



## Data Sheet CODESYS Static Analysis

CODESYS Static Analysis is an integrated tool for checking the source code on the basis of defined rules.

CODESYS Static Analysis is part of the tool bundle CODESYS Professional Developer Edition.

### Product description

With the tool CODESYS Static Analysis it is possible to check the source code based on pre-defined rules and naming conventions in addition to the compiler code check. Additional information on potential development problems is revealed and errors are detected and eliminated before the application will be tested in the field. A lot of time can be saved during the development of applications for debugging.

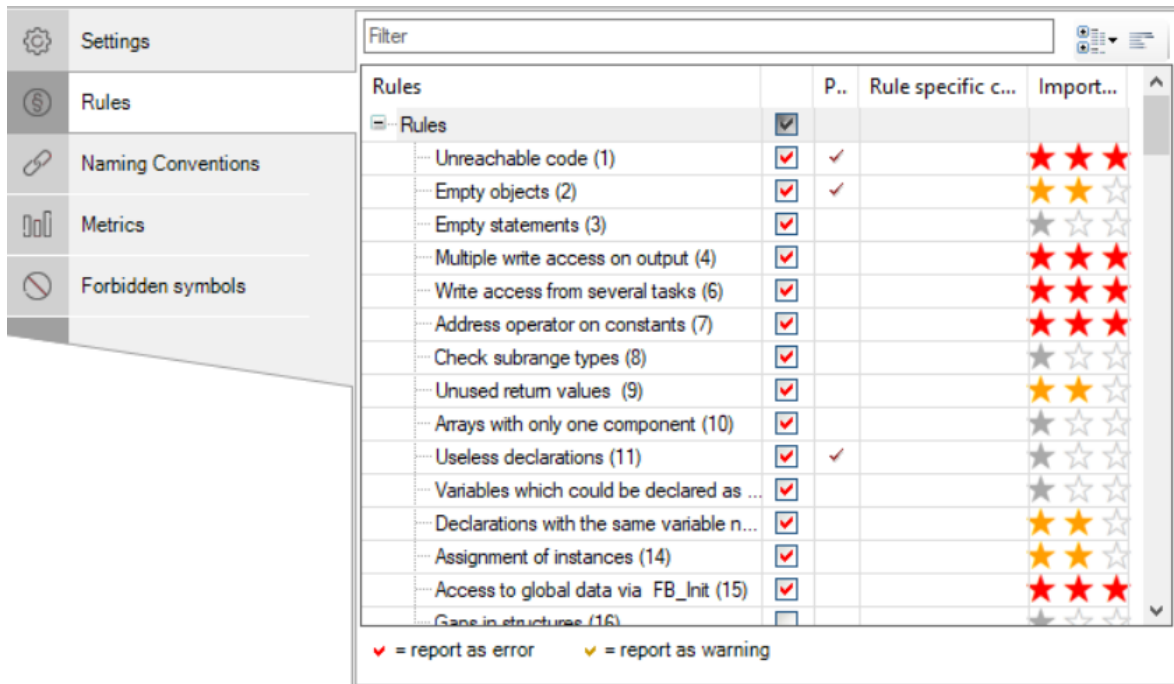
In CODESYS Static Analysis more than 100 pre-defined rules are implemented, where some of them are configurable. It is possible to combine these rules to individual sets of rules. The functions of the tool are completely integrated into the CODESYS Development System.

CODESYS Static Analysis helps to avoid mistakes already during programming and to get a better readable code. For example besides violations of coding rules also contradictory or not supported settings (e.g. of devices) can be detected.

A check of the application can be started via a menu command or automatically at each code generation. Besides this, pragmas and attributes can be used to disable the check for particular parts of the code

### Rules and Naming Conventions

Following the basic idea of the "Lint" analysis tool - a desired set of rules and naming conventions to be regarded can be configured in the project settings.



The following rules are part of the CODESYS Development System:

- |                            |                                   |
|----------------------------|-----------------------------------|
| • Unused variables         | • Concurrent access               |
| • Multiple usage of names  | • Multiple write access on output |
| • Overlapping memory areas |                                   |

With CODESYS Static Analysis the set of rules is extended by multiple rules like:

- |                                     |                                  |
|-------------------------------------|----------------------------------|
| • Detect unreachable code           | • Useless declarations           |
| • Find empty objects                | • Conversions                    |
| • Find empty statements             | • Write access to input variable |
| • Find empty objects                | • Rules for operators            |
| • Rules for FOR and CASE statements | • Check strict IEC rules         |
| • Working with prefixes             |                                  |

The complete set of rules is described in detail in CODESYS Online Help.

By means of naming conventions certain prefixes for types and scopes of variables, POU and DUTs in the application programming code can be defined.

