Data sheet CODESYS Professional Module Developer

The CODESYS Application Composer enables, for modular machines, the efficient, no-code creation and complete generation of suitable control applications of all sizes by combining and parameterizing multi-modal software modules.

With the one-year **Module Developer** workstation license, one can develop modules for the composition in the CODESYS Application Composer.

Further Information

Sales:	international: sales@codesys.com North America: US-Sales@codesys.com
Use and benefits (CODESYS Web Site)	international: https://www.codesys.com/products/engineering/application-composer North America: https://us.codesys.com/products/engineering/application-composer
Documentation: (CODESYS Online Help)	https://content.helpme-codesys.com/en/CODESYS%20Application%20Composer
Example projects:	https://content.helpme-codesys.com/en/CODESYS%20Examples/_ex_application_composer.html
Recommended Trainings: (CODESYS Academy)	Object-oriented programming, Application Composer: https://www.codesys.com/ecosystem/services/academy-training/training/

Technical Data

Operating the CODESYS features marked with '#' requires corresponding licenses on the controller.

Functional Areas

Module Definition	• Command "Create Module from Function Block" • Command "Create Macro Module" from module subtree • Command "Create Extension Module" for submodule slot • Module Declaration Editor • Deriving base module declarations via "IMPORTS"
Composition of module instances	• Definition of submodule slots with default submodules • Insertion into the module tree with selection of suitable submodules

	• Sequence editor for the execution order of submodules
Configuration of module instances	• Setting defined module parameters (e.g., size of a variable array) • Linking defined module I/Os with device channels via connection editor • Selecting a visualization for the module instance
Project generation	“Create” command and combined command “Create, Compile, and Log In”
Persistence Management	Assign IEC variables to persistence data groups, and configure persistence channels and storage files
Remote Modules	Exchange of module I/Os with module instances in other applications
IEC access to module tree	GetToplevelInstance, GetNextModule, GetFirstChild, NextSibling, GetParent, AncestorRange, …

Contents of the generated machine project

Generated IEC Programming	
• IEC Objects	POU, GVL, DUT, Task
• Module instance function block custom fit to module configuration	• Module Parameters: FB inputs set in the module editor • Module Slots: FB inputs filled in the module tree by a (default) submodule, with predefined parameterization and linking of the submodule-I/Os • Module References: FB inputs set in the module tree to other module instances§ • Module I/O: FB inputs and outputs mappable to device channels (channels of generated devices of the module, or via connection editor§) • VarArrays: Array with size determined by module parameters or by size of a multi-slot or a multi-reference • Proxies: Local proxies for module instances in other applications (§) Can also be located in another application of the project.
• Execution	• Automatic application and task assignment via module declaration • Execution order determined from module tree or sequence editor • Proxy data exchange via communication task, IRMPService, Network variables
Generated I/O level	
• devices per module	Generated as defined in module declaration or configured in module editor
• possible device types	All (SoftMotion axes#, field devices, bus couplers, device trees, …)

• possible fieldbuses#	All (PROFINET, EtherCAT, Ethernet/IP, …)
• configuration of the device	Generated as defined in module declaration or configured in module editor
• mapping of the device	• channels mapped to module I/O according to module declaration or connection editor • device FB mapped to module reference according to module declaration
Generated Visu Functionality	
• Visualization# per module	Visu objects for module FBs with embedded submodule visualization, Text Lists
• Alarms per module	Alarm Configuration Object, Alarm Storage Object, Alarm Class Object, Alarm Visualization
• Trends per module	TrendConfigObject, Trend Visualization

Miscellaneous Data

Scripting	YES (all functional areas named above)
Automation Platform SDK	• Define new sections in module definitions and generate them from them • Generation of additional project contents
Dependencies	• Installer 3.5.19.30 • Visualization 4.3.0.0 • Visualization Support 4.0.0.0 • Communication 4.0.0.0 • CFC 4.0.0.0 • Base Libraries 4.0.1.0 or 5.0.0.0 • Scripting 4.0.0.0 (optional) • License Provider 1.3.0.0 (optional) • Licensing Support 1.2.0.0 (optional)
Included Module Libraries	AC_ModuleBase, AC_DeviceDiagnosis, AC_Alarming, AC_Trend, AC_DataLog, AC_Persistence

General information

Supplier:

CODESYS GmbH
Memminger Strasse 151
87439 Kempten
Germany

Support:

Technical support is not included with this product. To receive technical support, please purchase a CODESYS Support Ticket.

https://support.codesys.com

Item:

CODESYS Professional Module Developer

Item number:

2111000035

Sales/Source of supply:

CODESYS Store
https://store.codesys.com

Included in delivery:

- Package “CODESYS Application Composer”
- license key

System requirements and restrictions

Programming System	CODESYS Development System 3.5.19.30 or higher
Runtime System	CODESYS Control Version 3.5.0.0
Supported Platforms/ Devices	Devices/PLCs on which the CODESYS runtime system is installed.
Additional Requirements	-
Restrictions	• Without valid license, the CODESYS Application Composer is functionally restricted to the Persistence Manager and the steps after the generation of the application (build, download, debug). • For licenses *Composer S* through *XL* you need CODESYS Application Composer Version 4.4.0.0 or newer.

	• For license *Composer Module Developer* you need CODESYS Application Composer Version 4.5.0.0 or newer.
Licensing	For the complete functionality (defining and composing own module to generate applications) a valid license is required. Licensing is done via the CODESYS License Provider mechanism and requires the following packages: • CODESYS License Provider >= 1.1.0.0 • CODESYS License Provider Enabler >= 1.1.0.0 • CODESYS Licensing Support >= 1.1.0.0 These are automatically solved and installed by the CODESYS installer during installation. The activation of the licenses is done via UFC license container (soft container or USB dongle) and is available as a workstation license. By simply reconnecting the CODESYS key, the license can be used at another workstation. The annual subscription is available as a workstation license. 
Required Accessories	Optional: CODESYS Key

Note: Technical specifications are subject to change. Errors and omissions excepted. The content of the current online version of this document applies.

Creation date: 2026-05-21