cancelled
• ERR_PROTOCOL	Error: Error in the maintenance command protocol ► Repeat the execution of the command.
• ERR_COMPATIBILITY	Error: Incompatible version of the hardware, file, runtime system, software or application
• ERR_CMND_NOT_SUPPORT	Error: At least one command is not supported by the device.
• ERR_UNKNOWN	Error: Unknown error
• ERR_INTERNAL	Error: Internal system error; contact the ifm Service Center!

7.1.2.4 ReadFromController

Function block type: Function block (FB)
Behaviour model: EXECUTE (→ Behaviour model EXECUTE □ 20)
Library: ifmSDKutil_ecomatDisplay

Symbol in CODESYS:



Description

The FB reads the data of a target device (eChannel, usiDownloadId, eDevice).

The user can determine which memory areas of the target device are to be read (stSegmList).

For access-protected CODESYS applications, the user can enter the corresponding password (sPassword).

The unread data is saved in a file (sFileName).

The FB displays the current status of the reading process (stReadSta).

Supported file formats: RESX, BINARY, H86

Input parameters

Name	Data type	Description	Value range
xExecute	BOOL	Control of the FB	- FALSE ⇒ TRUE: Execute the FB once. The values at the inputs will be adopted. Execute the FB (xExecute = TRUE) until the function block execution is successfully terminated (xDone = TRUE). If xDone = TRUE, then reset the input xExecute to FALSE. - TRUE ⇒ FALSE: Finish the FB processing.
eChannel	ifmDevice.CAN_CHANNEL	Identifier of the CAN Interface	Defined in: ifmDevice.CAN_CHANNEL (ENUM)
usiDownloadId	USINT	CAN node ID of the target device	1...127
eDevice	eDEVICES	Article number of the target device	Defined in eDEVICES (ENUM) (→ □ 86)
sFileName	STRING	Directory path and name of the BIN file	e.g. \data\MyData.bin
stSegmList	stSEGM_LIST	required memory segments	Defined in: stSEGM_LIST (→ □ 89)
ePassword	STRING	Optional: password for protected IEC applications	

Output parameters

Name	Data type	Description	Value range
xDone	BOOL	Processing status	- FALSE: FB / FUN is inactive or being executed - TRUE: - FB / FUN was successfully executed - FB / FUN can be called again

Name	Data type	Description	Value range
xError	BOOL	Error indication	- FALSE: no error occurred or FB / FUN is still being executed - TRUE: - an error has occurred - The action could not be executed - Observe the diagnostic information
eDiagInfo	eDIAG_INFO	Diagnostic information	→ List below (diagnostic codes)

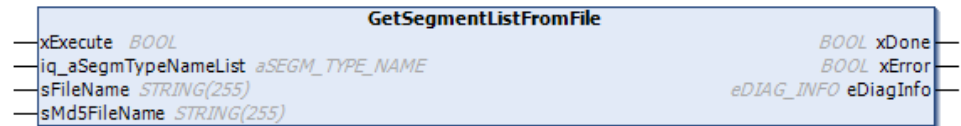
Diagnostic codes

• STAT_INACTIVE	Status: FB / function is inactive
• STAT_ABORT	Status: Execution of the FB / the function is cancelled
• STAT_BUSY	Status: The FB / function is being executed
• STAT_DONE	Status: FB / function successfully processed; there are valid results on the outputs.
• STAT_EVNT_OPEN_CHAN	Status: SDK-AP even <code>OpenChan</code> is opened.
• STAT_EVNT_OPEN_CLOS_CHAN	Status: SDK-API event <code>ClosChan</code> is opened.
• STAT_EVNT_OPEN_READ_SEGM	Status: SDK-API event <code>ReadSoftSegmFromDevi</code> is opened.
• STAT_EVNT_OPEN_WRIT_SEGM	Status: SDK-API event <code>WritDataSegmToFile</code> is opened.
• STAT_EVNT_OPEN_DONE	Status: SDK-API event open.
• STAT_EVNT_REGI_OPEN_CHAN	Status: SDK-API event <code>OpenChan</code> registered.
• STAT_EVNT_REGI_OPEN_CHAN	Status: SDK-API event <code>ClosChan</code> registered.
• STAT_EVNT_REGI_READ_SEGM	Status: SDK-API event <code>ReadSoftSegmFromDevi</code> is registered.
• STAT_EVNT_REGI_WRIT_SEGM	Status: SDK-API event <code>WritDataSegmToFile</code> registered.
• STAT_EVNT_REGI_DONE	Status: SDK-API event registered.
• STAT_EVNT_UNREGI	Status: SDK-API event will be unregistered.
• STAT_EVNT_UNREGI_DONE	Status: SDK-API event unregistered.
• STAT_EVNT_CLOS	Status: SDK-API event is closed.
• STAT_EVNT_CLOS_DONE	Status: SDK-API event closed.
• ERR_EVNT_CLOS	Error: occurred when an event was closed
• ERR_EVNT_OPEN	Error: occurred when an SDK-API event was opened
• ERR_EVNT_REGI	Error: occurred when the call-back function was registered
• ERR_EVNT_UNREGI	Error: occurred when the call-back function was unregistered.
• ERR_INVALID_VALUE	Error: at least 1 invalid input parameter or invalid combination of input parameters; processing was cancelled
• ERR_DYNAMIC_MEMORY	Error: Memory reservation failed during creation of the internal FB instance.
• ERR_CAN_INIT	Error: CAN channel is not initialised.
• ERR_INVALID_BAUD	Error: Invalid CAN baud rate.
• ERR_INVALID_DEVICE	Error: Wrong or unknown device
• ERR_INVALID_TARGET_ID	Error: Invalid target ID (e.g. CAN ID, IP address etc.)
• ERR_PROTOCOL	Error: Error in the maintenance command protocol ► Repeat the execution of the command.
• ERR_COMPATIBILITY	Error: Incompatible version of the hardware, file, runtime system, software or application
• ERR_INTERNAL	Error: Internal system error; contact the ifm Service Center!
• ERR_UNKNOWN	Error: Unknown error

7.1.2.5 GetSegmentListFromFile

Function block type: Function block (FB)
Behaviour model: EXECUTE (→ Behaviour model EXECUTE □ 20)
Library: ifmSDKutil_ecomatDisplay

Symbol in CODESYS:



Description

The FB reads all available software segments, their name and data types from one file (sFileName, sMd5Filename).

It stores the results in a data structure (iq_aSegmTypeNameList).

The data structure can store information of up to 16 software segments.

The selection of a software segment from the list is done by means of the dotted notation (iq_aSegmTypeNameList[n].xSele).



Supported file formats: RESX, BIN, H86

If the file is a BIN or H86 type file, the input sMd5FileName will not be evaluated.

Input parameters

Name	Data type	Description	Value range
xExecute	BOOL	Control of the FB	- FALSE ⇒ TRUE: Execute the FB once. The values at the inputs will be adopted. Execute the FB (xExecute = TRUE) until the function block execution is successfully terminated (xDone = TRUE). If xDone = TRUE, then reset the input xExecute to FALSE. - TRUE ⇒ FALSE: Finish the FB processing.
iq_aSegmTypeNameList	aSEGM_TYPE_NAME	Data structure for information about software segments	Defined in: aSEGM_TYPE_NAME (→ □ 102)
sFileName	STRING	Directory path and name of the BIN file	e.g. \data\MyData.bin
sMd5FileName	STRING	Directory path and name of the MD5 file	e.g. \data\MyData.md5

Output parameters

Name	Data type	Description	Value range
xDone	BOOL	Processing status	- FALSE: FB / FUN is inactive or being executed - TRUE: - FB / FUN was successfully executed - FB / FUN can be called again
xError	BOOL	Error indication	- FALSE: no error occurred or FB / FUN is still being executed - TRUE: - an error has occurred - The action could not be executed - Observe the diagnostic information

Name	Data type	Description	Value range
eDiagInfo	eDIAG_INFO	Diagnostic information	→ List below (diagnostic codes)

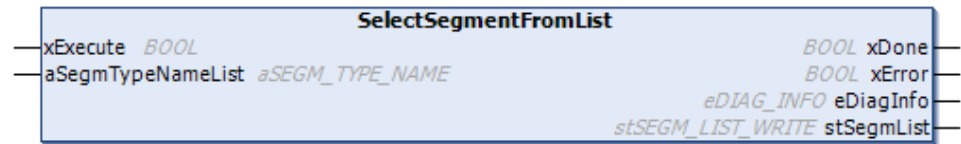
Diagnostic codes

• STAT_ABORT	Status: Execution of the FB / the function is cancelled
• STAT_BUSY	Status: The FB / function is being executed
• STAT_DONE	Status: FB / function successfully processed; there are valid results on the outputs.
• STAT_INACTIVE	Status: FB / function is inactive
• STAT_EVNT_OPEN_READ_FILE	Status: SDK-API event <code>ReadFileContent</code> is opened.
• STAT_EVNT_OPEN_DONE	Status: SDK-API event open.
• STAT_EVNT_REGI_READ_FILE	Status: SDK-API event <code>ReadFileContent</code> registered.
• STAT_EVNT_REGI_DONE	Status: SDK-API event registered.
• STAT_EVNT_UNREGI	Status: SDK-API event will be unregistered.
• STAT_EVNT_UNREGI_DONE	Status: SDK-API event unregistered.
• STAT_EVNT_CLOS	Status: SDK-API event is closed.
• STAT_EVNT_CLOS_DONE	Status: SDK-API event closed.
• ERR_EVNT_CLOS	Error: occurred when an event was closed
• ERR_EVNT_OPEN	Error: occurred when an SDK-API event was opened
• ERR_EVNT_REGI	Error: occurred when the call-back function was registered
• ERR_EVNT_UNREGI	Error: occurred when the call-back function was unregistered.
• ERR_BAD_CRC	Error: CRC error
• ERR_FILE_OPEN	Error: SDK-API reports an error when the file is opened.
• ERR_SEGM_COUNT_LIM	Error: The file contains more software segments than permitted or the file contains more than 2 versions of the same segment type.
• ERR_INVALID_VALUE	Error: at least 1 invalid input parameter or invalid combination of input parameters; processing was cancelled
• ERR_UNKNOWN	Error: Unknown error
• ERR_INTERNAL	Error: Internal system error; contact the ifm Service Center!

7.1.2.6 SelectSegmentFromList

Function block type: Function block (FB)
Behaviour model: EXECUTE (→ Behaviour model EXECUTE □ 20)
Library: ifmSDKutil_ecomatDisplay

Symbol in CODESYS:



Description

The FB reads a data structure (aSegmTypeNameList).

It creates a list of the software segments (stSegmList) from the read data.

At the same time, all data sets with xSele = TRUE are added to the list.

The list can be handed over to the FB WriteToController in order to write data to a controller.



The list from which the software segments are selected can be previously created with the FB GetSegmentListFromFile.

If the fields sType and sName of a dataset are empty, the dataset is not added to the list.

Input parameters

Name	Data type	Description	Value range
xExecute	BOOL	Control of the FB	- FALSE ⇒ TRUE: Execute the FB once. The values at the inputs will be adopted. Execute the FB (xExecute = TRUE) until the function block execution is successfully terminated (xDone = TRUE). If xDone = TRUE, then reset the input xExecute to FALSE. - TRUE ⇒ FALSE: Finish the FB processing.
aSegmTypeNameList	aSEGM_TYPE_NAME	Array with memory segments	Defined in: aSEGM_TYPE_NAME (→ □ 102)

Output parameters

Name	Data type	Description	Value range
xDone	BOOL	Processing status	- FALSE: FB / FUN is inactive or being executed - TRUE: - FB / FUN was successfully executed - FB / FUN can be called again
xError	BOOL	Error indication	- FALSE: no error occurred or FB / FUN is still being executed - TRUE: - an error has occurred - The action could not be executed - Observe the diagnostic information
eDiagInfo	eDIAG_INFO	Diagnostic information	→ List below (diagnostic codes)
stSegmList	stSEGM_LIST	required memory segments	Defined in: stSEGM_LIST (→ □ 89)

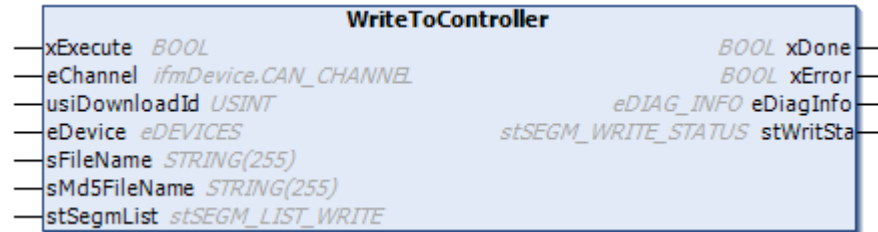
Diagnostic codes

• STAT_ABORT	Status: Execution of the FB / the function is cancelled
• STAT_BUSY	Status: The FB / function is being executed
• STAT_DONE	Status: FB / function successfully processed; there are valid results on the outputs.
• STAT_INACTIVE	Status: FB / function is inactive
• ERR_INVALID_VALUE	Error: at least 1 invalid input parameter or invalid combination of input parameters; processing was cancelled

7.1.2.7 WriteToController

Function block type: Function block (FB)
Behaviour model: EXECUTE (→ Behaviour model EXECUTE □ 20)
Library: ifmSDKutil_ecomatDisplay

Symbol in CODESYS:



Description

The FB writes firmware, applications or data on a target device (eChannel, usiDownloadId, eDevice).

The data to be written is read from a file with checksum (sFileName, sMd5FileName).

The user can select one or several memory areas to be used for writing (stSegmentList). The FB displays the current status of the write process (stWritSta).



The file must contain all the adequate data about the memory areas.

Supported file formats: RESX, BINARY, H86

Input parameters

Name	Data type	Description	Value range
xExecute	BOOL	Control of the FB	- FALSE ⇒ TRUE: Execute the FB once. The values at the inputs will be adopted. Execute the FB (xExecute = TRUE) until the function block execution is successfully terminated (xDone = TRUE). If xDone = TRUE, then reset the input xExecute to FALSE. - TRUE ⇒ FALSE: Finish the FB processing.
eChannel	ifmDevice.CAN_CHANNEL	Identifier of the CAN Interface	Defined in: ifmDevice.CAN_CHANNEL (ENUM)
usiDownloadId	USINT	CAN node ID of the target device	1...127
eDevice	eDEVICES	Article number of the target device	Defined in eDEVICES (ENUM) (→ □ 86)
sFileName	STRING	Directory path and name of the BIN file	e.g. \data\MyData.bin
sFileMd5	STRING	Directory path and name of the MD5 file	e.g. \data\MyData.md5
stSegmList	stSEGM_LIST	required memory segments	Defined in: stSEGM_LIST (→ □ 89)

Output parameters

Name	Data type	Description	Value range
xDone	BOOL	Processing status	- FALSE: FB / FUN is inactive or being executed - TRUE: - FB / FUN was successfully executed - FB / FUN can be called again

Name	Data type	Description	Value range
xError	BOOL	Error indication	- FALSE: no error occurred or FB / FUN is still being executed - TRUE: - an error has occurred - The action could not be executed - Observe the diagnostic information
eDiagInfo	eDIAG_INFO	Diagnostic information	→ List below (diagnostic codes)
stWritSta	stSEGM_WRITE_STATUS	Status of the writing process	Defined in: stSEGM_WRITE_STATUS (→ 91)

Diagnostic codes

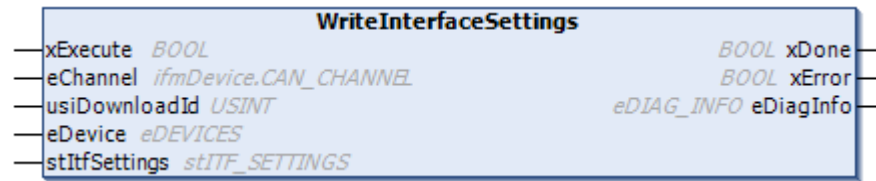
• STAT_ABORT	Status: Execution of the FB / the function is cancelled
• STAT_BUSY	Status: The FB / function is being executed
• STAT_DONE	Status: FB / function successfully processed; there are valid results on the outputs.
• STAT_INACTIVE	Status: FB / function is inactive
• STAT_EVNT_OPEN_CHAN	Status: SDK-AP even <code>OpenChan</code> is opened.
• STAT_EVNT_OPEN_CLOS_CHAN	Status: SDK-API event <code>ClosChan</code> is opened.
• STAT_EVNT_OPEN_READ_SEGM	Status: SDK-API event <code>ReadSoftSegmFromDevi</code> is opened.
• STAT_EVNT_OPEN_WRIT_SEGM	Status: SDK-API event <code>WritDataSegmToFile</code> is opened.
• STAT_EVNT_OPEN_DONE	Status: SDK-API event open.
• STAT_EVNT_REGI_OPEN_CHAN	Status: SDK-API event <code>OpenChan</code> registered.
• STAT_EVNT_REGI_OPEN_CHAN	Status: SDK-API event <code>ClosChan</code> registered.
• STAT_EVNT_REGI_READ_SEGM	Status: SDK-API event <code>ReadSoftSegmFromDevi</code> is registered.
• STAT_EVNT_REGI_WRIT_SEGM	Status: SDK-API event <code>WritDataSegmToFile</code> registered.
• STAT_EVNT_REGI_DONE	Status: SDK-API event registered.
• STAT_EVNT_UNREGI	Status: SDK-API event will be unregistered.
• STAT_EVNT_UNREGI_DONE	Status: SDK-API event unregistered.
• STAT_EVNT_CLOS	Status: SDK-API event is closed.
• STAT_EVNT_CLOS_DONE	Status: SDK-API event closed.
• ERR_CAN_INIT	Error: CAN channel is not initialised.
• ERR_DYNAMIC_MEMORY	Error: Memory reservation failed during creation of the internal FB instance.
• ERR_EVNT_CLOS	Error: occurred when an event was closed
• ERR_EVNT_OPEN	Error: occurred when an SDK-API event was opened
• ERR_EVNT_REGI	Error: occurred when the call-back function was registered
• ERR_EVNT_UNREGI	Error: occurred when the call-back function was unregistered.
• ERR_INVALID_VALUE	Error: at least 1 invalid input parameter or invalid combination of input parameters; processing was cancelled
• ERR_COMPATIBILITY	Error: Incompatible version of the hardware, file, runtime system, software or application
• ERR_INVALID_BAUD	Error: Invalid CAN baud rate.
• ERR_INVALID_DEVICE	Error: Wrong or unknown device
• ERR_INVALID_TARGET_ID	Error: Invalid target ID (e.g. CAN ID, IP address etc.)
• ERR_PROTOCOL	Error: Error in the maintenance command protocol ► Repeat the execution of the command.

- ERR_BAD_CRC Error: CRC error
- ERR_FILE_OPEN Error: SDK-API reports an error when the file is opened.
- ERR_INTERNAL Error: Internal system error; contact the ifm Service Center!
- ERR_UNKNOWN Error: Unknown error

7.1.2.8 WriteInterfaceSettings

Function block type: Function block (FB)
Behaviour model: EXECUTE (→ Behaviour model EXECUTE □ 20)
Library: ifmSDKutil_ecomatDisplay

Symbol in CODESYS:



Description

The FB is used to change the node ID and CAN baud rate of the specified CAN channel of the target device.