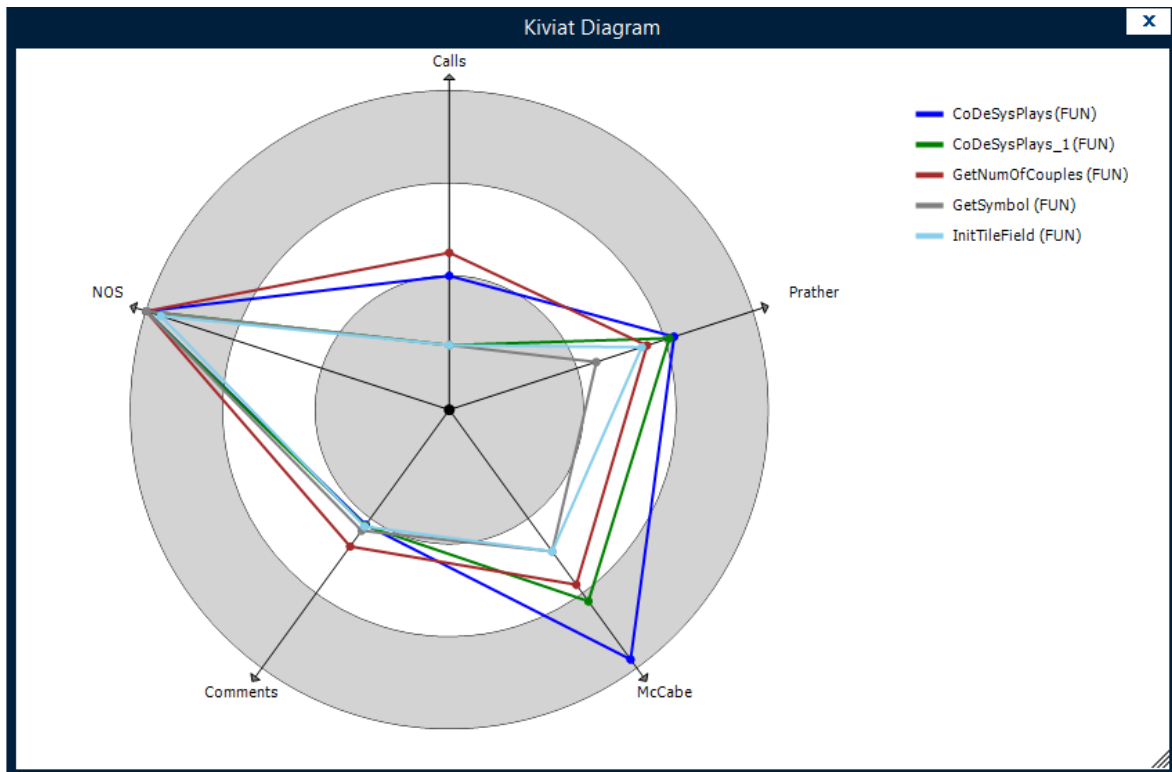
The results of the analysis, i.e. messages on violations of the rules will be displayed as errors in the messages view. Each violation has a unique identification number and can be clearly attributed to the configured rules and naming conventions.

Messages - Total 7 error(s), 0 warning(s), 5 message(s)

Static analysis messages			
6 error(s) 0 warning(s) 0 message(s)			
Description	Project	Object	Position
Static analysis...			
SA0031: Unused Function 'S_ADD'	StaticA...	S_ADD ...	
SA0031: Unused Type 'DUT'	StaticA...	DUT [D...	
SA0057: Possible loss of decimal places	StaticA...	PLC_PR...	Line 2, ...
SA0040: Possible division by zero	StaticA...	PLC_PR...	Line 2, ...
SA0001: Unreachable code detected in 'PLC_PRG'	StaticA...	PLC_PR...	Line 4, ...
SA0090: Return statement before end of function	StaticA...	PLC_PR...	Line 3, ...
Static analysis complete -- 6 errors, 0 warnings			

### Metrics

For evaluation of the code quality different metrics like the number of code lines, memory consumption or the software complexity as well as upper and lower metric limits can be configured. The result of the calculated metrics can be displayed and analyzed in tabular or graphical (Kiviati diagram) form.



### Notice

The analysis is only done on the application code in the current project. Libraries are not regarded.

## General information

### Supplier:

CODESYS GmbH  
 Memminger Strasse 151  
 87439 Kempten  
 Germany

### Support:

<https://support.codesys.com>

This product includes a CODESYS Support Ticket, i.e. 1 hour of support of CODESYS. More details can be found in the CODESYS Store product: [CODESYS Support Ticket](#).

### Item:

CODESYS Static Analysis

### Item number:

2101000002

### Sales/Source of supply:

CODESYS Store  
<https://store.codesys.com>

### Included in delivery:

- Package for the CODESYS Development System including license agreement and online help
- License key

## System requirements and restrictions

<b>Programming System</b>	CODESYS Development System V3.5.19.30 or higher
<b>Runtime System</b>	-
<b>Supported Platforms/ Devices</b>	-
<b>Additional Requirements</b>	Subscription of the CODESYS Professional Developer Edition.
<b>Restrictions</b>	64 bit support with version 4.2.0.0 and higher
<b>Licensing</b>	see CODESYS Professional Developer Edition
<b>Required accessories</b>	Optional: CODESYS Key

### Detailed compatibility information

Version	Programming System
4.4.3.0	3.5.19.30 - newest

4.4.2.0	3.5.16.0 - 3.5.19.20
4.4.1.0	3.5.16.0 - 3.5.19.20
4.4.0.0	3.5.16.0 - 3.5.19.20
4.3.1.0	3.5.12.30 - 3.5.15.x
4.3.0.0	3.5.12.30 - 3.5.15.x
4.2.4.0	3.5.12.30 - 3.5.15.x
4.2.3.0	3.5.12.30 - 3.5.15.x
4.2.2.0	3.5.12.30 - 3.5.15.x
4.2.1.0	3.5.12.0 - 3.5.15.x
4.2.0.0	3.5.11.0 - 3.5.15.x

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

Creation date: 2023-10-05