If the baud rate or node ID of the CAN channel of the target device is changed, the communication between the ecomatDisplay and the target device will be interrupted.

- ▶ Execute the FB again with the new communication settings.

ATTENTION

Multiple assignment of the node ID in the same CAN network

- ▷ Communication disruption
- ▶ The node ID must be unique in a CAN network.

ATTENTION

Different baud rate for different devices in the same CAN network

- ▷ Communication disruption
- ▶ The set baud rate of all devices in a CAN network must be identical.
- ▶ Change the baud rate of the devices one by one and meanwhile disconnect all other devices from the CAN network.



For devices of the Basic System type: Changing the node ID and baud rate is possible during operation without restarting the device.

For R360 Classic/Extended type controllers: After changing the node ID and baud rate, the device must be restarted.

Input parameters

Name	Data type	Description	Value range
xExecute	BOOL	Control of the FB	• FALSE ⇒ TRUE: Execute the FB once. The values at the inputs will be adopted. Execute the FB (xExecute = TRUE) until the function block execution is successfully terminated (xDone = TRUE). If xDone = TRUE, then reset the input xExecute to FALSE. • TRUE ⇒ FALSE: Finish the FB processing.

Name	Data type	Description	Value range
eChannel	ifmDevice. CAN_ CHANNEL	Identifier of the CAN Interface	Defined in: ifmDevice.CAN_CHANNEL (ENUM)
usiDownloadId	USINT	CAN node ID of the target device	1...127
eDevice	eDEVICES	Article number of the target device	Defined in eDEVICES (ENUM) (→ □ 86)
stlTfSettings	stlTF_ SETTINGS	List of interfaces of the controller incl. parameters	Defined in: stlTF_SETTINGS (→ □ 92)

Output parameters

Name	Data type	Description	Value range
xDone	BOOL	Processing status	- FALSE: FB / FUN is inactive or being executed - TRUE: - FB / FUN was successfully executed - FB / FUN can be called again
xError	BOOL	Error indication	- FALSE: no error occurred or FB / FUN is still being executed - TRUE: - an error has occurred - The action could not be executed - Observe the diagnostic information
eDiagInfo	eDIAG_ INFO	Diagnostic information	→ List below (diagnostic codes)

Diagnostic codes

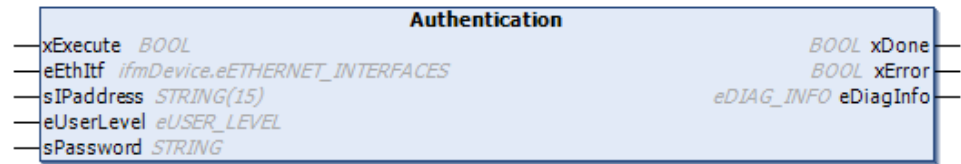
• STAT_INACTIVE	Status: FB / function is inactive
• STAT_BUSY	Status: The FB / function is being executed
• STAT_DONE	Status: FB / function successfully processed; there are valid results on the outputs.
• STAT_ABORT	Status: Execution of the FB / the function is cancelled
• STAT_EVNT_OPEN_CHAN	Status: SDK-AP even <code>OpenChan</code> is opened.
• STAT_EVNT_OPEN_CLOS_CHAN	Status: SDK-API event <code>ClosChan</code> is opened.
• STAT_EVNT_OPEN_CHANGE_NODE_ID	Status: SDK-API event <code>ChngDeviNodeId</code> is opened.
• STAT_EVNT_OPEN_CHANGE_BAUD_RATE	Status: SDK-API event <code>ChngDeviBaud</code> is opened.
• STAT_EVNT_OPEN_DONE	Status: SDK-API event open.
• STAT_EVNT_REGI_OPEN_CHAN	Status: SDK-API event <code>OpenChan</code> registered.
• STAT_EVNT_REGI_OPEN_CHAN	Status: SDK-API event <code>ClosChan</code> registered.
• STAT_EVNT_REGI_CHANGE_NODE_ID	Status: SDK-API event <code>ChngDeviNodeId</code> registered.
• STAT_EVNT_REGI_CHANGE_BAUDRATE	Status: SDK-API event <code>ChngDeviBaud</code> registered.
• STAT_EVNT_REGI_DONE	Status: SDK-API event registered.
• STAT_EVNT_UNREGI	Status: SDK-API event will be unregistered.
• STAT_EVNT_UNREGI_DONE	Status: SDK-API event unregistered.
• STAT_EVNT_CLOS	Status: SDK-API event is closed.
• STAT_EVNT_CLOS_DONE	Status: SDK-API event closed.
• ERR_EVNT_OPEN	Error: occurred when an SDK-API event was opened
• ERR_EVNT_REGI	Error: occurred when the call-back function was registered
• ERR_EVNT_UNREGI	Error: occurred when the call-back function was unregistered.

• ERR_EVNT_CLOS	Error: occurred when an event was closed
• ERR_INVALID_VALUE	Error: at least 1 invalid input parameter or invalid combination of input parameters; processing was cancelled
• ERR_DYNAMIC_MEMORY	Error: Memory reservation failed during creation of the internal FB instance.
• ERR_CAN_INIT	Error: CAN channel is not initialised.
• ERR_INVALID_BAUD	Error: Invalid CAN baud rate.
• ERR_INVALID_DEVICE	Error: Wrong or unknown device
• ERR_INVALID_TARGET_ID	Error: Invalid target ID (e.g. CAN ID, IP address etc.)
• ERR_PROTOCOL	Error: Error in the maintenance command protocol ► Repeat the execution of the command.
• ERR_COMPATIBILITY	Error: Incompatible version of the hardware, file, runtime system, software or application
• ERR_UNKNOWN	Error: Unknown error
• ERR_INTERNAL	Error: Internal system error; contact the ifm Service Center!

7.1.2.9 Authentication

Function block type: Function block (FB)
Behaviour model: EXECUTE (→ Behaviour model EXECUTE □ 20)
Library: ifmSDKutil_ecomatDisplay

Symbol in CODESYS:



Description

The FB is used for authentication on an ecomatController:

- On rising edge at FB input `xExecute` : Log in to the ecomatController with the set login information.
- With a falling edge at the FB input `xExecute` : Log out the current user.



The ecomatController offers a user-based security management system with different authentication levels. It consists of various predefined user accounts that have certain user rights on the system. More information: → ecomatController programming manual

If a password is configured for the user on the machine:

- Log in to the device with user name and password before using functions from this library that require logging in.

Input parameters

Name	Data type	Description	Value range
xExecute	BOOL	Control of the FB	• FALSE ⇒ TRUE: Execute the FB once. The values at the inputs will be adopted. Execute the FB (xExecute = TRUE) until the function block execution is successfully terminated (xDone = TRUE). If xDone = TRUE, then reset the input xExecute to FALSE. • TRUE ⇒ FALSE: Finish the FB processing.
eEthItf	ifmDevice.eETHERNET_INTERFACES	Ethernet interface of the dialogue unit, e.g. ecomatDisplay	Defined in: → ifmDevice.eETHERNET_INTERFACES
sIPaddress	STRING (15)	IP address of the target device, e.g. ecomatController	Default value: 192.168.82.247
ePassword	STRING	Optional: password for protected IEC applications	

Output parameters

Name	Data type	Description	Value range
xDone	BOOL	Processing status	• FALSE: FB / FUN is inactive or being executed • TRUE: – FB / FUN was successfully executed – FB / FUN can be called again

Name	Data type	Description	Value range
xError	BOOL	Error indication	- FALSE: no error occurred or FB / FUN is still being executed - TRUE: - an error has occurred - The action could not be executed - Observe the diagnostic information
eDiagInfo	eDIAG_INFO	Diagnostic information	→ List below (diagnostic codes)

Diagnostic codes

• STAT_INACTIVE	Status: FB / function is inactive
• STAT_ABORT	Status: Execution of the FB / the function is cancelled
• STAT_BUSY	Status: The FB / function is being executed
• STAT_DONE	Status: FB / function successfully processed; there are valid results on the outputs.
• STAT_EVNT_OPEN_CHAN_ETH	Status: SDK-API event <code>r3603_OpenChannel</code> is opened.
• STAT_EVNT_OPEN_CLOS_CHAN_ETH	Status: SDK-API event <code>r3603_CloseChannel</code> is opened.
• STAT_EVNT_OPEN_LOGIN	Status: SDK-API event <code>r3603_Login</code> is opened.
• STAT_EVNT_OPEN_LOGOUT	Status: SDK-API event <code>r3603_Logout</code> is opened.
• STAT_EVNT_OPEN_READ_AUTH	Status: SDK-API event <code>r3603_ReadAuthenticationState</code> is opened.
• STAT_EVNT_OPEN_DONE	Status: SDK-API event open.
• STAT_EVNT_REGI_OPEN_CHAN_ETH	Status: SDK-API event <code>r3603_OpenChannel</code> registered.
• STAT_EVNT_REGI_CLOS_CHAN_ETH	Status: SDK-API event <code>r3603_CloseChannel</code> registered.
• STAT_EVNT_REGI_LOGIN	Status: SDK-API event <code>r3603_Login</code> registered.
• STAT_EVNT_REGI_LOGOUT	Status: SDK-API event <code>r3603_Logout</code> registered.
• STAT_EVNT_REGI_READ_AUTH	Status: SDK-API event <code>ReadAuthenticationState</code> registered.
• STAT_EVNT_REGI_DONE	Status: SDK-API event registered.
• STAT_EVNT_UNREGI	Status: SDK-API event will be unregistered.
• STAT_EVNT_UNREGI_DONE	Status: SDK-API event unregistered.
• STAT_EVNT_CLOS	Status: SDK-API event is closed.
• STAT_EVNT_CLOS_DONE	Status: SDK-API event closed.
• ERR_EVNT_CLOS	Error: occurred when an event was closed
• ERR_EVNT_OPEN	Error: occurred when an SDK-API event was opened
• ERR_EVNT_REGI	Error: occurred when the call-back function was registered
• ERR_EVNT_UNREGI	Error: occurred when the call-back function was unregistered.
• ERR_DYNAMIC_MEMORY	Error: Memory reservation failed during creation of the internal FB instance.
• ERR_INVALID_VALUE	Error: at least 1 invalid input parameter or invalid combination of input parameters; processing was cancelled
• ERR_COMMUNICATION	Error: Communication error
• ERR_COMM_INTERFACE	Error: Communication interface error
• ERR_INVALID_TARGET_ID	Error: Invalid target ID (e.g. CAN ID, IP address etc.)
• ERR_INVALID_DEVICE	Error: Wrong or unknown device
• ERR_COMPATIBILITY	Error: Incompatible version of the hardware, file, runtime system, software or application
• ERR_INVALID_AUTH	Error: Invalid or missing authentication
• ERR_CMND_FAILED	Error: Execution of the command has failed.

- ERR_UNKNOWN Error: Unknown error
- ERR_INTERNAL Error: Internal system error; contact the ifm Service Center!

7.1.2.10 GetSdkInterfaces

Function block type: Function block (FB)
Behaviour model: EXECUTE (→ Behaviour model EXECUTE □ 20)
Library: ifmSDKutil_ecomatDisplay

Symbol in CODESYS:



Description

The FB searches for SDK interfaces on the client system (ecomatDisplay) and retrieves a list of available communication interfaces.

Input parameters

Name	Data type	Description	Value range
xExecute	BOOL	Control of the FB	- FALSE ⇒ TRUE: Execute the FB once. The values at the inputs will be adopted. Execute the FB (xExecute = TRUE) until the function block execution is successfully terminated (xDone = TRUE). If xDone = TRUE, then reset the input xExecute to FALSE. - TRUE ⇒ FALSE: Finish the FB processing.
eEthlIf	ifmDevice.eETHERNET_INTERFACES	Ethernet interface of the dialogue unit, e.g. ecomatDisplay	Defined in: → ifmDevice.eETHERNET_INTERFACES
sIPaddress	STRING (15)	IP address of the target device, e.g. ecomatController	Default value: 192.168.82.247

Output parameters

Name	Data type	Description	Value range
xDone	BOOL	Processing status	- FALSE: FB / FUN is inactive or being executed - TRUE: - FB / FUN was successfully executed - FB / FUN can be called again
xError	BOOL	Error indication	- FALSE: no error occurred or FB / FUN is still being executed - TRUE: - an error has occurred - The action could not be executed - Observe the diagnostic information
eDiagInfo	eDIAG_INFO	Diagnostic information	→ List below (diagnostic codes)
aIntf	aINTF_LIST	List with available communication interfaces	Defined in: aINTF_LIST (→ □ 102)

Diagnostic codes

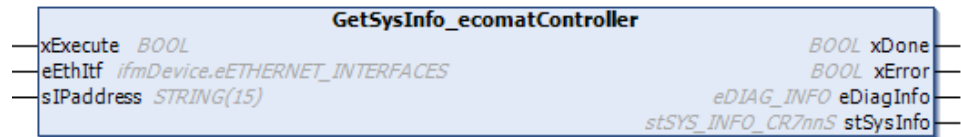
- STAT_INACTIVE Status: FB / function is inactive
- STAT_BUSY Status: The FB / function is being executed

• STAT_DONE	Status: FB / function successfully processed; there are valid results on the outputs.
• STAT_ABORT	Status: Execution of the FB / the function is cancelled
• STAT_EVT_OPEN_GET_INTF	Status: SDK-API event <code>GetIntf</code> is opened.
• STAT_EVT_OPEN_DONE	Status: SDK-API event open.
• STAT_EVT_REGI_GET_INTF	Status: SDK-API event <code>GetIntf</code> registered.
• STAT_EVT_REGI_DONE	Status: SDK-API event registered.
• STAT_EVT_UNREGI	Status: SDK-API event will be unregistered.
• STAT_EVT_UNREGI_DONE	Status: SDK-API event unregistered.
• STAT_EVT_CLOS	Status: SDK-API event is closed.
• STAT_EVT_CLOS_DONE	Status: SDK-API event closed.
• ERR_EVT_OPEN	Error: occurred when an SDK-API event was opened
• ERR_EVT_REGI	Error: occurred when the call-back function was registered
• ERR_EVT_UNREGI	Error: occurred when the call-back function was unregistered.
• ERR_EVT_CLOS	Error: occurred when an event was closed
• ERR_INVALID_VALUE	Error: at least 1 invalid input parameter or invalid combination of input parameters; processing was cancelled
• ERR_COMM_INTERFACE	Error: Communication interface error
• ERR_INVALID_DEVICE	Error: Wrong or unknown device
• ERR_COMPATIBILITY	Error: Incompatible version of the hardware, file, runtime system, software or application
• ERR_CMND_FAILED	Error: Execution of the command has failed.
• ERR_UNKNOWN	Error: Unknown error
• ERR_INTERNAL	Error: Internal system error; contact the ifm Service Center!

7.1.2.11 GetSysInfo_ecomatController

Function block type: Function block (FB)
Behaviour model: EXECUTE (→ Behaviour model EXECUTE □ 20)
Library: ifmSDKutil_ecomatDisplay

Symbol in CODESYS:



Description

This FB reads the following information from the target device (ecomatController):

- Software information: Returns information about the current software.
- Device information: Returns detailed information about the hardware and network interface of the machine.



The response to this command depends on the type of device, the hardware status and the firmware version of the device.

Input parameters

Name	Data type	Description	Value range
xExecute	BOOL	Control of the FB	• FALSE ⇒ TRUE: Execute the FB once. The values at the inputs will be adopted. Execute the FB (xExecute = TRUE) until the function block execution is successfully terminated (xDone = TRUE). If xDone = TRUE, then reset the input xExecute to FALSE. • TRUE ⇒ FALSE: Finish the FB processing.
eEthIf	ifmDevice.eETHERNET_INTERFACES	Ethernet interface of the dialogue unit, e.g. ecomatDisplay	Defined in: → ifmDevice.eETHERNET_INTERFACES
sIPaddress	STRING (15)	IP address of the target device, e.g. ecomatController	Default value: 192.168.82.247

Output parameters

Name	Data type	Description	Value range
xDone	BOOL	Processing status	• FALSE: FB / FUN is inactive or being executed • TRUE: – FB / FUN was successfully executed – FB / FUN can be called again
xError	BOOL	Error indication	• FALSE: no error occurred or FB / FUN is still being executed • TRUE: – an error has occurred – The action could not be executed – Observe the diagnostic information
eDiagInfo	eDIAG_INFO	Diagnostic information	→ List below (diagnostic codes)
stSysInfo	stSYS_INFO	Structure variable in which the read information is stored	Defined in: stSYS_INFO (→ □ 89)

Diagnostic codes

• STAT_INACTIVE	Status: FB / function is inactive
• STAT_BUSY	Status: The FB / function is being executed
• STAT_DONE	Status: FB / function successfully processed; there are valid results on the outputs.
• STAT_ABORT	Status: Execution of the FB / the function is cancelled
• STAT_EVNT_OPEN_CHAN_ETH	Status: SDK-API event <code>r3603_OpenChannel</code> is opened.
• STAT_EVNT_OPEN_CLOS_CHAN_ETH	Status: SDK-API event <code>r3603_CloseChannel</code> is opened.
• STAT_EVNT_OPEN_READ_DEVI_INFO	Status: SDK-API event <code>ReadDeviInfo</code> is opened.
• STAT_EVNT_OPEN_READ_SW_INFO	Status: SDK-API event <code>r3603_ReadSoftwareList</code> is opened.
• STAT_EVNT_REGI_DONE	Status: SDK-API event registered.
• STAT_EVNT_UNREGI	Status: SDK-API event will be unregistered.
• STAT_EVNT_UNREGI_DONE	Status: SDK-API event unregistered.
• STAT_EVNT_CLOS	Status: SDK-API event is closed.
• STAT_EVNT_CLOS_DONE	Status: SDK-API event closed.
• ERR_EVNT_OPEN	Error: occurred when an SDK-API event was opened
• ERR_EVNT_REGI	Error: occurred when the call-back function was registered
• ERR_EVNT_UNREGI	Error: occurred when the call-back function was unregistered.
• ERR_EVNT_CLOS	Error: occurred when an event was closed
• ERR_INVALID_VALUE	Error: at least 1 invalid input parameter or invalid combination of input parameters; processing was cancelled
• ERR_DYNAMIC_MEMORY	Error: Memory reservation failed during creation of the internal FB instance.
• ERR_COMMUNICATION	Error: Communication error
• ERR_COMM_INTERFACE	Error: Communication interface error
• ERR_INVALID_TARGET_ID	Error: Invalid target ID (e.g. CAN ID, IP address etc.)
• ERR_INVALID_DEVICE	Error: Wrong or unknown device
• ERR_COMPATIBILITY	Error: Incompatible version of the hardware, file, runtime system, software or application
• ERR_INVALID_AUTH	Error: Invalid or missing authentication
• ERR_CMND_FAILED	Error: Execution of the command has failed.
• ERR_UNKNOWN	Error: Unknown error
• ERR_INTERNAL	Error: Internal system error; contact the ifm Service Center!

7.1.2.12 Command_ecomatController

Function block type: Function block (FB)
Behaviour model: EXECUTE (→ Behaviour model EXECUTE □ 20)
Library: ifmSDKutil_ecomatDisplay

Symbol in CODESYS:



Description

The FB is used to execute the following commands on the target device:

- Start the application: Starts the IEC standard application in the device.
- Stop application: Stops the IEC standard application and the IEC safety application in the device.
- Delete the application: Deletes the IEC standard application and the IEC safety application in the device including the retain data.
- Reboot: PowerOn reset or restart command

The target device is set through the inputs `eEthItf` and `sIPaddress`.

The command is set at the `eCommand` input.



▶ Do not execute the FB if the device is in bootloader mode, otherwise an error will be returned.

Input parameters

Name	Data type	Description	Value range
xExecute	BOOL	Control of the FB	• FALSE ⇒ TRUE: Execute the FB once. The values at the inputs will be adopted. Execute the FB (xExecute = TRUE) until the function block execution is successfully terminated (xDone = TRUE). If xDone = TRUE, then reset the input xExecute to FALSE. • TRUE ⇒ FALSE: Finish the FB processing.
eEthItf	ifmDevice.eETHERNET_INTERFACES	Ethernet interface of the dialogue unit, e.g. ecomatDisplay	Defined in: → ifmDevice.eETHERNET_INTERFACES
sIPaddress	STRING (15)	IP address of the target device, e.g. ecomatController	Default value: 192.168.82.247
eCommand	eCMD_LIST_CR7nnS	Command to be executed.	Defined in: eCMD_LIST_CR7nnS (ENUM) (→ □ 87)

Output parameters

Name	Data type	Description	Value range
xDone	BOOL	Processing status	• FALSE: FB / FUN is inactive or being executed • TRUE: – FB / FUN was successfully executed – FB / FUN can be called again

Name	Data type	Description	Value range
xError	BOOL	Error indication	- FALSE: no error occurred or FB / FUN is still being executed - TRUE: - an error has occurred - The action could not be executed - Observe the diagnostic information
eDiagInfo	eDIAG_INFO	Diagnostic information	→ List below (diagnostic codes)

Diagnostic codes

• STAT_INACTIVE	Status: FB / function is inactive
• STAT_BUSY	Status: The FB / function is being executed
• STAT_DONE	Status: FB / function successfully processed; there are valid results on the outputs.
• STAT_ABORT	Status: Execution of the FB / the function is cancelled
• STAT_EVNT_OPEN_CHAN_ETH	Status: SDK-API event <code>r3603_OpenChannel</code> is opened.
• STAT_EVNT_OPEN_CLOS_CHAN_ETH	Status: SDK-API event <code>r3603_CloseChannel</code> is opened.
• STAT_EVNT_OPEN_STRT_STOP_APP	Status: SDK-API event <code>r3603_ExecStartStopApplication</code> is opened.
• STAT_EVNT_OPEN_RST_DEVI	Status: SDK-API event <code>r3603_ExecResetDevice</code> is opened.
• STAT_EVNT_OPEN_DELE_SEGM	Status: SDK-API event <code>r3603_ExecDeleteSoftwareSegment</code> is opened.
• STAT_EVNT_OPEN_DONE	Status: SDK-API event open.
• STAT_EVNT_REGI_OPEN_CHAN_ETH	Status: SDK-API event <code>r3603_OpenChannel</code> registered.
• STAT_EVNT_REGI_CLOS_CHAN_ETH	Status: SDK-API event <code>r3603_CloseChannel</code> registered.
• STAT_EVNT_REGI_STRT_STOP_APP	Status: SDK-API event <code>r3603_ExecStartStopApplication</code> registered.
• STAT_EVNT_REGI_RST_DEVI	Status: SDK-API event <code>r3603_ExecResetDevice</code> registered.
• STAT_EVNT_REGI_DELE_SEGM	Status: SDK-API event <code>r3603_ExecDeleteSoftwareSegment</code> registered.
• STAT_EVNT_REGI_DONE	Status: SDK-API event registered.
• STAT_EVNT_UNREGI	Status: SDK-API event will be unregistered.
• STAT_EVNT_UNREGI_DONE	Status: SDK-API event unregistered.
• STAT_EVNT_CLOS	Status: SDK-API event is closed.
• STAT_EVNT_CLOS_DONE	Status: SDK-API event closed.
• ERR_EVNT_OPEN	Error: occurred when an SDK-API event was opened
• ERR_EVNT_REGI	Error: occurred when the call-back function was registered
• ERR_EVNT_UNREGI	Error: occurred when the call-back function was unregistered.
• ERR_EVNT_CLOS	Error: occurred when an event was closed
• ERR_INVALID_VALUE	Error: at least 1 invalid input parameter or invalid combination of input parameters; processing was cancelled
• ERR_DYNAMIC_MEMORY	Error: Memory reservation failed during creation of the internal FB instance.
• ERR_COMMUNICATION	Error: Communication error
• ERR_COMM_INTERFACE	Error: Communication interface error
• ERR_INVALID_TARGET_ID	Error: Invalid target ID (e.g. CAN ID, IP address etc.)
• ERR_INVALID_DEVICE	Error: Wrong or unknown device
• ERR_COMPATIBILITY	Error: Incompatible version of the hardware, file, runtime system, software or application
• ERR_INVALID_AUTH	Error: Invalid or missing authentication
• ERR_CMND_FAILED	Error: Execution of the command has failed.

- ERR_UNKNOWN Error: Unknown error
- ERR_INTERNAL Error: Internal system error; contact the ifm Service Center!

7.1.2.13 ReadPackageFrom_ecomatController

Function block type: Function block (FB)
Behaviour model: EXECUTE (→ Behaviour model EXECUTE □ 20)
Library: ifmSDKutil_ecomatDisplay

Symbol in CODESYS:



Description

This FB reads software segments from an ecomatController and saves them.

For each segment type selected at the FB input `stSegmList`, the FB generates a zip file containing all configuration files and associated data for the selected segment type.

Segment type	file name
Firmware	<code>sFileName + _ifmOS.zip</code>
Standard application	<code>sFileName + _stdapp.zip</code>
Safety application:	<code>sFileName + _safeapp.zip</code>

- The files read out can be written to other ecomatControllers of the same design.
- Data transmission takes a certain amount of time. While reading from the controller: check the status messages at the FB output `stReadSta`.
- An error will be returned if the selected segment is not available on the device.
- If a file with the same name already exists in the destination path, the content of the existing file will be overwritten.
- Segments from the device are read in the following order: Firmware, standard application, safety application.



So that the FB can access the target device:

- ▶ Before executing the FB: Log in using the FB `Authentication` with the required user rights (user level) and the corresponding password on the target device.
Further information on the required user rights: → Programming manual of the target device



- ▶ Do not use this FB to save more than one segment in a single file, otherwise the existing segments will be overwritten by the new segment.

Input parameters

Name	Data type	Description	Value range
xExecute	BOOL	Control of the FB	• FALSE ⇒ TRUE: Execute the FB once. The values at the inputs will be adopted. Execute the FB (xExecute = TRUE) until the function block execution is successfully terminated (xDone = TRUE). If xDone = TRUE, then reset the input xExecute to FALSE. • TRUE ⇒ FALSE: Finish the FB processing.
eEthIf	ifmDevice.eETHERNET_INTERFACES	Ethernet interface of the dialogue unit, e.g. ecomatDisplay	Defined in: → ifmDevice.eETHERNET_INTERFACES
sIPAddress	STRING (15)	IP address of the target device, e.g. ecomatController	Default value: 192.168.82.247

Name	Data type	Description	Value range
sFileName	STRING	Directory path and name of the BIN file	e.g. \data\MyData.bin
stPackageLst	stPACKAGE_LIST_CR7nnS	List of software packages that are read by the device. If no selection is made, no read operation is triggered.	Defined in: stPACKAGE_LIST_CR7nnS (→ 97)

Output parameters

Name	Data type	Description	Value range
xDone	BOOL	Processing status	- FALSE: FB / FUN is inactive or being executed - TRUE: - FB / FUN was successfully executed - FB / FUN can be called again
xError	BOOL	Error indication	- FALSE: no error occurred or FB / FUN is still being executed - TRUE: - an error has occurred - The action could not be executed - Observe the diagnostic information
eDiagInfo	eDIAG_INFO	Diagnostic information	→ List below (diagnostic codes)
stReadSta	stSEGM_READ_STATUS	Status of the reading process	Defined in: stSEGM_READ_STATUS (→ 91)

Diagnostic codes

• STAT_INACTIVE	Status: FB / function is inactive
• STAT_BUSY	Status: The FB / function is being executed
• STAT_DONE	Status: FB / function successfully processed; there are valid results on the outputs.
• STAT_ABORT	Status: Execution of the FB / the function is cancelled
• STAT_SELE_SEGM	Status: Software segment selection in progress.
• STAT_READ_SEGM	Status: Reading the software segment from the device is in progress.
• STAT_EVNT_OPEN_CHAN_ETH	Status: SDK-API event <code>r3603_OpenChannel</code> is opened.
• STAT_EVNT_OPEN_CLOS_CHAN_ETH	Status: SDK-API event <code>r3603_CloseChannel</code> is opened.
• STAT_EVNT_OPEN_READ_SEGM	Status: SDK-API event <code>ReadSoftwarePackageFromDevice</code> is opened.
• STAT_EVNT_OPEN_DONE	Status: SDK-API event open.
• STAT_EVNT_REGI_OPEN_CHAN_ETH	Status: SDK-API event <code>r3603_OpenChannel</code> registered.
• STAT_EVNT_REGI_CLOS_CHAN_ETH	Status: SDK-API event <code>r3603_CloseChannel</code> registered.
• STAT_EVNT_REGI_READ_SEGM	Status: SDK-API event <code>r3603_ReadSoftwarePackageFromDevice</code> registered.
• ERR_EVNT_OPEN	Error: occurred when an SDK-API event was opened
• ERR_EVNT_REGI	Error: occurred when the call-back function was registered
• ERR_EVNT_UNREGI	Error: occurred when the call-back function was unregistered.
• ERR_EVNT_CLOS	Error: occurred when an event was closed
• ERR_INVALID_VALUE	Error: at least 1 invalid input parameter or invalid combination of input parameters; processing was cancelled
• ERR_DYNAMIC_MEMORY	Error: Memory reservation failed during creation of the internal FB instance.
• ERR_COMMUNICATION	Error: Communication error

• ERR_COMM_INTERFACE	Error: Communication interface error
• ERR_INVALID_TARGET_ID	Error: Invalid target ID (e.g. CAN ID, IP address etc.)
• ERR_INVALID_DEVICE	Error: Wrong or unknown device
• ERR_COMPATIBILITY	Error: Incompatible version of the hardware, file, runtime system, software or application
• ERR_INVALID_AUTH	Error: Invalid or missing authentication
• ERR_CMND_FAILED	Error: Execution of the command has failed.
• ERR_NO_SOFTWARE	Error: The software is not available on the target device.
• ERR_READ	Error: Error reading a segment or package from the device.
• ERR_UNKNOWN	Error: Unknown error
• ERR_INTERNAL	Error: Internal system error; contact the ifm Service Center!

7.1.2.14 UpdateFirmware_ecomatController

Function block type: Function block (FB)
Behaviour model: EXECUTE (→ Behaviour model EXECUTE □ 20)
Library: ifmSDKutil_ecomatDisplay

Symbol in CODESYS:



Description

ATTENTION

Aborting the firmware update process

- ▷ Damage to the device possible
- ▶ Do not cancel the execution of the FB during the firmware update process.
- ▶ Set the value of the FB input `xExecute` to TRUE during the entire firmware update process.
- ▶ Only end the execution of the FB (`xExecute = FALSE`) when the FB outputs are `xDone = TRUE` or `xError = TRUE` .

The FB loads the firmware update (zip package file) provided by ifm electronic onto the ecomatController.

- Data transmission takes a certain amount of time.
- The firmware update is successfully completed when the FB output is `xDone = TRUE` .
- The firmware update is finished with error if the FB output is `xError = TRUE` .
- ▶ While loading the firmware onto the controller: check the status messages to the FB outputs `usiProgress` and `sRespDesc` .
- ▶ The source file type at the FB input `sFileName` that is to be read and sent to the controller is indicated with file extension `.zip` .



So that the FB can access the target device:

- ▶ Before executing the FB: Log in using the FB `Authentication` with the required user rights (user level) and the corresponding password on the target device.
 Further information on the required user rights: → Programming manual of the target device

Input parameters

Name	Data type	Description	Value range
xExecute	BOOL	Control of the FB	• FALSE ⇒ TRUE: Execute the FB once. The values at the inputs will be adopted. Execute the FB (<code>xExecute = TRUE</code>) until the function block execution is successfully terminated (<code>xDone = TRUE</code>). If <code>xDone = TRUE</code>, then reset the input <code>xExecute</code> to FALSE. • TRUE ⇒ FALSE: Finish the FB processing.
eEthItf	ifmDevice.eETHERNET_INTERFACES	Ethernet interface of the dialogue unit, e.g. ecomatDisplay	Defined in: → ifmDevice.eETHERNET_INTERFACES

Name	Data type	Description	Value range
slAddress	STRING (15)	IP address of the target device, e.g. ecomatController	Default value: 192.168.82.247
sFileName	STRING	Directory path and name of the BIN file	e.g. \data\MyData.bin

Output parameters

Name	Data type	Description	Value range
xDone	BOOL	Processing status	- FALSE: FB / FUN is inactive or being executed - TRUE: - FB / FUN was successfully executed - FB / FUN can be called again
xError	BOOL	Error indication	- FALSE: no error occurred or FB / FUN is still being executed - TRUE: - an error has occurred - The action could not be executed - Observe the diagnostic information
eDiagInfo	eDIAG_INFO	Diagnostic information	→ List below (diagnostic codes)
usiProgress	USINT	Update progress in percent	0...100
sRespDesc	STRING	Description of the update progress	

Diagnostic codes

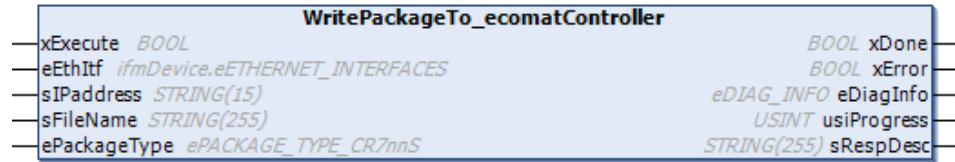
• STAT_INACTIVE	Status: FB / function is inactive
• STAT_BUSY	Status: The FB / function is being executed
• STAT_DONE	Status: FB / function successfully processed; there are valid results on the outputs.
• STAT_ABORT	Status: Execution of the FB / the function is cancelled
• STAT_EVNT_OPEN_CHAN_ETH	Status: SDK-API event <code>r3603_OpenChannel</code> is opened.
• STAT_EVNT_OPEN_CLOS_CHAN_ETH	Status: SDK-API event <code>r3603_CloseChannel</code> is opened.
• STAT_EVNT_OPEN_FIRMWARE_UPDATE	Status: SDK-API event <code>r3603_WriteFirmwareUpdate</code> is opened.
• STAT_EVNT_OPEN_DONE	Status: SDK-API event open.
• STAT_EVNT_REGI_OPEN_CHAN_ETH	Status: SDK-API event <code>r3603_OpenChannel</code> registered.
• STAT_EVNT_REGI_CLOS_CHAN_ETH	Status: SDK-API event <code>r3603_CloseChannel</code> registered.
• STAT_EVNT_REGI_FIRMWARE_UPDATE	Status: SDK-API event <code>r3603_WriteFirmwareUpdate</code> registered.
• STAT_EVNT_REGI_DONE	Status: SDK-API event registered.
• STAT_EVNT_UNREGI	Status: SDK-API event will be unregistered.
• STAT_EVNT_UNREGI_DONE	Status: SDK-API event unregistered.
• STAT_EVNT_CLOS	Status: SDK-API event is closed.
• STAT_EVNT_CLOS_DONE	Status: SDK-API event closed.
• ERR_EVNT_OPEN	Error: occurred when an SDK-API event was opened
• ERR_EVNT_REGI	Error: occurred when the call-back function was registered
• ERR_EVNT_UNREGI	Error: occurred when the call-back function was unregistered.
• ERR_EVNT_CLOS	Error: occurred when an event was closed
• ERR_INVALID_VALUE	Error: at least 1 invalid input parameter or invalid combination of input parameters; processing was cancelled

• ERR_DYNAMIC_MEMORY	Error: Memory reservation failed during creation of the internal FB instance.
• ERR_COMMUNICATION	Error: Communication error
• ERR_COMM_INTERFACE	Error: Communication interface error
• ERR_INVALID_TARGET_ID	Error: Invalid target ID (e.g. CAN ID, IP address etc.)
• ERR_INVALID_DEVICE	Error: Wrong or unknown device
• ERR_COMPATIBILITY	Error: Incompatible version of the hardware, file, runtime system, software or application
• ERR_INVALID_AUTH	Error: Invalid or missing authentication
• ERR_CMND_FAILED	Error: Execution of the command has failed.
• ERR_WRITE	Error: Error while writing a segment or package to the device.
• ERR_UNKNOWN	Error: Unknown error
• ERR_INTERNAL	Error: Internal system error; contact the ifm Service Center!

7.1.2.15 WritePackageTo_ecomatController

Function block type: Function block (FB)
Behaviour model: EXECUTE (→ Behaviour model EXECUTE □ 20)
Library: ifmSDKutil_ecomatDisplay

Symbol in CODESYS:



Description

ATTENTION

Aborting the write process to the controller

- ▷ Damage to the device possible
- ▶ Do not cancel the execution of the FB during the writing process.
- ▶ Set the value of the FB input `xExecute` to TRUE during the entire firmware update process.
- ▶ Only end the execution of the FB (`xExecute = FALSE`) when the FB outputs are `xDone = TRUE` or `xError = TRUE` .

The FB enables the user to write selected software from a zip software package file to an ecomatController.

- Data transmission takes a certain amount of time.
- ▶ While writing to the controller, check the status messages at the FB outputs `usiProgress` and `sRespDesc` .
- The type of software package to be written to the device is selected via the FB input `ePackageType` .
- An error will be returned if the selected segment is not available in the specified software package file.



So that the FB can access the target device:

- ▶ Before executing the FB: Log in using the FB `Authentication` with the required user rights (user level) and the corresponding password on the target device.
 Further information on the required user rights: → Programming manual of the target device

Input parameters

Name	Data type	Description	Value range
xExecute	BOOL	Control of the FB	• FALSE ⇒ TRUE: Execute the FB once. The values at the inputs will be adopted. Execute the FB (<code>xExecute = TRUE</code>) until the function block execution is successfully terminated (<code>xDone = TRUE</code>). If <code>xDone = TRUE</code>, then reset the input <code>xExecute</code> to FALSE. • TRUE ⇒ FALSE: Finish the FB processing.
eEthIf	ifmDevice.eETHERNET_INTERFACES	Ethernet interface of the dialogue unit, e.g. ecomatDisplay	Defined in: → ifmDevice.eETHERNET_INTERFACES

Name	Data type	Description	Value range
slAddress	STRING (15)	IP address of the target device, e.g. ecomatController	Default value: 192.168.82.247
sFileName	STRING	Directory path and name of the BIN file	e.g. \data\MyData.bin
ePackageType	ePACKAGE_TYPE_CR7nnS	List of software packages that can be written to the controller. The selection must match the software package type contained in the specified file.	Defined in: ePACKAGE_TYPE_CR7nnS (ENUM) (→ 87)

Output parameters

Name	Data type	Description	Value range
xDone	BOOL	Processing status	- FALSE: FB / FUN is inactive or being executed - TRUE: - FB / FUN was successfully executed - FB / FUN can be called again
xError	BOOL	Error indication	- FALSE: no error occurred or FB / FUN is still being executed - TRUE: - an error has occurred - The action could not be executed - Observe the diagnostic information
eDiagInfo	eDIAG_INFO	Diagnostic information	→ List below (diagnostic codes)
usiProgress	USINT	Update progress in percent	0...100
sRespDesc	STRING	Description of the update progress	

Diagnostic codes

• STAT_INACTIVE	Status: FB / function is inactive
• STAT_BUSY	Status: The FB / function is being executed
• STAT_DONE	Status: FB / function successfully processed; there are valid results on the outputs.
• STAT_ABORT	Status: Execution of the FB / the function is cancelled
• STAT_EVNT_OPEN_CHAN_ETH	Status: SDK-API event <code>r3603_OpenChannel</code> is opened.
• STAT_EVNT_OPEN_CLOS_CHAN_ETH	Status: SDK-API event <code>r3603_CloseChannel</code> is opened.
• STAT_EVNT_OPEN_WRIT_SEGM	Status: SDK-API event <code>WritDataSegmToFile</code> is opened.
• STAT_EVNT_OPEN_DONE	Status: SDK-API event open.
• STAT_EVNT_REGI_OPEN_CHAN_ETH	Status: SDK-API event <code>r3603_OpenChannel</code> registered.
• STAT_EVNT_REGI_CLOS_CHAN_ETH	Status: SDK-API event <code>r3603_CloseChannel</code> registered.
• STAT_EVNT_REGI_WRIT_SEGM	Status: SDK-API event <code>WritDataSegmToFile</code> registered.
• STAT_EVNT_REGI_DONE	Status: SDK-API event registered.
• STAT_EVNT_UNREGI	Status: SDK-API event will be unregistered.
• STAT_EVNT_UNREGI_DONE	Status: SDK-API event unregistered.
• STAT_EVNT_CLOS	Status: SDK-API event is closed.
• STAT_EVNT_CLOS_DONE	Status: SDK-API event closed.
• ERR_EVNT_OPEN	Error: occurred when an SDK-API event was opened
• ERR_EVNT_REGI	Error: occurred when the call-back function was registered
• ERR_EVNT_UNREGI	Error: occurred when the call-back function was unregistered.

• ERR_EVNT_CLOS	Error: occurred when an event was closed
• ERR_INVALID_VALUE	Error: at least 1 invalid input parameter or invalid combination of input parameters; processing was cancelled
• ERR_DYNAMIC_MEMORY	Error: Memory reservation failed during creation of the internal FB instance.
• ERR_COMMUNICATION	Error: Communication error
• ERR_COMM_INTERFACE	Error: Communication interface error
• ERR_INVALID_TARGET_ID	Error: Invalid target ID (e.g. CAN ID, IP address etc.)
• ERR_INVALID_DEVICE	Error: Wrong or unknown device
• ERR_COMPATIBILITY	Error: Incompatible version of the hardware, file, runtime system, software or application
• ERR_INVALID_AUTH	Error: Invalid or missing authentication
• ERR_CMND_FAILED	Error: Execution of the command has failed.
• ERR_WRITE	Error: Error while writing a segment or package to the device.
• ERR_UNKNOWN	Error: Unknown error
• ERR_INTERNAL	Error: Internal system error; contact the ifm Service Center!

7.1.2.16 ReadIOSettings_ecomatController

Function block type: Function block (FB)
Behaviour model: EXECUTE (→ Behaviour model EXECUTE □ 20)
Library: ifmSDKutil_ecomatDisplay

Symbol in CODESYS:



Description

The FB reads the settings of all IOs from the ecomatController.



So that the FB can access the target device:

- Before executing the FB: Log in using the FB Authentication with the required user rights (user level) and the corresponding password on the target device.
 Further information on the required user rights: → Programming manual of the target device

Input parameters

Name	Data type	Description	Value range
xExecute	BOOL	Control of the FB	- FALSE ⇒ TRUE: Execute the FB once. The values at the inputs will be adopted. Execute the FB (xExecute = TRUE) until the function block execution is successfully terminated (xDone = TRUE). If xDone = TRUE, then reset the input xExecute to FALSE. - TRUE ⇒ FALSE: Finish the FB processing.
eEthIf	ifmDevice.eETHERNET_INTERFACES	Ethernet interface of the dialogue unit, e.g. ecomatDisplay	Defined in: → ifmDevice.eETHERNET_INTERFACES
sIPaddress	STRING (15)	IP address of the target device, e.g. ecomatController	Default value: 192.168.82.247

Output parameters

Name	Data type	Description	Value range
xDone	BOOL	Processing status	- FALSE: FB / FUN is inactive or being executed - TRUE: - FB / FUN was successfully executed - FB / FUN can be called again
xError	BOOL	Error indication	- FALSE: no error occurred or FB / FUN is still being executed - TRUE: - an error has occurred - The action could not be executed - Observe the diagnostic information
eDiagInfo	eDIAG_INFO	Diagnostic information	→ List below (diagnostic codes)
stIOSettings	stIO_SETTINGS_CR7nnS	Settings of all available IOs	Defined in: stIO_SETTINGS_CR7nnS (→ □ 94)

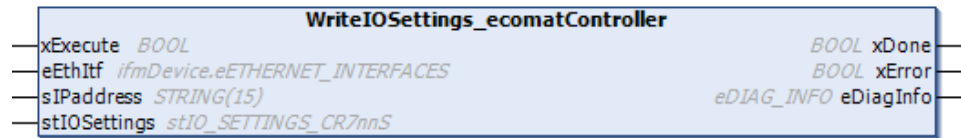
Diagnostic codes

• STAT_INACTIVE	Status: FB / function is inactive
• STAT_BUSY	Status: The FB / function is being executed
• STAT_DONE	Status: FB / function successfully processed; there are valid results on the outputs.
• STAT_ABORT	Status: Execution of the FB / the function is cancelled
• STAT_EVNT_OPEN_CHAN_ETH	Status: SDK-API event <code>r3603_OpenChannel</code> is opened.
• STAT_EVNT_OPEN_CLOS_CHAN_ETH	Status: SDK-API event <code>r3603_CloseChannel</code> is opened.
• STAT_EVNT_OPEN_READ_IO_SETTINGS	Status: SDK-API event <code>r3603_ReadIoSettings</code> is opened.
• STAT_EVNT_OPEN_DONE	Status: SDK-API event open.
• STAT_EVNT_REGI_OPEN_CHAN_ETH	Status: SDK-API event <code>r3603_OpenChannel</code> registered.
• STAT_EVNT_REGI_CLOS_CHAN_ETH	Status: SDK-API event <code>r3603_CloseChannel</code> registered.
• STAT_EVNT_REGI_READ_IO_SETTINGS	Status: SDK-API event <code>r3603_ReadIoSettings</code> registered.
• STAT_EVNT_REGI_DONE	Status: SDK-API event registered.
• STAT_EVNT_UNREGI	Status: SDK-API event will be unregistered.
• STAT_EVNT_UNREGI_DONE	Status: SDK-API event unregistered.
• STAT_EVNT_CLOS	Status: SDK-API event is closed.
• STAT_EVNT_CLOS_DONE	Status: SDK-API event closed.
• ERR_EVNT_OPEN	Error: occurred when an SDK-API event was opened
• ERR_EVNT_REGI	Error: occurred when the call-back function was registered
• ERR_EVNT_UNREGI	Error: occurred when the call-back function was unregistered.
• ERR_EVNT_CLOS	Error: occurred when an event was closed
• ERR_INVALID_VALUE	Error: at least 1 invalid input parameter or invalid combination of input parameters; processing was cancelled
• ERR_DYNAMIC_MEMORY	Error: Memory reservation failed during creation of the internal FB instance.
• ERR_COMMUNICATION	Error: Communication error
• ERR_COMM_INTERFACE	Error: Communication interface error
• ERR_INVALID_TARGET_ID	Error: Invalid target ID (e.g. CAN ID, IP address etc.)
• ERR_INVALID_DEVICE	Error: Wrong or unknown device
• ERR_COMPATIBILITY	Error: Incompatible version of the hardware, file, runtime system, software or application
• ERR_INVALID_AUTH	Error: Invalid or missing authentication
• ERR_CMND_FAILED	Error: Execution of the command has failed.
• ERR_WRITE	Error: Error while writing a segment or package to the device.
• ERR_UNKNOWN	Error: Unknown error
• ERR_INTERNAL	Error: Internal system error; contact the ifm Service Center!

7.1.2.17 WriteIOSettings_ecomatController

Function block type: Function block (FB)
Behaviour model: EXECUTE (→ Behaviour model EXECUTE □ 20)
Library: ifmSDKutil_ecomatDisplay

Symbol in CODESYS:



Description

ATTENTION

Aborting the write process to the controller

- ▷ Damage to the device possible
- ▶ Do not cancel the execution of the FB during the writing process.
- ▶ Set the value of the FB input `xExecute` to TRUE during the entire firmware update process.
- ▶ Only end the execution of the FB (`xExecute` = FALSE) when the FB outputs are `xDone` = TRUE or `xError` = TRUE .

The FB writes changed IO settings to the ecomatController.



So that the FB can access the target device:

- ▶ Before executing the FB: Log in using the FB `Authentication` with the required user rights (user level) and the corresponding password on the target device.
 Further information on the required user rights: → Programming manual of the target device

Input parameters

Name	Data type	Description	Value range
xExecute	BOOL	Control of the FB	- FALSE ⇒ TRUE: Execute the FB once. The values at the inputs will be adopted. Execute the FB (xExecute = TRUE) until the function block execution is successfully terminated (xDone = TRUE). If xDone = TRUE, then reset the input xExecute to FALSE. - TRUE ⇒ FALSE: Finish the FB processing.
eEthIf	ifmDevice.eETHERNET_INTERFACES	Ethernet interface of the dialogue unit, e.g. ecomatDisplay	Defined in: → ifmDevice.eETHERNET_INTERFACES
sIPaddress	STRING (15)	IP address of the target device, e.g. ecomatController	Default value: 192.168.82.247
stIOSettings	stIO_SETTINGS_CR7nnS	Settings of all available IOs	Defined in: stIO_SETTINGS_CR7nnS (→ □ 94)

Output parameters

Name	Data type	Description	Value range
xDone	BOOL	Processing status	- FALSE: FB / FUN is inactive or being executed - TRUE: - FB / FUN was successfully executed - FB / FUN can be called again
xError	BOOL	Error indication	- FALSE: no error occurred or FB / FUN is still being executed - TRUE: - an error has occurred - The action could not be executed - Observe the diagnostic information
eDiagInfo	eDIAG_INFO	Diagnostic information	→ List below (diagnostic codes)

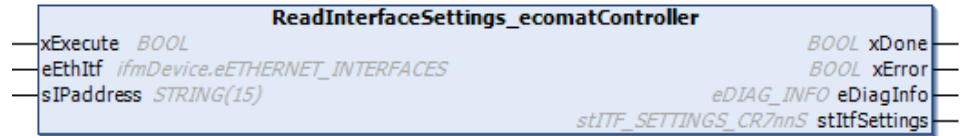
Diagnostic codes

• STAT_INACTIVE	Status: FB / function is inactive
• STAT_BUSY	Status: The FB / function is being executed
• STAT_DONE	Status: FB / function successfully processed; there are valid results on the outputs.
• STAT_ABORT	Status: Execution of the FB / the function is cancelled
• STAT_EVNT_OPEN_CHAN_ETH	Status: SDK-API event <code>r3603_OpenChannel</code> is opened.
• STAT_EVNT_OPEN_CLOS_CHAN_ETH	Status: SDK-API event <code>r3603_CloseChannel</code> is opened.
• STAT_EVNT_OPEN_WRIT_IO_SETTINGS	Status: SDK-API event <code>WriteIOSettings</code> is opened.
• STAT_EVNT_REGI_DONE	Status: SDK-API event registered.
• STAT_EVNT_UNREGI	Status: SDK-API event will be unregistered.
• STAT_EVNT_UNREGI_DONE	Status: SDK-API event unregistered.
• STAT_EVNT_CLOS	Status: SDK-API event is closed.
• STAT_EVNT_CLOS_DONE	Status: SDK-API event closed.
• ERR_EVNT_OPEN	Error: occurred when an SDK-API event was opened
• ERR_EVNT_REGI	Error: occurred when the call-back function was registered
• ERR_EVNT_UNREGI	Error: occurred when the call-back function was unregistered.
• ERR_EVNT_CLOS	Error: occurred when an event was closed
• ERR_INVALID_VALUE	Error: at least 1 invalid input parameter or invalid combination of input parameters; processing was cancelled
• ERR_DYNAMIC_MEMORY	Error: Memory reservation failed during creation of the internal FB instance.
• ERR_COMMUNICATION	Error: Communication error
• ERR_COMM_INTERFACE	Error: Communication interface error
• ERR_INVALID_TARGET_ID	Error: Invalid target ID (e.g. CAN ID, IP address etc.)
• ERR_INVALID_DEVICE	Error: Wrong or unknown device
• ERR_COMPATIBILITY	Error: Incompatible version of the hardware, file, runtime system, software or application
• ERR_INVALID_AUTH	Error: Invalid or missing authentication
• ERR_CMND_FAILED	Error: Execution of the command has failed.
• ERR_WRITE_IO_SETTING	Error: Error when writing the IO address
• ERR_UNKNOWN	Error: Unknown error
• ERR_INTERNAL	Error: Internal system error; contact the ifm Service Center!

7.1.2.18 ReadInterfaceSettings_ecomatController

Function block type: Function block (FB)
Behaviour model: EXECUTE (→ Behaviour model EXECUTE □ 20)
Library: ifmSDKutil_ecomatDisplay

Symbol in CODESYS:



Description

The FB returns a list of the interfaces incl. parameters of the ecomatController.



So that the FB can access the target device:

- Before executing the FB: Log in using the FB Authentication with the required user rights (user level) and the corresponding password on the target device.
 Further information on the required user rights: → Programming manual of the target device

Input parameters

Name	Data type	Description	Value range
xExecute	BOOL	Control of the FB	- FALSE ⇒ TRUE: Execute the FB once. The values at the inputs will be adopted. Execute the FB (xExecute = TRUE) until the function block execution is successfully terminated (xDone = TRUE). If xDone = TRUE, then reset the input xExecute to FALSE. - TRUE ⇒ FALSE: Finish the FB processing.
eEthIf	ifmDevice.eETHERNET_INTERFACES	Ethernet interface of the dialogue unit, e.g. ecomatDisplay	Defined in: → ifmDevice.eETHERNET_INTERFACES
sIPaddress	STRING (15)	IP address of the target device, e.g. ecomatController	Default value: 192.168.82.247

Output parameters

Name	Data type	Description	Value range
xDone	BOOL	Processing status	- FALSE: FB / FUN is inactive or being executed - TRUE: - FB / FUN was successfully executed - FB / FUN can be called again
xError	BOOL	Error indication	- FALSE: no error occurred or FB / FUN is still being executed - TRUE: - an error has occurred - The action could not be executed - Observe the diagnostic information
eDiagInfo	eDIAG_INFO	Diagnostic information	→ List below (diagnostic codes)
stIfSettings	stIfSettings_CR7nnS	List of interfaces of the ecomat-Controller incl. parameters	Defined in: stIfSettings_CR7nnS (→ □ 94)

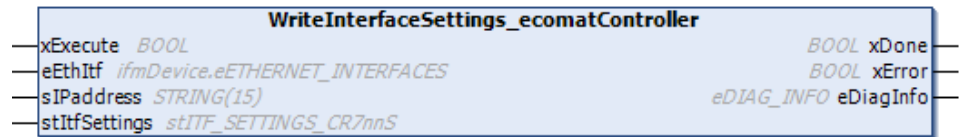
Diagnostic codes

• STAT_INACTIVE	Status: FB / function is inactive
• STAT_BUSY	Status: The FB / function is being executed
• STAT_DONE	Status: FB / function successfully processed; there are valid results on the outputs.
• STAT_ABORT	Status: Execution of the FB / the function is cancelled
• STAT_EVNT_OPEN_CHAN_ETH	Status: SDK-API event <code>r3603_OpenChannel</code> is opened.
• STAT_EVNT_OPEN_CLOS_CHAN_ETH	Status: SDK-API event <code>r3603_CloseChannel</code> is opened.
• STAT_EVNT_OPEN_READ_ITF_SETTINGS	Status: SDK-API event <code>r3603_ReadInterfaceSettings</code> is opened.
• STAT_EVNT_OPEN_DONE	Status: SDK-API event open.
• STAT_EVNT_REGI_OPEN_CHAN_ETH	Status: SDK-API event <code>r3603_OpenChannel</code> registered.
• STAT_EVNT_REGI_CLOS_CHAN_ETH	Status: SDK-API event <code>r3603_CloseChannel</code> registered.
• STAT_EVNT_REGI_READ_ITF_SETTINGS	Status: SDK-API event <code>r3603_ReadInterfaceSettings</code> registered.
• STAT_EVNT_REGI_DONE	Status: SDK-API event registered.
• STAT_EVNT_UNREGI	Status: SDK-API event will be unregistered.
• STAT_EVNT_UNREGI_DONE	Status: SDK-API event unregistered.
• STAT_EVNT_CLOS	Status: SDK-API event is closed.
• STAT_EVNT_CLOS_DONE	Status: SDK-API event closed.
• ERR_EVNT_OPEN	Error: occurred when an SDK-API event was opened
• ERR_EVNT_REGI	Error: occurred when the call-back function was registered
• ERR_EVNT_UNREGI	Error: occurred when the call-back function was unregistered.
• ERR_EVNT_CLOS	Error: occurred when an event was closed
• ERR_INVALID_VALUE	Error: at least 1 invalid input parameter or invalid combination of input parameters; processing was cancelled
• ERR_DYNAMIC_MEMORY	Error: Memory reservation failed during creation of the internal FB instance.
• ERR_COMMUNICATION	Error: Communication error
• ERR_COMM_INTERFACE	Error: Communication interface error
• ERR_INVALID_TARGET_ID	Error: Invalid target ID (e.g. CAN ID, IP address etc.)
• ERR_INVALID_DEVICE	Error: Wrong or unknown device
• ERR_COMPATIBILITY	Error: Incompatible version of the hardware, file, runtime system, software or application
• ERR_INVALID_AUTH	Error: Invalid or missing authentication
• ERR_CMND_FAILED	Error: Execution of the command has failed.
• ERR_READ	Error: Error reading a segment or package from the device.
• ERR_UNKNOWN	Error: Unknown error
• ERR_INTERNAL	Error: Internal system error; contact the ifm Service Center!

7.1.2.19 WriteInterfaceSettings_ecomatController

Function block type: Function block (FB)
Behaviour model: EXECUTE (→ Behaviour model EXECUTE □ 20)
Library: ifmSDKutil_ecomatDisplay

Symbol in CODESYS:



Description

The FB writes the interface parameters set at the FB input `stIfSettings` to the `ecomatController`.



So that the FB can access the target device:

- Before executing the FB: Log in using the FB `Authentication` with the required user rights (user level) and the corresponding password on the target device.
 Further information on the required user rights: → Programming manual of the target device

Input parameters

Name	Data type	Description	Value range
xExecute	BOOL	Control of the FB	- FALSE ⇒ TRUE: Execute the FB once. The values at the inputs will be adopted. Execute the FB (xExecute = TRUE) until the function block execution is successfully terminated (xDone = TRUE). If xDone = TRUE, then reset the input xExecute to FALSE. - TRUE ⇒ FALSE: Finish the FB processing.
eEthIf	ifmDevice.eETHERNET_INTERFACES	Ethernet interface of the dialogue unit, e.g. <code>ecomatDisplay</code>	Defined in: → <code>ifmDevice.eETHERNET_INTERFACES</code>
sIPaddress	STRING (15)	IP address of the target device, e.g. <code>ecomatController</code>	Default value: 192.168.82.247
stIfSettings	stITF_SETTINGS_CR7nnS	List of interfaces of the <code>ecomatController</code> incl. parameters	Defined in: <code>stITF_SETTINGS_CR7nnS</code> (→ □ 94)

Output parameters

Name	Data type	Description	Value range
xDone	BOOL	Processing status	- FALSE: FB / FUN is inactive or being executed - TRUE: - FB / FUN was successfully executed - FB / FUN can be called again
xError	BOOL	Error indication	- FALSE: no error occurred or FB / FUN is still being executed - TRUE: - an error has occurred - The action could not be executed - Observe the diagnostic information
eDiagInfo	eDIAG_INFO	Diagnostic information	→ List below (diagnostic codes)

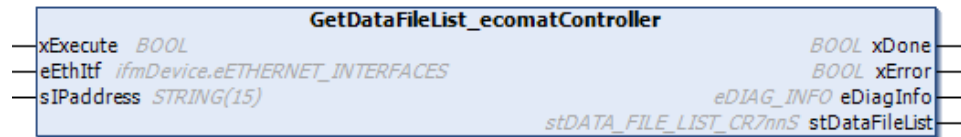
Diagnostic codes

• STAT_INACTIVE	Status: FB / function is inactive
• STAT_BUSY	Status: The FB / function is being executed
• STAT_DONE	Status: FB / function successfully processed; there are valid results on the outputs.
• STAT_ABORT	Status: Execution of the FB / the function is cancelled
• STAT_EVNT_OPEN_CHAN_ETH	Status: SDK-API event <code>r3603_OpenChannel</code> is opened.
• STAT_EVNT_OPEN_CLOS_CHAN_ETH	Status: SDK-API event <code>r3603_CloseChannel</code> is opened.
• STAT_EVNT_OPEN_WRIT_ITF_SETTINGS	Status: SDK-API event <code>r3603_WriteInterfaceSettings</code> is opened.
• STAT_EVNT_OPEN_DONE	Status: SDK-API event open.
• STAT_EVNT_REGI_OPEN_CHAN_ETH	Status: SDK-API event <code>r3603_OpenChannel</code> registered.
• STAT_EVNT_REGI_CLOS_CHAN_ETH	Status: SDK-API event <code>r3603_CloseChannel</code> registered.
• STAT_EVNT_REGI_WRIT_ITF_SETTINGS	Status: SDK-API event <code>r3603_WriteInterfaceSettings</code> registered.
• STAT_EVNT_REGI_DONE	Status: SDK-API event registered.
• STAT_EVNT_UNREGI	Status: SDK-API event will be unregistered.
• STAT_EVNT_UNREGI_DONE	Status: SDK-API event unregistered.
• STAT_EVNT_CLOS	Status: SDK-API event is closed.
• STAT_EVNT_CLOS_DONE	Status: SDK-API event closed.
• ERR_EVNT_OPEN	Error: occurred when an SDK-API event was opened
• ERR_EVNT_REGI	Error: occurred when the call-back function was registered
• ERR_EVNT_UNREGI	Error: occurred when the call-back function was unregistered.
• ERR_EVNT_CLOS	Error: occurred when an event was closed
• ERR_INVALID_VALUE	Error: at least 1 invalid input parameter or invalid combination of input parameters; processing was cancelled
• ERR_DYNAMIC_MEMORY	Error: Memory reservation failed during creation of the internal FB instance.
• ERR_COMMUNICATION	Error: Communication error
• ERR_COMM_INTERFACE	Error: Communication interface error
• ERR_INVALID_TARGET_ID	Error: Invalid target ID (e.g. CAN ID, IP address etc.)
• ERR_INVALID_DEVICE	Error: Wrong or unknown device
• ERR_COMPATIBILITY	Error: Incompatible version of the hardware, file, runtime system, software or application
• ERR_INVALID_AUTH	Error: Invalid or missing authentication
• ERR_CMND_FAILED	Error: Execution of the command has failed.
• ERR_WRITE_ITF_SETTINGS	Error: Error while writing the interface settings.
• ERR_UNKNOWN	Error: Unknown error
• ERR_INTERNAL	Error: Internal system error; contact the ifm Service Center!

7.1.2.20 GetDataFileList_ecomatController

Function block type: Function block (FB)
Behaviour model: EXECUTE (→ Behaviour model EXECUTE □ 20)
Library: ifmSDKutil_ecomatDisplay

Symbol in CODESYS:



Description

The FB returns the list of all files located in the /data folder of the ecomatController.



So that the FB can access the target device:

- Before executing the FB: Log in using the FB Authentication with the required user rights (user level) and the corresponding password on the target device.
 Further information on the required user rights: → Programming manual of the target device

Input parameters

Name	Data type	Description	Value range
xExecute	BOOL	Control of the FB	- FALSE ⇒ TRUE: Execute the FB once. The values at the inputs will be adopted. Execute the FB (xExecute = TRUE) until the function block execution is successfully terminated (xDone = TRUE). If xDone = TRUE, then reset the input xExecute to FALSE. - TRUE ⇒ FALSE: Finish the FB processing.
eEthIf	ifmDevice.eETHERNET_INTERFACES	Ethernet interface of the dialogue unit, e.g. ecomatDisplay	Defined in: → ifmDevice.eETHERNET_INTERFACES
sIPaddress	STRING (15)	IP address of the target device, e.g. ecomatController	Default value: 192.168.82.247

Output parameters

Name	Data type	Description	Value range
xDone	BOOL	Processing status	- FALSE: FB / FUN is inactive or being executed - TRUE: - FB / FUN was successfully executed - FB / FUN can be called again
xError	BOOL	Error indication	- FALSE: no error occurred or FB / FUN is still being executed - TRUE: - an error has occurred - The action could not be executed - Observe the diagnostic information
eDiagInfo	eDIAG_INFO	Diagnostic information	→ List below (diagnostic codes)
stDataFileList	stDATA_FILE_LIST_CR7nnS	List of files located in the folder /data of the ecomatController.	Defined in: stDATA_FILE_LIST_CR7nnS (→ □ 96)

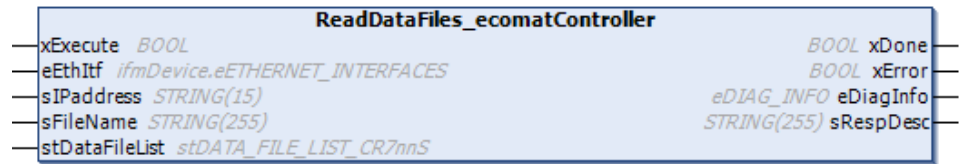
Diagnostic codes

• STAT_INACTIVE	Status: FB / function is inactive
• STAT_BUSY	Status: The FB / function is being executed
• STAT_DONE	Status: FB / function successfully processed; there are valid results on the outputs.
• STAT_ABORT	Status: Execution of the FB / the function is cancelled
• STAT_EVNT_OPEN_CHAN_ETH	Status: SDK-API event <code>r3603_OpenChannel</code> is opened.
• STAT_EVNT_OPEN_CLOS_CHAN_ETH	Status: SDK-API event <code>r3603_CloseChannel</code> is opened.
• STAT_EVNT_OPEN_GET_DATA_FILE_LIST	Status: SDK-API event <code>r3603_ReadFileList</code> is opened.
• STAT_EVNT_OPEN_DONE	Status: SDK-API event open.
• STAT_EVNT_REGI_OPEN_CHAN_ETH	Status: SDK-API event <code>r3603_OpenChannel</code> registered.
• STAT_EVNT_REGI_CLOS_CHAN_ETH	Status: SDK-API event <code>r3603_CloseChannel</code> registered.
• STAT_EVNT_REGI_GET_DATA_FILE_LIST	Status: SDK-API event <code>r3603_ReadFileList</code> registered.
• STAT_EVNT_REGI_DONE	Status: SDK-API event registered.
• STAT_EVNT_UNREGI	Status: SDK-API event will be unregistered.
• STAT_EVNT_UNREGI_DONE	Status: SDK-API event unregistered.
• STAT_EVNT_CLOS	Status: SDK-API event is closed.
• STAT_EVNT_CLOS_DONE	Status: SDK-API event closed.
• ERR_EVNT_OPEN	Error: occurred when an SDK-API event was opened
• ERR_EVNT_REGI	Error: occurred when the call-back function was registered
• ERR_EVNT_UNREGI	Error: occurred when the call-back function was unregistered.
• ERR_EVNT_CLOS	Error: occurred when an event was closed
• ERR_INVALID_VALUE	Error: at least 1 invalid input parameter or invalid combination of input parameters; processing was cancelled
• ERR_DYNAMIC_MEMORY	Error: Memory reservation failed during creation of the internal FB instance.
• ERR_COMMUNICATION	Error: Communication error
• ERR_COMM_INTERFACE	Error: Communication interface error
• ERR_INVALID_TARGET_ID	Error: Invalid target ID (e.g. CAN ID, IP address etc.)
• ERR_INVALID_DEVICE	Error: Wrong or unknown device
• ERR_COMPATIBILITY	Error: Incompatible version of the hardware, file, runtime system, software or application
• ERR_INVALID_AUTH	Error: Invalid or missing authentication
• ERR_CMND_FAILED	Error: Execution of the command has failed.
• ERR_READ	Error: Error reading a segment or package from the device.
• ERR_UNKNOWN	Error: Unknown error
• ERR_INTERNAL	Error: Internal system error; contact the ifm Service Center!

7.1.2.21 ReadDataFiles_ecomatController

Function block type: Function block (FB)
Behaviour model: EXECUTE (→ Behaviour model EXECUTE □ 20)
Library: ifmSDKutil_ecomatDisplay

Symbol in CODESYS:



Description

The FB reads one or more user files located in the `/data` folder of the ecomatController and saves them in a target path as a zip container file.

- The names of all user files that are to be read from the folder `/data` of the ecomatController are entered into the array at the FB input `stDataFileList`.
- Use the FB `GetDataFileList_ecomatController` to determine the list of files available in the folder `/data` of the ecomatController.
- The destination path and the name of the zip container file in which the read files are to be saved are entered at the FB input `sFileName`.



So that the FB can access the target device:

- Before executing the FB: Log in using the FB `Authentication` with the required user rights (user level) and the corresponding password on the target device.
 Further information on the required user rights: → Programming manual of the target device

Input parameters

Name	Data type	Description	Value range
xExecute	BOOL	Control of the FB	• FALSE ⇒ TRUE: Execute the FB once. The values at the inputs will be adopted. Execute the FB (xExecute = TRUE) until the function block execution is successfully terminated (xDone = TRUE). If xDone = TRUE, then reset the input xExecute to FALSE. • TRUE ⇒ FALSE: Finish the FB processing.
eEthIf	ifmDevice.eETHERNET_INTERFACES	Ethernet interface of the dialogue unit, e.g. ecomatDisplay	Defined in: → ifmDevice.eETHERNET_INTERFACES
sIPAddress	STRING (15)	IP address of the target device, e.g. ecomatController	Default value: 192.168.82.247
sFileName	STRING	Directory path and name of the BIN file	e.g. \data\MyData.bin
stPackageLst	stPACKAGE_LIST_CR7nnS	List of software packages that are read by the device. If no selection is made, no read operation is triggered.	Defined in: stPACKAGE_LIST_CR7nnS (→ □ 97)

Output parameters

Name	Data type	Description	Value range
xDone	BOOL	Processing status	- FALSE: FB / FUN is inactive or being executed - TRUE: - FB / FUN was successfully executed - FB / FUN can be called again
xError	BOOL	Error indication	- FALSE: no error occurred or FB / FUN is still being executed - TRUE: - an error has occurred - The action could not be executed - Observe the diagnostic information
eDiagInfo	eDIAG_INFO	Diagnostic information	→ List below (diagnostic codes)
stReadSta	stSEGM_READ_STATUS	Status of the reading process	Defined in: stSEGM_READ_STATUS (→ 91)

Diagnostic codes

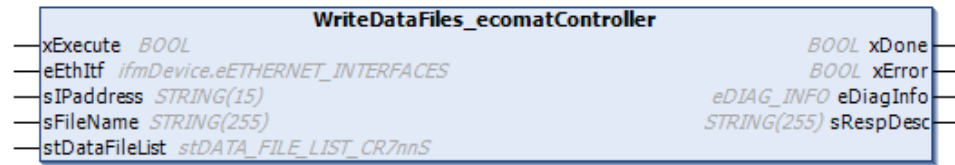
• STAT_INACTIVE	Status: FB / function is inactive
• STAT_BUSY	Status: The FB / function is being executed
• STAT_DONE	Status: FB / function successfully processed; there are valid results on the outputs.
• STAT_ABORT	Status: Execution of the FB / the function is cancelled
• STAT_EVNT_OPEN_CHAN_ETH	Status: SDK-API event <code>r3603_OpenChannel</code> is opened.
• STAT_EVNT_OPEN_CLOS_CHAN_ETH	Status: SDK-API event <code>r3603_CloseChannel</code> is opened.
• STAT_EVNT_OPEN_READ_DATA_FILES	Status: SDK-API event <code>r3603_ReadDataFiles</code> is opened.
• STAT_EVNT_OPEN_DONE	Status: SDK-API event open.
• STAT_EVNT_REGI_OPEN_CHAN_ETH	Status: SDK-API event <code>r3603_OpenChannel</code> registered.
• STAT_EVNT_REGI_CLOS_CHAN_ETH	Status: SDK-API event <code>r3603_CloseChannel</code> registered.
• STAT_EVNT_REGI_READ_DATA_FILES	Status: SDK-API event <code>r3603_ReadDataFiles</code> registered.
• STAT_EVNT_REGI_DONE	Status: SDK-API event registered.
• STAT_EVNT_UNREGI	Status: SDK-API event will be unregistered.
• STAT_EVNT_UNREGI_DONE	Status: SDK-API event unregistered.
• STAT_EVNT_CLOS	Status: SDK-API event is closed.
• STAT_EVNT_CLOS_DONE	Status: SDK-API event closed.
• ERR_EVNT_OPEN	Error: occurred when an SDK-API event was opened
• ERR_EVNT_REGI	Error: occurred when the call-back function was registered
• ERR_EVNT_UNREGI	Error: occurred when the call-back function was unregistered.
• ERR_EVNT_CLOS	Error: occurred when an event was closed
• ERR_INVALID_VALUE	Error: at least 1 invalid input parameter or invalid combination of input parameters; processing was cancelled
• ERR_DYNAMIC_MEMORY	Error: Memory reservation failed during creation of the internal FB instance.
• ERR_COMMUNICATION	Error: Communication error
• ERR_COMM_INTERFACE	Error: Communication interface error
• ERR_INVALID_TARGET_ID	Error: Invalid target ID (e.g. CAN ID, IP address etc.)
• ERR_INVALID_DEVICE	Error: Wrong or unknown device

• ERR_COMPATIBILITY	Error: Incompatible version of the hardware, file, runtime system, software or application
• ERR_INVALID_AUTH	Error: Invalid or missing authentication
• ERR_CMND_FAILED	Error: Execution of the command has failed.
• ERR_READ	Error: Error reading a segment or package from the device.
• ERR_UNKNOWN	Error: Unknown error
• ERR_INTERNAL	Error: Internal system error; contact the ifm Service Center!

7.1.2.22 WriteDataFiles_ecomatController

Function block type: Function block (FB)
Behaviour model: EXECUTE (→ Behaviour model EXECUTE □ 20)
Library: ifmSDKutil_ecomatDisplay

Symbol in CODESYS:



Description

The FB writes one or more user files from a zip container file (source) to the folder `/data` of the ecomatController (destination).

- The names of all user files that are to be written to the `/data` folder of the ecomatController are entered into the array at the FB input `stDataFileList`.
- The path and name of the zip container file is entered at the FB input `sFileName`.
- The maximum possible size of the zip container file is 768 kB
- No hierarchical folder structure is allowed within the zip container file.
- All files must be in the root directory of the zip container file.
- All file types are allowed.
- Filename:
 - Permitted characters: a-z, A-Z, 0-9, +, _
 - Other characters and special characters are not allowed.
- File name length: Maximum 17 characters including file extension

ATTENTION

Exceeding the available memory

- ▷ Damage to the file system on the device possible
- ▶ Check available memory on the ecomatController before loading user files on it.
- ▶ Do not exceed the maximum possible size of the zip container file of 768 kB to avoid data corruption.
- ▶ Refer to the ecomatController programming manual for information on the available memory for user files.



So that the FB can access the target device:

- ▶ Before executing the FB: Log in using the FB `Authentication` with the required user rights (user level) and the corresponding password on the target device.
 Further information on the required user rights: → Programming manual of the target device

Input parameters

Name	Data type	Description	Value range
xDone	BOOL	Processing status	• FALSE: FB / FUN is inactive or being executed • TRUE: – FB / FUN was successfully executed – FB / FUN can be called again

Name	Data type	Description	Value range
xError	BOOL	Error indication	- FALSE: no error occurred or FB / FUN is still being executed - TRUE: - an error has occurred - The action could not be executed - Observe the diagnostic information
eDiagInfo	eDIAG_INFO	Diagnostic information	→ List below (diagnostic codes)
sRespDesc	STRING	Description of the update progress	

Output parameters

Name	Data type	Description	Value range
xDone	BOOL	Processing status	- FALSE: FB / FUN is inactive or being executed - TRUE: - FB / FUN was successfully executed - FB / FUN can be called again
xError	BOOL	Error indication	- FALSE: no error occurred or FB / FUN is still being executed - TRUE: - an error has occurred - The action could not be executed - Observe the diagnostic information
eDiagInfo	eDIAG_INFO	Diagnostic information	→ List below (diagnostic codes)
stReadSta	stSEGM_READ_STATUS	Status of the reading process	Defined in: stSEGM_READ_STATUS (→ 91)

Diagnostic codes

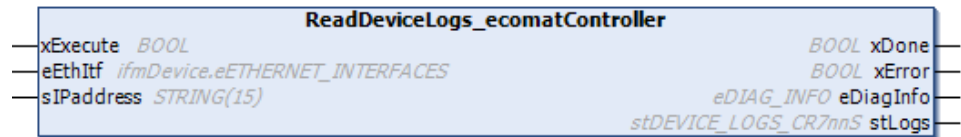
• STAT_INACTIVE	Status: FB / function is inactive
• STAT_BUSY	Status: The FB / function is being executed
• STAT_DONE	Status: FB / function successfully processed; there are valid results on the outputs.
• STAT_ABORT	Status: Execution of the FB / the function is cancelled
• STAT_EVNT_OPEN_CHAN_ETH	Status: SDK-API event <code>r3603_OpenChannel</code> is opened.
• STAT_EVNT_OPEN_CLOS_CHAN_ETH	Status: SDK-API event <code>r3603_CloseChannel</code> is opened.
• STAT_EVNT_OPEN_WRIT_DATA_FILES	Status: SDK-API event <code>r3603_WriteDataFiles</code> is opened.
• STAT_EVNT_OPEN_DONE	Status: SDK-API event open.
• STAT_EVNT_REGI_OPEN_CHAN_ETH	Status: SDK-API event <code>r3603_OpenChannel</code> registered.
• STAT_EVNT_REGI_CLOS_CHAN_ETH	Status: SDK-API event <code>r3603_CloseChannel</code> registered.
• STAT_EVNT_REGI_WRIT_DATA_FILES	Status: SDK-API event <code>r3603_WriteDataFiles</code> registered.
• STAT_EVNT_REGI_DONE	Status: SDK-API event registered.
• STAT_EVNT_UNREGI	Status: SDK-API event will be unregistered.
• STAT_EVNT_UNREGI_DONE	Status: SDK-API event unregistered.
• STAT_EVNT_CLOS	Status: SDK-API event is closed.
• STAT_EVNT_CLOS_DONE	Status: SDK-API event closed.
• ERR_EVNT_OPEN	Error: occurred when an SDK-API event was opened
• ERR_EVNT_REGI	Error: occurred when the call-back function was registered

• ERR_EVNT_UNREGI	Error: occurred when the call-back function was unregistered.
• ERR_EVNT_CLOS	Error: occurred when an event was closed
• ERR_INVALID_VALUE	Error: at least 1 invalid input parameter or invalid combination of input parameters; processing was cancelled
• ERR_DYNAMIC_MEMORY	Error: Memory reservation failed during creation of the internal FB instance.
• ERR_COMMUNICATION	Error: Communication error
• ERR_COMM_INTERFACE	Error: Communication interface error
• ERR_INVALID_TARGET_ID	Error: Invalid target ID (e.g. CAN ID, IP address etc.)
• ERR_INVALID_DEVICE	Error: Wrong or unknown device
• ERR_COMPATIBILITY	Error: Incompatible version of the hardware, file, runtime system, software or application
• ERR_INVALID_AUTH	Error: Invalid or missing authentication
• ERR_CMND_FAILED	Error: Execution of the command has failed.
• ERR_WRITE	Error: Error while writing a segment or package to the device.
• ERR_UNKNOWN	Error: Unknown error
• ERR_INTERNAL	Error: Internal system error; contact the ifm Service Center!

7.1.2.23 ReadDeviceLogs_ecomatController

Function block type: Function block (FB)
Behaviour model: EXECUTE (→ Behaviour model EXECUTE □ 20)
Library: ifmSDKutil_ecomatDisplay

Symbol in CODESYS:



Description

The FB reads the event log with logged errors, information and maintenance activities from the device.

- The event log is output as a list of logging events at the FB output `stLogs`.
- Order: The event with the most recent timestamp will be provided first.
- Maximum number of events: The event list may contain a maximum of 2001 entries.



So that the FB can access the target device:

- Before executing the FB: Log in using the FB `Authentication` with the required user rights (user level) and the corresponding password on the target device.
 Further information on the required user rights: → Programming manual of the target device

Input parameters

Name	Data type	Description	Value range
xExecute	BOOL	Control of the FB	• FALSE ⇒ TRUE: Execute the FB once. The values at the inputs will be adopted. Execute the FB (xExecute = TRUE) until the function block execution is successfully terminated (xDone = TRUE). If xDone = TRUE, then reset the input xExecute to FALSE. • TRUE ⇒ FALSE: Finish the FB processing.
eEthItf	ifmDevice.eETHERNET_INTERFACES	Ethernet interface of the dialogue unit, e.g. ecomatDisplay	Defined in: → ifmDevice.eETHERNET_INTERFACES
sIPaddress	STRING (15)	IP address of the target device, e.g. ecomatController	Default value: 192.168.82.247

Output parameters

Name	Data type	Description	Value range
xDone	BOOL	Processing status	• FALSE: FB / FUN is inactive or being executed • TRUE: – FB / FUN was successfully executed – FB / FUN can be called again
xError	BOOL	Error indication	• FALSE: no error occurred or FB / FUN is still being executed • TRUE: – an error has occurred – The action could not be executed – Observe the diagnostic information

Name	Data type	Description	Value range
eDiagInfo	eDIAG_INF0	Diagnostic information	→ List below (diagnostic codes)
stLogs	stDEVICE_LOGS_CR7nnS	Event log of the ecomatControl-ler	Defined in: stDEVICE_LOGS_CR7nnS (→ 96)

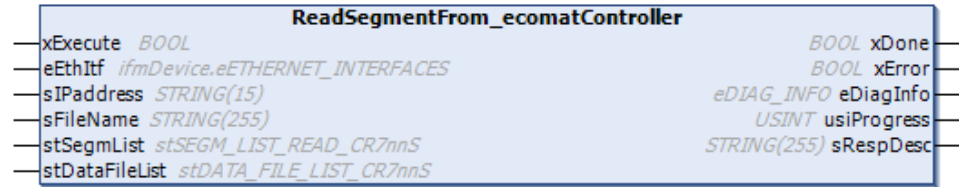
Diagnostic codes

- | | |
|---------------------------------|--|
| • STAT_INACTIVE | Status: FB / function is inactive |
| • STAT_BUSY | Status: The FB / function is being executed |
| • STAT_DONE | Status: FB / function successfully processed; there are valid results on the outputs. |
| • STAT_ABORT | Status: Execution of the FB / the function is cancelled |
| • STAT_EVTN_OPEN_CHAN_ETH | Status: SDK-API event <code>r3603_OpenChannel</code> is opened. |
| • STAT_EVTN_OPEN_CLOS_CHAN_ETH | Status: SDK-API event <code>r3603_CloseChannel</code> is opened. |
| • STAT_EVTN_OPEN_READ_DEVI_LOGS | Status: SDK-API event <code>r3603_ReadLogging</code> is opened. |
| • STAT_EVTN_OPEN_DONE | Status: SDK-API event open. |
| • STAT_EVTN_REGI_OPEN_CHAN_ETH | Status: SDK-API event <code>r3603_OpenChannel</code> registered. |
| • STAT_EVTN_REGI_CLOS_CHAN_ETH | Status: SDK-API event <code>r3603_CloseChannel</code> registered. |
| • STAT_EVTN_REGI_READ_DEVI_LOGS | Status: SDK-API event <code>3603_ReadLogging</code> registered. |
| • STAT_EVTN_REGI_DONE | Status: SDK-API event registered. |
| • STAT_EVTN_UNREGI | Status: SDK-API event will be unregistered. |
| • STAT_EVTN_UNREGI_DONE | Status: SDK-API event unregistered. |
| • ERR_EVTN_OPEN | Error: occurred when an SDK-API event was opened |
| • ERR_EVTN_REGI | Error: occurred when the call-back function was registered |
| • ERR_EVTN_UNREGI | Error: occurred when the call-back function was unregistered. |
| • ERR_EVTN_CLOS | Error: occurred when an event was closed |
| • ERR_INVALID_VALUE | Error: at least 1 invalid input parameter or invalid combination of input parameters; processing was cancelled |
| • ERR_DYNAMIC_MEMORY | Error: Memory reservation failed during creation of the internal FB instance. |
| • ERR_COMMUNICATION | Error: Communication error |
| • ERR_COMM_INTERFACE | Error: Communication interface error |
| • ERR_INVALID_TARGET_ID | Error: Invalid target ID (e.g. CAN ID, IP address etc.) |
| • ERR_INVALID_DEVICE | Error: Wrong or unknown device |
| • ERR_COMPATIBILITY | Error: Incompatible version of the hardware, file, runtime system, software or application |
| • ERR_INVALID_AUTH | Error: Invalid or missing authentication |
| • ERR_CMND_FAILED | Error: Execution of the command has failed. |
| • ERR_READ | Error: Error reading a segment or package from the device. |
| • ERR_UNKNOWN | Error: Unknown error |
| • ERR_INTERNAL | Error: Internal system error; contact the ifm Service Center! |

7.1.2.24 ReadSegmentFrom_ecomatController

Function block type: Function block (FB)
Behaviour model: EXECUTE (→ Behaviour model EXECUTE □ 20)
Library: ifmSDKutil_ecomatDisplay

Symbol in CODESYS:



Description

The FB allows the user to read one or more software segments and user files from an ecomatController and save them in the specified zip file.



- ▶ Do not make any manual changes to the segment files.
The content is a valid set of CRC-checked files. All changes from outside make the files unusable for the ifm SDK API.



So that the FB can access the target device:

- ▶ Before executing the FB: Log in using the FB Authentication with the required user rights (user level) and the corresponding password on the target device.
Further information on the required user rights: → Programming manual of the target device

- This FB is only supported for ecomatController with firmware version 3.1 or higher.
- If a file already exists in the path with the same name, the content of the file will be overwritten.
- The software segment to be read out from the ecomatController can be selected via the FB input `stSegmList`.
- The data files to be read from the folder `/data` of the ecomatController can be set via the FB input `stDataFileList`.
- By default, the file `journal.jnl` is read from the folder `/data` of the ecomatController because the underlying SDK API must specify at least one file name from the folder `/data`.
- ▶ Use the FB `GetDataFileList_ecomatController` to determine the list of files that are available in the `/data` folder of the ecomatController.
- An error will be returned if the selected segment or user file is not available in the device.
- The output of the FB is a base64-encoded data array in a zip package file containing all software segments selected by the user.
- Data transmission takes a certain amount of time.
- ▶ While reading from the controller: check the status messages on the FB outputs `usiProgress` and `sRespDesc`.

Input parameters

Name	Data type	Description	Value range
xExecute	BOOL	Control of the FB	- FALSE ⇒ TRUE: Execute the FB once. The values at the inputs will be adopted. Execute the FB (xExecute = TRUE) until the function block execution is successfully terminated (xDone = TRUE). If xDone = TRUE, then reset the input xExecute to FALSE. - TRUE ⇒ FALSE: Finish the FB processing.
eEthIf	ifmDevice.eETHERNET_INTERFACES	Ethernet interface of the dialogue unit, e.g. ecomatDisplay	Defined in: → ifmDevice.eETHERNET_INTERFACES
sIPaddress	STRING (15)	IP address of the target device, e.g. ecomatController	Default value: 192.168.82.247
sFileName	STRING	Directory path and name of the BIN file	e.g. \data\MyData.bin
stSegmList	stSEGM_LIST	required memory segments	Defined in: stSEGM_LIST (→ 89)
stDataFileList	stDATA_FILE_LIST_CR7nnS	List of all files to be read from the /data folder of the ecomat-Controller. ► Indicate at least one file.	Defined in: stDATA_FILE_LIST_CR7nnS (→ 96)

Output parameters

Name	Data type	Description	Value range
xDone	BOOL	Processing status	- FALSE: FB / FUN is inactive or being executed - TRUE: - FB / FUN was successfully executed - FB / FUN can be called again
xError	BOOL	Error indication	- FALSE: no error occurred or FB / FUN is still being executed - TRUE: - an error has occurred - The action could not be executed - Observe the diagnostic information
eDiagInfo	eDIAG_INFO	Diagnostic information	→ List below (diagnostic codes)
usiProgress	USINT	Update progress in percent	0...100
sRespDesc	STRING	Description of the update progress	

Diagnostic codes

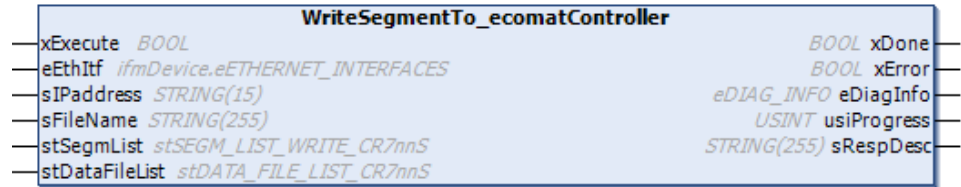
- | | |
|--------------------------------|---|
| • STAT_INACTIVE | Status: FB / function is inactive |
| • STAT_BUSY | Status: The FB / function is being executed |
| • STAT_DONE | Status: FB / function successfully processed; there are valid results on the outputs. |
| • STAT_ABORT | Status: Execution of the FB / the function is cancelled |
| • STAT_EVNT_OPEN_CHAN_ETH | Status: SDK-API event <code>r3603_OpenChannel</code> is opened. |
| • STAT_EVNT_OPEN_CLOS_CHAN_ETH | Status: SDK-API event <code>r3603_CloseChannel</code> is opened. |
| • STAT_EVNT_OPEN_READ_SEGM | Status: SDK-API event <code>ReadSoftSegmFromDevi</code> is opened. |
| • STAT_EVNT_OPEN_DONE | Status: SDK-API event open. |

• STAT_EVTN_REGI_OPEN_CHAN_ETH	Status: SDK-API event <code>r3603_OpenChannel</code> registered.
• STAT_EVTN_REGI_CLOS_CHAN_ETH	Status: SDK-API event <code>r3603_CloseChannel</code> registered.
• STAT_EVTN_REGI_READ_SEGM	Status: SDK-API event <code>ReadSoftSegmFromDevi</code> is registered.
• STAT_EVTN_REGI_DONE	Status: SDK-API event registered.
• STAT_EVTN_UNREGI	Status: SDK-API event will be unregistered.
• STAT_EVTN_UNREGI_DONE	Status: SDK-API event unregistered.
• STAT_EVTN_CLOS	Status: SDK-API event is closed.
• STAT_EVTN_CLOS_DONE	Status: SDK-API event closed.
• ERR_EVTN_OPEN	Error: occurred when an SDK-API event was opened
• ERR_EVTN_REGI	Error: occurred when the call-back function was registered
• ERR_EVTN_UNREGI	Error: occurred when the call-back function was unregistered.
• ERR_EVTN_CLOS	Error: occurred when an event was closed
• ERR_INVALID_VALUE	Error: at least 1 invalid input parameter or invalid combination of input parameters; processing was cancelled
• ERR_DYNAMIC_MEMORY	Error: Memory reservation failed during creation of the internal FB instance.
• ERR_COMMUNICATION	Error: Communication error
• ERR_COMM_INTERFACE	Error: Communication interface error
• ERR_INVALID_TARGET_ID	Error: Invalid target ID (e.g. CAN ID, IP address etc.)
• ERR_INVALID_DEVICE	Error: Wrong or unknown device
• ERR_COMPATIBILITY	Error: Incompatible version of the hardware, file, runtime system, software or application
• ERR_INVALID_AUTH	Error: Invalid or missing authentication
• ERR_CMND_FAILED	Error: Execution of the command has failed.
• ERR_READ	Error: Error reading a segment or package from the device.
• ERR_UNKNOWN	Error: Unknown error
• ERR_INTERNAL	Error: Internal system error; contact the ifm Service Center!

7.1.2.25 WriteSegmentTo_ecomatController

Function block type: Function block (FB)
Behaviour model: EXECUTE (→ Behaviour model EXECUTE □ 20)
Library: ifmSDKutil_ecomatDisplay

Symbol in CODESYS:



Description

The FB enables the user to write one or more software segments and user files from the specified zip file to the ecomatController.

ATTENTION

Aborting the write process to the controller

▷ Damage to the device possible

- ▶ Do not cancel the execution of the FB during the writing process.
- ▶ Set the value of the FB input `xExecute` to TRUE during the entire firmware update process.
- ▶ Only end the execution of the FB (`xExecute = FALSE`) when the FB outputs are `xDone = TRUE` or `xError = TRUE`.



So that the FB can access the target device:

- ▶ Before executing the FB: Log in using the FB `Authentication` with the required user rights (user level) and the corresponding password on the target device.
 Further information on the required user rights: → Programming manual of the target device

- This FB is only supported for ecomatController with firmware version 3.1 or higher.
- The software segment to be written to ecomatController can be selected via the FB input `stSegmList`.
- The selected segments will be written in the required order. The content of the specified segment files will not be validated.
- The names of all user files that are to be written to the /data folder of the ecomatController will be entered into the array at the FB input `stDataFileList`.
- The path and name of the zip container file will be entered at FB input `sFileName`.
- The maximum possible size of the zip container file is 768 kB
- No hierarchical folder structure is allowed within the zip container file.
- All files must be in the root directory of the zip container file.
- All file types are allowed.
- Filename:
 - Permitted characters: a-z, A-Z, 0-9, +, _
 - Other characters and special characters are not allowed.
- File name length: Maximum 17 characters including file extension

- By default, the `journal.jnl` file from the zip package is written to the '/data' folder of the `ecomatController`, as the underlying SDK API must specify at least one file name for the '/data' folder.
- An error will be returned if the selected segment or user file is not available in the zip package file.
- Data transmission takes a certain amount of time.
- ▶ Check the status messages at the FB output `sRespDesc` while writing to the controller.

ATTENTION

Exceeding the available memory

- ▷ Damage to the file system on the device possible
- ▶ Check available memory on the `ecomatController` before loading user files on it.
- ▶ Do not exceed the maximum possible size of the zip container file of 768 kB to avoid data corruption.
- ▶ Refer to the `ecomatController` programming manual for information on the available memory for user files.

- After updating the memory configuration, the safety application and the standard application will be deleted.
- ▶ To load a memory configuration file segment into the `ecomatController`, ensure that the file is named `memconf.ifm` and is located at `/Device/cfg` in the zip package file.
- There are often several variants of MemConfig files in the package file. These are named differently.
- ▶ Rename the file to be uploaded to the device to `memconf.ifm` and remove all other variants from the file.
- ▶ After loading applications onto the device, either perform a (manual) PowerOn reset or a reset with the reset command to transfer the data to the flash memory.



Manual loading of individual segments onto the `ecomatController` is at the user's own risk.

The user is responsible for the valid content of the corresponding segment files.

If you do not have the appropriate knowledge, it is strongly recommended to use the FB `WritePackageTo_ecomatController` instead.

Input parameters

Name	Data type	Description	Value range
<code>xExecute</code>	BOOL	Control of the FB	• FALSE ⇒ TRUE: Execute the FB once. The values at the inputs will be adopted. Execute the FB (<code>xExecute = TRUE</code>) until the function block execution is successfully terminated (<code>xDone = TRUE</code>). If <code>xDone = TRUE</code>, then reset the input <code>xExecute</code> to FALSE. • TRUE ⇒ FALSE: Finish the FB processing.
<code>eEthIf</code>	<code>ifmDevice.eETHERNET_INTERFACES</code>	Ethernet interface of the dialogue unit, e.g. <code>ecomatDisplay</code>	Defined in: → <code>ifmDevice.eETHERNET_INTERFACES</code>
<code>sIPAddress</code>	STRING (15)	IP address of the target device, e.g. <code>ecomatController</code>	Default value: 192.168.82.247
<code>sFileName</code>	STRING	Directory path and name of the BIN file	e.g. <code>\data\MyData.bin</code>

Name	Data type	Description	Value range
stSegmList	stSEGM_LIST	required memory segments	Defined in: stSEGM_LIST (→ 89)
stDataFileList	stDATA_FILE_LIST_CR7nnS	List of all files to be read from the /data folder of the ecomat-Controller. ► Indicate at least one file.	Defined in: stDATA_FILE_LIST_CR7nnS (→ 96)

Output parameters

Name	Data type	Description	Value range
xDone	BOOL	Processing status	- FALSE: FB / FUN is inactive or being executed - TRUE: - FB / FUN was successfully executed - FB / FUN can be called again
xError	BOOL	Error indication	- FALSE: no error occurred or FB / FUN is still being executed - TRUE: - an error has occurred - The action could not be executed - Observe the diagnostic information
eDiagInfo	eDIAG_INFO	Diagnostic information	→ List below (diagnostic codes)
usiProgress	USINT	Update progress in percent	0...100
sRespDesc	STRING	Description of the update progress	

Diagnostic codes

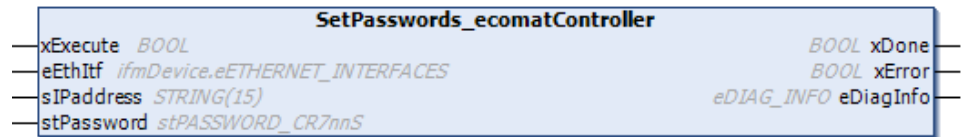
• STAT_INACTIVE	Status: FB / function is inactive
• STAT_BUSY	Status: The FB / function is being executed
• STAT_DONE	Status: FB / function successfully processed; there are valid results on the outputs.
• STAT_ABORT	Status: Execution of the FB / the function is cancelled
• STAT_EVNT_OPEN_CHAN_ETH	Status: SDK-API event <code>r3603_OpenChannel</code> is opened.
• STAT_EVNT_OPEN_CLOS_CHAN_ETH	Status: SDK-API event <code>r3603_CloseChannel</code> is opened.
• STAT_EVNT_OPEN_WRIT_SEGM	Status: SDK-API event <code>WritDataSegmToFile</code> is opened.
• STAT_EVNT_OPEN_DONE	Status: SDK-API event open.
• STAT_EVNT_REGI_OPEN_CHAN_ETH	Status: SDK-API event <code>r3603_OpenChannel</code> registered.
• STAT_EVNT_REGI_CLOS_CHAN_ETH	Status: SDK-API event <code>r3603_CloseChannel</code> registered.
• STAT_EVNT_REGI_WRIT_SEGM	Status: SDK-API event <code>WritDataSegmToFile</code> registered.
• STAT_EVNT_REGI_DONE	Status: SDK-API event registered.
• STAT_EVNT_UNREGI	Status: SDK-API event will be unregistered.
• STAT_EVNT_UNREGI_DONE	Status: SDK-API event unregistered.
• STAT_EVNT_CLOS	Status: SDK-API event is closed.
• STAT_EVNT_CLOS_DONE	Status: SDK-API event closed.
• ERR_EVNT_OPEN	Error: occurred when an SDK-API event was opened
• ERR_EVNT_REGI	Error: occurred when the call-back function was registered
• ERR_EVNT_UNREGI	Error: occurred when the call-back function was unregistered.
• ERR_EVNT_CLOS	Error: occurred when an event was closed

• ERR_INVALID_VALUE	Error: at least 1 invalid input parameter or invalid combination of input parameters; processing was cancelled
• ERR_DYNAMIC_MEMORY	Error: Memory reservation failed during creation of the internal FB instance.
• ERR_COMMUNICATION	Error: Communication error
• ERR_COMM_INTERFACE	Error: Communication interface error
• ERR_INVALID_TARGET_ID	Error: Invalid target ID (e.g. CAN ID, IP address etc.)
• ERR_INVALID_DEVICE	Error: Wrong or unknown device
• ERR_COMPATIBILITY	Error: Incompatible version of the hardware, file, runtime system, software or application
• ERR_INVALID_AUTH	Error: Invalid or missing authentication
• ERR_CMND_FAILED	Error: Execution of the command has failed.
• ERR_WRITE	Error: Error while writing a segment or package to the device.
• ERR_UNKNOWN	Error: Unknown error
• ERR_INTERNAL	Error: Internal system error; contact the ifm Service Center!

7.1.2.26 SetPasswords_ecomatController

Function block type: Function block (FB)
Behaviour model: EXECUTE (→ Behaviour model EXECUTE □ 20)
Library: ifmSDKutil_ecomatDisplay

Symbol in CODESYS:



Description

The FB writes a password file to the ecomatController.

- The password file must always contain all users that need to be assigned.
- In any case, the admin user and password must be set.
- The maximum password length can be 32 characters and can only support ASCII characters.
- After the password update, the user name and password become effective after the controller is switched on and off.
- This FB is only supported for ecomatController with firmware version 3.1 or higher.



So that the FB can access the target device:

- ▶ Before executing the FB: Log in using the FB **Authentication** with the required user rights (user level) and the corresponding password on the target device.
 Further information on the required user rights: → Programming manual of the target device

ATTENTION

Aborting the write process to the controller

- ▷ Damage to the device possible
- ▶ Do not cancel the execution of the FB during the writing process.
- ▶ Set the value of the FB input **xExecute** to TRUE during the entire firmware update process.
- ▶ Only end the execution of the FB (**xExecute** = FALSE) when the FB outputs are **xDone** = TRUE or **xError** = TRUE .

Input parameters

Name	Data type	Description	Value range
xExecute	BOOL	Control of the FB	- FALSE ⇒ TRUE: Execute the FB once. The values at the inputs will be adopted. Execute the FB (xExecute = TRUE) until the function block execution is successfully terminated (xDone = TRUE). If xDone = TRUE, then reset the input xExecute to FALSE. - TRUE ⇒ FALSE: Finish the FB processing.
eEthItf	ifmDevice.eETHERNET_INTERFACES	Ethernet interface of the dialogue unit, e.g. ecomatDisplay	Defined in: → ifmDevice.eETHERNET_INTERFACES
sIPaddress	STRING (15)	IP address of the target device, e.g. ecomatController	Default value: 192.168.82.247

Name	Data type	Description	Value range
ePassword	STRING	Optional: password for protected IEC applications	

Output parameters

Name	Data type	Description	Value range
xDone	BOOL	Processing status	- FALSE: FB / FUN is inactive or being executed - TRUE: - FB / FUN was successfully executed - FB / FUN can be called again
xError	BOOL	Error indication	- FALSE: no error occurred or FB / FUN is still being executed - TRUE: - an error has occurred - The action could not be executed - Observe the diagnostic information
eDiagInfo	eDIAG_INFO	Diagnostic information	→ List below (diagnostic codes)

Diagnostic codes

• STAT_INACTIVE	Status: FB / function is inactive
• STAT_BUSY	Status: The FB / function is being executed
• STAT_DONE	Status: FB / function successfully processed; there are valid results on the outputs.
• STAT_ABORT	Status: Execution of the FB / the function is cancelled
• STAT_EVNT_OPEN_CHAN_ETH	Status: SDK-API event <code>r3603_OpenChannel</code> is opened.
• STAT_EVNT_OPEN_CLOS_CHAN_ETH	Status: SDK-API event <code>r3603_CloseChannel</code> is opened.
• STAT_EVNT_OPEN_SET_PASSWORD	Status: SDK-API event <code>r3603_WritePassword</code> is opened.
• STAT_EVNT_OPEN_DONE	Status: SDK-API event open.
• STAT_EVNT_REGI_OPEN_CHAN_ETH	Status: SDK-API event <code>r3603_OpenChannel</code> registered.
• STAT_EVNT_REGI_CLOS_CHAN_ETH	Status: SDK-API event <code>r3603_CloseChannel</code> registered.
• STAT_EVNT_REGI_SET_PASSWORD	Status: SDK-API event <code>r3603_WritePassword</code> registered.
• STAT_EVNT_REGI_DONE	Status: SDK-API event registered.
• STAT_EVNT_UNREGI	Status: SDK-API event will be unregistered.
• STAT_EVNT_UNREGI_DONE	Status: SDK-API event unregistered.
• STAT_EVNT_CLOS	Status: SDK-API event is closed.
• STAT_EVNT_CLOS_DONE	Status: SDK-API event closed.
• ERR_EVNT_OPEN	Error: occurred when an SDK-API event was opened
• ERR_EVNT_REGI	Error: occurred when the call-back function was registered
• ERR_EVNT_UNREGI	Error: occurred when the call-back function was unregistered.
• ERR_EVNT_CLOS	Error: occurred when an event was closed
• ERR_INVALID_VALUE	Error: at least 1 invalid input parameter or invalid combination of input parameters; processing was cancelled
• ERR_DYNAMIC_MEMORY	Error: Memory reservation failed during creation of the internal FB instance.
• ERR_COMMUNICATION	Error: Communication error
• ERR_COMM_INTERFACE	Error: Communication interface error
• ERR_INVALID_TARGET_ID	Error: Invalid target ID (e.g. CAN ID, IP address etc.)

• ERR_INVALID_DEVICE	Error: Wrong or unknown device
• ERR_COMPATIBILITY	Error: Incompatible version of the hardware, file, runtime system, software or application
• ERR_INVALID_AUTH	Error: Invalid or missing authentication
• ERR_CMND_FAILED	Error: Execution of the command has failed.
• ERR_WRITE	Error: Error while writing a segment or package to the device.
• ERR_WRITE_PASSWORD	Error: Error while writing the password.
• ERR_UNKNOWN	Error: Unknown error
• ERR_INTERNAL	Error: Internal system error; contact the ifm Service Center!

7.1.3 ENUM

7.1.3.1 eCMD_LIST (ENUM)

List of device commands for R360 Basic, Classic / Extended Controllers.

Symbol	Description	Data type	Value
None	selection invalid	USINT	0
START_APP	Start IEC application		1
STOP_APP	Stop IEC application		2
RESET_WARM	Execute CODESYS reset (warm)		3
RESET_COLD	Execute CODESYS reset (cold)		4
RESET_ORIGIN	Execute CODESYS reset (origin)		5
REBOOT_BASIC_SYSTEM	Restart the device		6

7.1.3.2 eDEVICES (ENUM)

List of item numbers supported by this library.

ecomatControllers are supported in addition to this list.

Symbol	Description	Data type	Value
None	selection invalid	USINT	0
BASIC_SYSTEM	All devices of the BasicSystem family (includes CR0401, CR0403, CR0411, CR0451, CR0452, CR2530, CR2532)		1
CR0020	R360 System Controller CR0020 (16 bits)		2
CR0032	R360 System Controller CR0032 (32 bits)		3
CR0033	R360 System Controller CR0033 (32 bits)		4
CR0133	R360 System Controller CR0133 (32 bits)		5
CR0200	R360 System Controller CR0200 (16 bits)		6
CR0232	R360 System Controller CR0232 (32 bits)		7
CR0233	R360 System Controller CR0233 (32 bits)		8
CR0234	R360 System Controller CR0234 (32 bits)		9
CR0235	R360 System Controller CR0235 (32 bits)		10
CR0303	R360 System Controller CR0303 (16 bits)		11
CR0505	R360 System Controller CR0505 (16 bits)		12
CR2500	R360 System Controller CR2500 (16 bits)		13
CR2530	R360 System Controller CR2530 (32 bits)		14

Symbol	Description	Data type	Value
CR2532	R360 System Controller CR2532 (32 bits)	USINT	15
CR7021	R360 System Controller CR7021 (16 bits)		16
CR7032	R360 System Controller CR7032 (32 bits)		17
CR7121	R360 System Controller CR7121 (16 bits)		18
CR7132	R360 System Controller CR7232 (32 bits)		19
CR7201	R360 System Controller CR7201 (16 bits)		20
CR7506	R360 System Controller CR7506 (32 bits)		21

7.1.3.3 eSEGM_TYPE (ENUM)

List of software segments for R360 Basic, Classic / Extended Controllers.

Element	Description	Data type	Value
None	selection invalid	USINT	0
FIRMWARE	Firmware		1
IEC_APP	IEC application		2
IEC_APP_CFG	IEC application configuration		3
FILE_FLASH	File flash		4
USER_FLASH	User flash		5
DATA_FLASH	Data flash		6
RETA_DATA	Retain data / IEC user retain		7
MEMO_RETA	Memory retain		8
EEPROM	EEPROM		9

7.1.3.4 eCMD_LIST_CR7nnS (ENUM)

List of device commands for ecomatControllers.

Symbol	Description	Data type	Value
None	selection invalid	USINT	0
START_STANDARD_APP	Start the IEC standard application		1
STOP_STANDARD_APP	Stop the IEC standard application		2
STOP_SAFE_APP	Stop the IEC safety application		3
DELETE_STANDARD_APP	Delete the IEC standard application		4
DELETE_SAFE_APP	Start the IEC safety application		5
REBOOT	Reboot the device (power-on reset)		6

7.1.3.5 ePACKAGE_TYPE_CR7nnS (ENUM)

List of software packages for ecomatController.

Element	Description	Data type	Value
None	selection invalid	USINT	0
FIRMWARE	Firmware (OR runtime system) package, contains the segments: compat, boot, sissys, ifmOS		1
IEC_APP_STD	IEC standard application package, contains the segments: standard app, standard retain, standard memory bytes		2
IEC_APP_SAFE	IE safety application package, contains the segments: safe app, safe retain and safe memory bytes		3
IEC_APPS	IEC application package, contains the segments: safe app, standard app, memory settings package files (optional), iomapping (optional)		4
MEMORY_SETTINGS	Memory settings package, contains the segments: 'compat' and 'memconf'		5

7.1.3.6 eUSER_LEVEL (ENUM)

User level for the ecomatController.

Element	Description	Data type	Value
None	selection invalid	INT	0
Admin	→ ecomatController programming manual		1
S_ADMIN			2
S_SERVICE			3
N_ADMIN			4
N_SERVICE			5

7.1.4 STRUCT

7.1.4.1 stSYS_INFO

Defines available system information of the connected R360 Basic, Classic/ Extended Controllers.

Not all devices may provide all system information.

If a connected device does not provide a specific element of this structure, the value of the string is `-not available-`.

Element	Data type	Description	Possible values
sArticle	STRING	article number of the device	CR0401
sHardVers	STRING	Hardware version	
sSoftVers	STRING	Firmware version	
xBootLodrMode	BOOL	Status of the boot loader mode.	• TRUE: Boot loader mode active. • FALSE: Boot loader mode inactive. E.g. when the device is in runtime mode.
sAppName	STRING	Name of the current application (Not with the Basic System)	
sAppState	STRING	State of the application	• notavailable: No application loaded • notrunning: Application is loaded, but it does not run • running: Application running • unknown: The state of the application could not be determined. E.g. when the controller is in the boot loader mode.
udiAppCRC	UDINT	Application CRC	

7.1.4.2 stSEGM_LIST

List of software segments for R360 Basic, Classic / Extended Controllers.

Not all devices may provide all memory segments.

If the connected device does not provide at least one of the selected memory segments, an error occurs during the function (read or write).

Element	Data type	Description	Possible values
xFirmware	BOOL	Firmware segment	TRUE: Read firmware
xIECApp	BOOL	IEC application segment	TRUE: Read application
xIECAppCfg	BOOL	IEC application configuration segment Only for Basic System. Must also be selected if the IEC application segment is selected.	TRUE: Read IEC application configuration
xFileFlash	BOOL	File flash segment	TRUE: Read file flash
xUserFlash	BOOL	User file flash segment	TRUE: Read user file flash
xDataFlash	BOOL	Data flash segment Only for R360 system	TRUE: Read data flash
xRetaData	BOOL	Retain data segment	TRUE: Read retain data
xMemoReta	BOOL	Memory Retain segment Only for 32 bit R360 system	TRUE: Read memory retain

Element	Data type	Description	Possible values
xEEPROM	BOOL	EEPROM segment Only for R360 system	TRUE: Read EEPROM

7.1.4.3 stSEGM_NAME

Segment name with selection option.

Element	Data type	Description	Possible values
xSele	BOOL	Selection of the segment	- TRUE: selected - FALSE: not selected
sName	STRING(255)	Segment name	Depending on the container file types

7.1.4.4 stSEGM_TYPE_NAME

Segment type and segment name with selection option.

Element	Data type	Description	Possible values
xSele	BOOL	Selection of the segment	- TRUE: selected - FALSE: not selected
sType	STRING(20)	Segment data type	e.g.: fw, app, appconf, fileflash, userflash, etc.
sName	STRING(255)	Segment name	Depending on the container file types

7.1.4.5 stSEGM_LIST_WRITE

List of software segment types to be written to the controller, together with segment type and segment name. The selection to write to the device is made by the xSele Struct member.

Not all devices may provide all memory segments.

If the connected device does not provide at least one of the selected memory segments, an error occurs during the function (read or write).

Element	Data type	Description
stFirmware	ifmSDKUtil.stSEGM_NAME	Firmware segment
stIECApp		IEC application segment
stIECAppCfg		IEC application configuration segment. Only for Basic System. Must also be selected if the IEC application segment is selected.
stFileFlash		File flash segment
stUserFlash		User file flash segment
stDataFlash		Data flash segment Only for R360 system
stRetaData		Retain data segment.
stMemoReta		Memory retain segment Only for 32 bit R360 system
stEEPROM		EEPROM segment Only for R360 system

7.1.4.6 stLAST_EVNT_RESP

Latest response code received from the SDK API callback event.

Element	Data type	Description	Possible values
diErrorCode	DINT	Latest response code received from the SDK API callback event.	< 0: Error message = 0: Done > 0: In progress
sRespDesc	STRING(251)	Description according to the error code.	

7.1.4.7 stRD_WR_INFO_TYPE

Additional information when reading or writing software segments.

Element	Data type	Description	Possible values
xSele	BOOL	Segment selection	- FALSE: Segment selection not done - TRUE: Segment selection done
xRead	BOOL	Reading a segment	TRUE: Reading the segment from a file or from the device takes place
xWrit	BOOL	Writing a segment	TRUE: Writing the segment to a file or to the device takes place
usiProg	USINT	Reading progress or writing progress in %	0...100 %

7.1.4.8 stSEGM_READ_STATUS

Status of various operations when reading software segments from Basic, Classic / Extended controllers.

Element	Data type	Description
stFirmware	ifmSDKutil.stRD_WR_INFO_TYPE	Firmware segment status
stIECApp		IEC application segment status
stIECAppCfg		IEC application configuration segment status
stFileFlash		File flash segment status
stUserFlash		User flash segment status
stDataFlash		Data flash segment status
stRetaData		Retain data / IEC user retain segment status
stMemoReta		Memory retain segment status
stEEPROM		EEPROM segment status
xError	BOOL	- TRUE: The SDK API call has returned an error. - FALSE: The SDK API call did not return an error.

7.1.4.9 stSEGM_WRITE_STATUS

Status of various operations when writing software segments of Basic, Classic / Extended controllers.

Element	Data type	Description
stFirmware	ifmSDKutil.stRD_WR_INFO_TYPE	Firmware segment status
stIECApp		IEC application segment status
stIECAppCfg		IEC application configuration segment status
stFileFlash		File flash segment status
stUserFlash		Status User Flash Segment

Element	Data type	Description
stDataFlash	ifmSDKutil.stRD_WR_INFO_TYPE	Data flash segment status
stRetaData		Retain data / IEC user retain segment status
stMemoReta		Memory retain segment status
stEEPROM		EEPROM segment status
xError	BOOL	• TRUE: The SDK API call has returned an error. • FALSE: The SDK API call did not return an error.

7.1.4.10 stITF_SETTINGS

Interface parameters of the Basic and R360 controllers.

Element	Data type	Description
usiChannel	USINT	CAN channel number of the Basic/R360 controller on which the communication settings are to be changed. • 1 = CAN1 • 2 = CAN2
usiDownloadId	USINT	Node ID of the Basic/R360 controller on which the communication settings are to be changed.
eBaudrate	ifmDevice.CAN_BAUDRATE	CAN network baud rate of the Basic/R360 controller on which the communication settings are to be changed. Defined in: ifmDevice.CAN_BAUDRATE
xSameNetwork	BOOL	Note: Applies only to Basic type units: Indicates whether the interface settings are changed on the CAN channel of the target device to which the ecomatDisplay is connected. • FALSE = No • TRUE = Yes

7.1.4.11 stSymbolCfgInfo

Information on symbol configuration

Element	Data type	Description
sName	STRING	Name of the symbol file stored on the device.
udiSize	UDINT	The actual size of the symbol configuration
udiOCC_Size	UDINT	The size occupied by the symbol configuration on the controller.

7.1.4.12 stSOFTWARE_INFO_CR7nnS

System software information of the connected ecomatController.

Element	Data type	Description	Possible values
sSegmentType	STRING	Type of the software segment	
sSegmentName	STRING	Name of the software segment	
sCodeCRC32	STRING	CRC32 of the software segment	

Element	Data type	Description	Possible values
sHeadCRC32	STRING	CRC32 of the software segment header	
sVersion	STRING	Optional: Version of the software segment	
sBuildDate	STRING	Optional: Build date of the software segment	
xActive	BOOL	Indicates whether the software segment is active (only available for bootloader and ifm operating system)	• TRUE: active • FALSE: not active

7.1.4.13 stDEVICE_INFO_CR7nnS

Device information about the connected ecomatController.

Element	Data type	Description
sArticle	STRING	article number of the device
sArticleDesc	STRING	Part description
sHardwareVers	STRING	Hardware version
sHardwareRev	STRING	Optional: Hardware revision
sSerialNb	STRING	Serial number
sAssemblyDate	STRING	Optional: Production date

7.1.4.14 stNETWORK_INFO_CR7nnS

Network information of the connected ecomatController.

Element	Data type	Description
uiID	UINT	Network ID
sMAC	STRING	MAC address
sDNS	STRING	DNS

7.1.4.15 stSYS_INFO_CR7nnS

System information of the connected ecomatController.

Element	Data type	Description
aSoftwareInfo	ARRAY [0..31] OF ifmSDKutil.stSOFTWARE_INFO_CR7nnS	Software information
stDeviceInfo	ifmSDKutil.stDEVICE_INFO_CR7nnS	Device information
aNetworkInfo	ARRAY [0..8] OF ifmSDKutil.stNETWORK_INFO_CR7nnS	Network information

7.1.4.16 stREAD_INFO_TYPE_CR7nnS

Additional information when reading software segments from the ecomatController.

Element	Data type	Description	Possible values
xSele	BOOL	status: Segment or package selected	FALSE: Segment or package is selected TRUE: Segment or package is selected
xRead	BOOL	Reading status	FALSE: Reading of the segment or packet is inactive TRUE: The reading of the segment or the packet is running / done

Element	Data type	Description	Possible values
usiProg	USINT	Reading progress in %	0...100 %
sRespDesc	STRING(255)	Description of reading progress	

7.1.4.17 stIO_SETTINGS_CR7nnS

IO settings of the ecomatController.

Element	Data type	Description
stInfo	ifmSDKutil.stIO_INFO_CR7nnS	IO file information
usiTotElem	USINT(0..201)	Number of valid array elements in the astData variable of this struct.
astData	ifmSDKutil.aIO_DATA_CR7nnS	IO mapping data Each element in the array represents an IO file record.

7.1.4.18 stIO_INFO_CR7nnS

IO file information of the ecomatController.

Element	Data type	Description	Possible values
sName	STRING(64)	Name of the controller	- Format: CRnnnS-AllStandard (or other meaningful designation) - Length: 3 to 64 characters - Example: CR720S-AllStandard
sVersion	STRING	File version	- Format: Vxx.xx.xx.xxxx - Set-up: Major.Minor.Patch.Buildnumber - Major : 1-2 digits - Minor : 1-2 digits - Patch : 1-2 digits - Build number : 1-4 digits - Example: V1.0.1.1
sDate	STRING	Release date	Example: 2021.07.29 00:00:00

7.1.4.19 stIO_DATA_CR7nnS

IO mapping data of the ecomatController.

Element	Data type	Description	Possible values
sIOName	STRING	Symbolic name of the IO.	Examples: - IN0000 - VBB30 - TEMPERATUREBOARD0 - USERLED0
sIOType	STRING	Mapping type of the IO.	- standard: Standard application - safe: Safety application

7.1.4.20 stITF_SETTINGS_CR7nnS

List of interfaces and associated parameters of the ecomatController.

Element	Data type	Description
usiNumOfItf_CAN	USINT(0..8)	Number of valid array elements in the variable astItfList_CAN in this STRUCT

Element	Data type	Description
astItfList_CAN	ifmSDKutil.aITF_SETTINGS_CAN_CR7nnS	List of CAN interfaces and associated parameters
usiNumOfItf_ETH	USINT(0..8)	Number of valid array elements in the variable <code>astItfList_ETH</code> in this STRUCT
astItfList_ETH	ifmSDKutil.aITF_SETTINGS_ETH_CR7nnS	List of Ethernet interfaces and associated parameters
usiNumOfItf_RS232	USINT(0..8)	Number of valid array elements in the variable <code>astItfList_RS232</code> in this STRUCT
astItfList_RS232	ifmSDKutil.aITF_SETTINGS_RS232_CR7nnS	List of the RS232 interfaces and associated parameters

7.1.4.21 stITF_SETTINGS_CAN_CR7nnS

List of the CAN interfaces and the associated parameters of the ecomatController.

Element	Data type	Description	Possible values
usiItfId	USINT	Interface ID (zero-based)	
udiBaudrate	UDINT	Baud rate of the CAN interface	
uiNodeId	UINT	Node ID of the CAN interface	
xActive	BOOL	Status of the interface	• TRUE: active • FALSE: not active

7.1.4.22 stITF_SETTINGS_ETH_CR7nnS

List of the Ethernet interfaces and the associated parameters of the ecomatController.

Element	Data type	Description	Possible values
usiItfId	USINT	Interface ID (zero-based)	
sIPv4_Address	UDINT	IP address	
sIPv4_Netmask	UINT	Subnet mask	
sIPv4_Gateway	BOOL	Gateway address	
xActive	BOOL	Status of the interface	• TRUE: Active • FALSE: Inactive

7.1.4.23 stITF_SETTINGS_RS232_CR7nnS

List of the RS232 interfaces and the associated parameters of the ecomatController

Element	Data type	Description	Possible values
usiItfId	USINT	Interface ID (zero-based)	
udiBaudrate	UDINT	The baud rate of the interface	
usiBits	USINT	Number of data bits	
usiParity	USINT	Parity	• 0: No parity • 1: Even parity • 2: Odd parity
usiStop	USINT	Stop bit	• 0: No stop bit • 1: Stop bit
xActive	BOOL	Status of the interface	• TRUE: Active • FALSE: Disabled

7.1.4.24 stDATA_FILE_LIST_CR7nnS

List of user files in the ecomatController folder /data

Element	Data type	Description
uiTotElem	UINT(0..501)	Number of valid array elements in the variable <code>asFileName</code> in this Struct
asFileName	ARRAY [0..500] OF STRING	List of all file names with file extension and without file path

7.1.4.25 stDEVICE_LOGS_CR7nnS

Event log list of the ecomatController.

Element	Data type	Description
uiTotElem	UINT(0..2001)	Number of valid array elements of the variable <code>astInfo</code> in this Struct
astList	aDEVICE_LOG_INFO_CR7nnS	Event log list of the ecomatController

7.1.4.26 stDEVICE_LOG_INFO_CR7nnS

Log information of the ecomatController.

Element	Data type	Description
uliEventTime	ULINT	Event time, relative to the start of the device
sEventTimestamp	STRING	Event time in readable format
sEventType	STRING	Event type
sErrorCcode	STRING	Error code
sEventComponent	STRING	Event source
uiEventChannel	UINT	Event channel
udiEventValue	UDINT	Event value

7.1.4.27 stSEGM_LIST_READ_CR7nnS

List of software segments that are read from the ecomatController.

Element	Data type	Description
xSwInfo	BOOL	Software information
xAppInfo	BOOL	Application information
xDevInfo	BOOL	Device information
xComConf	BOOL	Communication interface configuration
xlOMapping	BOOL	IO mapping
xMemConf	BOOL	Memory configuration
xlECAApp_Standard	BOOL	Standard application
xlECAApp_StandardRetain	BOOL	Standard retain data
xlECAApp_StandardMemoryBytes	BOOL	Standard memory bytes
xlECAApp_Safe	BOOL	Safety application
xlECAApp_SafeRetain	BOOL	Safety retain data
xlECAApp_SafeMemoryBytes	BOOL	Safety memory bytes
xCompat	BOOL	Compatibility
xBootLoader	BOOL	Boot Loader
xSisSys	BOOL	System information service

Element	Data type	Description
xIfmOS	BOOL	Firmware
xPassword	BOOL	Password file
xLogging	BOOL	Logging Info
xUserFileList	BOOL	List of user files in the /data folder of the ecomatController. Available for devices with firmware V3.1 or higher. This variable is hidden and not visible to the user.

7.1.4.28 stSEGM_LIST_WRITE_CR7nnS

List of software segments that are written to the ecomatController.

Element	Data type	Description
xComConf	BOOL	Communication interface configuration
xIOMapping	BOOL	IO mapping
xMemConf	BOOL	Memory configuration
xIECApp_Standard	BOOL	Standard application
xIECApp_StandardRetain	BOOL	Standard retain data
xIECApp_StandardMemoryBytes	BOOL	Standard memory bytes
xIECApp_Safe	BOOL	Safety application
xIECApp_SafeRetain	BOOL	Safety retain data
xIECApp_SafeMemoryBytes	BOOL	Safety memory bytes
xCompat	BOOL	Compatibility
xPassword	BOOL	Password file
xUserFileList	BOOL	List of user files in the /data folder of the ecomatController. Available for devices with firmware V3.1 or higher. This variable is hidden and not visible to the user.

7.1.4.29 stPACKAGE_LIST_CR7nnS

List of applicable software package types for the ecomatController.

Element	Data type	Description
xFirmware	BOOL	Firmware (OR runtime system) package, contains these segments: compat, boot, sissys, ifmOS
xIECApp_Standard	BOOL	IEC standard application package, contains these segments: standard app, standard retain, standard memory bytes
xIECApp_Safe	BOOL	IEC safety application package, contains these segments: safe app, safe retain and safe memory bytes
xIECApps	BOOL	IEC application package, contains these segments: safe app, standard app, memory settings package files (optional) and iomapping (optional)
xMemorySettings	BOOL	Memory settings package, contains these segments: compat and memconf segments

7.1.4.30 stPACKAGE_READ_STATUS_CR7nnS

Status of various processes when reading the software package from the ecomatController.

Element	Data type	Description
stFirmware	stREAD_INFO_TYPE_CR7nnS (→ 93)	Firmware (OR runtime system) package, contains these segments: compat, boot, sissys, ifmOS

Element	Data type	Description
stIECApp_Standard	stREAD_INFO_TYPE_CR7nnS (→ 93)	IEC standard application package, contains these segments: standard app, standard retain, standard memory bytes
stIECApp_Safe		IEC safety application package, contains these segments: safe app, safe retain and safe memory bytes
stIECApps		IEC application package, contains these segments: safe app, standard app, memory settings package files (optional) and iomapping (optional)
stMemorySettings		Memory settings package, contains these segments: compat and memconf segments
xError	BOOL	• TRUE: The SDK API call has returned an error. • FALSE: The SDK API call did not return an error.

7.1.4.31 stPASSWORD_CR7nnS

Password list for each user level of the ecomatController.

► If no password is set, pass an empty string.

Element	Data type	Description
sAdmin	STRING(32)	Default: 'admin'
sS_Admin	STRING(32)	Default: 'admin'
sS_Service	STRING(32)	Default: 'admin'
sN_Admin	STRING(32)	Default: 'admin'
sN_Service	STRING(32)	Default: 'admin'

7.1.4.32 stINTF

Interface list.

Element	Data type	Description
sName	STRING(31)	List of available SDK communication interfaces.

7.1.5 GVL

7.1.5.1 gv_EvntCallBack

stLastEvt

The variable shows the last diagnostic response received from the SDK API callback event.

The variable can be used for diagnostic purposes by recording its value in the associated state of the FB, e.g. on a rising edge at the FB output `xError`.

The variable is of the type `stLAST_EVT_RESP` (→ [91](#)).

xEnableLogs

The variable can be used to enable or disable CODESYS device protocols for debugging purposes. These logs can be used to identify what is happening within the FB and to react in case of problems.

The variable is of the type `BOOL`.

TRUE: Protocols enabled, FALSE: Protocols disabled, default setting: FALSE

7.1.6 GCL

7.1.6.1 gc_AllSegments

List of default preset software segments for different controller types.

stBasic

This global constant list is used to select the software segments to be read from or written to a Basic Controller.

The constant is of the type stSEGM_LIST (→ 89)

Element	Data type	Description	Preset value for the selection
xFirmware	BOOL	Firmware segment	TRUE
xIECApp	BOOL	IEC application segment	TRUE
xIECAppCfg	BOOL	IEC application configuration segment	TRUE
xFileFlash	BOOL	File Flash segment	FALSE
xUserFlash	BOOL	User Flash segment	TRUE
xDataFlash	BOOL	Data Flash segment	FALSE
xRetaData	BOOL	Retain Data segment	TRUE
xMemoReta	BOOL	Memory Retain segment	FALSE
xEEPROM	BOOL	EEPROM segment	FALSE

TRUE = selected

FALSE = not selected

stR360_32bit

This global constant list is used to select the software segments to be read from or written to an R360 32-bit controller.

The constant is of the type stSEGM_LIST (→ 89).

Element	Data type	Description	Preset value for the selection
xFirmware	BOOL	Firmware segment	TRUE
xIECApp	BOOL	IEC application segment	TRUE
xIECAppCfg	BOOL	IEC application configuration segment	FALSE
xFileFlash	BOOL	File Flash segment	TRUE
xUserFlash	BOOL	User Flash segment	TRUE
xDataFlash	BOOL	Data Flash segment	TRUE
xRetaData	BOOL	Retain Data segment	TRUE
xMemoReta	BOOL	Memory Retain segment	FALSE
xEEPROM	BOOL	EEPROM segment	FALSE

TRUE = selected

FALSE = not selected

stR360_16bit

This global constant list is used to select the software segments to be read from or written to an R360 16 bit controller.

The constant is of the type stSEGM_LIST (→ 89).

Element	Data type	Description	Preset value for the selection
xFirmware	BOOL	Firmware segment	TRUE
xIECApp	BOOL	IEC application segment	TRUE
xIECAppCfg	BOOL	IEC application configuration segment	FALSE
xFileFlash	BOOL	File Flash segment	TRUE
xUserFlash	BOOL	User Flash segment	TRUE
xDataFlash	BOOL	Data Flash segment	TRUE
xRetaData	BOOL	Retain Data segment	TRUE
xMemoReta	BOOL	Memory Retain segment	FALSE
xEEPROM	BOOL	EEPROM segment	TRUE

TRUE = selected

FALSE = not selected

stReadR360_CR7nnS

This global constant list is used to select the software segments that are to be read from or written to an ecomatController.

The constant is of the type stPACKAGE_LIST_CR7nnS (→ 97).

Element	Data type	Description	Preset value for the selection
xFirmware	BOOL	Firmware / runtime system, bootloader and associated segments	TRUE
xIECApp_Standard	BOOL	IEC standard application and associated segments	TRUE
xIECApp_Safe	BOOL	IEC safety application and associated segments	TRUE
xIECApps	BOOL	IEC standard application, IEC safety application and associated segments	TRUE
xMemorySettings	BOOL	Package with memory settings	TRUE

TRUE = selected

FALSE = not selected

7.1.7 DUT

7.1.7.1 aSEGM_TYPE_NAME

Name	Data type	Description
aITF_SEGM_TYPE_NAME	ARRAY [0..7] of stSEGM_TYPE_NAME (→ 90)	List of names of segment types available in the file for R360 Basic, Classic / Extended Controllers

7.1.7.2 aINTF_LIST

Name	Data type	Description
aINTF_LIST	ARRAY [0..31] of stINTF (→ 98)	List of the names of the available SDK communication interfaces on the client system (e.g. ecomatDisplay)

7.1.7.3 aIO_DATA_CR7nnS

Name	Data type	Description
aIO_DATA_CR7nnS	ARRAY [0..200] of stIO_DATA_CR7nnS (→ 94)	IO mapping data of the ecomatController. Each array element represents an I/O data set.

7.1.7.4 aITF_SETTINGS_CAN_CR7nnS

Name	Data type	Description
aITF_SETTINGS_CAN_CR7nnS	ARRAY [0..7] of stITF_SETTINGS_CAN_CR7nnS (→ 95)	List of the CAN interfaces and the associated parameters of the ecomatController.

7.1.7.5 aITF_SETTINGS_ETH_CR7nnS

Name	Data type	Description
aITF_SETTINGS_ETH_CR7nnS	ARRAY [0..7] of stITF_SETTINGS_ETH_CR7nnS (→ 95)	List of the Ethernet interfaces and the associated parameters of the ecomatController.

7.1.7.6 aITF_SETTINGS_RS232_CR7nnS

Name	Data type	Description
aITF_SETTINGS_RS232_CR7nnS	ARRAY [0..7] of stITF_SETTINGS_RS232_CR7nnS (→ 95)	List of the RS232 interfaces and the associated parameters of the ecomatController

7.1.7.7 aDEVICE_LOG_INFO_CR7nnS

Name	Data type	Description
aDEVICE_LOG_INFO_CR7nnS	ARRAY [0..2000] of stDEVICE_LOG_INFO_CR7nnS (→ 96)	List of event logs of the ecomatController

7.1.7.8 aNETWORK_INFO_CR7nnS

Name	Data type	Description
aNETWORK_ INFO_ CR7nnS	ARRAY [0..8] of stNETWORK_INFO_CR7nnS (→ 93)	Network information of the ecomatCon- troller

7.1.7.9 aSOFTWARE_INFO_CR7nnS

Name	Data type	Description
aSOFTWARE_ INFO_ CR7nnS	ARRAY [0..31] of stSOFTWARE_INFO_CR7nnS (→ 92)	Software information of the ecomat- Controller