

# Product specification for products from the CODESYS Store

CODESYS® is a registered trademark. Technical specifications are subject to change. Errors and omissions excepted. No reproduction or distribution, in whole or in part, without prior permission. Please note: Not all CODESYS features are available in all territories. For more information on geographic restrictions, please contact [support@codesys.com](mailto:support@codesys.com).

Version: 1.0

File name: Produktspezifikation

# CONTENT

	Page
<b>1 Programmierrichtlinien</b>	Fehler! Textmarke nicht definiert.
1.1 Plug-ins	3
1.2 Libraries	<b>Fehler! Textmarke nicht definiert.</b>
<b>2 Creating a package</b>	Fehler! Textmarke nicht definiert.
2.1 General	3
2.2 Component	4
2.3 „StringTable“	4
2.4 „FileTable“	4
2.5 Further settings	<b>Fehler! Textmarke nicht definiert.</b>
<b>3 Miscellaneous</b>	Fehler! Textmarke nicht definiert.
3.1 Software	5
3.2 Updates	5
<b>Change History</b>	<b>6</b>

## 1 Programming guidelines

There are guidelines for the development of libraries and plug-ins that must be observed for all CODESYS products. A distinction is made between the following cases:


### 1.1 Plug-ins

For plug-ins, the programming guidelines of the CODESYS Automation Platform apply. We will provide you with the current version of the guidelines on request.

### 1.2 Libraries

The general guidelines for library development in CODESYS apply here. The current version of the coding guidelines is available in our online help for the CODESYS Development System in the chapter ["Guidelines for Creating Libraries"](#).

## 2 Creating a package

- Use the [CODESYS Package Designer](#) to create a package directly in the CODESYS programming system.
- Install the CODESYS Package Designer in the CODESYS Development System using the [CODESYS Installer](#) and restart the CODESYS Development System.
- Open the CODESYS Package Designer in the CODESYS Development System via the menu item Tools\Package Designer.
- First, use  to create a new package. Four entries are created in the Package Designer menu tree: "General", "Component", "StringTable", and "FileTable". Each package must contain at least one component ("Component"). For certain products (e.g. libraries with a sample project), it is advisable to create several components within one package. Each component then corresponds to a selectable feature during installation. You create a component by right-clicking on "General" > "Add Object" > "Component".
- Set the values in the created categories as described in the table under 2.1. Where necessary, set the values in multiple languages (at least in German and English), for example under "Description" or "LicenseAgreement". The procedure is explained in the sections "StringTable" and "FileTable".

### 2.1 General

Element	Value	
Copyright	Copyright information of the package	Mandatory field
Description	Short description of the product	Mandatory field
HTML	HTML file for file information	Optional
Icon	ICO file that is assigned to the package	Optional or default value
ID	GUID, is created uniquely for each new package (so please do not copy and reuse)	Mandatory field
License agreement (EULA)	HTML file containing the licensing conditions	Mandatory field
Name	Product name of the package	Mandatory field
ReadMe	txt or rtf file containing the „ReadMe“ information	Optional
RequiredInstallerVersion	Minimum version of the programming system required to run the package (at least CODESYS 3.5.1.0)	Mandatory field
Vendor	Manufacturer of the package	Mandatory field


Version	Package version in the format 1.0.0.0 (major.minor.build.revision)	Mandatory field
---------	---	-----------------

## 2.2 Component

Element	Value	
Description	Short description of the component	Mandatory field
Icon	ICO file that is assigned to the package	Optional oder Default value
ID	Internal ID of the component, is generated automatically	Default value
Name	Name of the component	Mandatory field
ProfileSelection.List	Some elements of the package can be installed in folders that users can define.	Default value
RequiredComponentIds	Definition of dependencies between components; Default value is selectable	Default value
Selectable	Determines whether users can activate or deactivate the component during installation.	Default value
SelectableByDefault	Determines whether the component is initially activated or not.	Default value

## 2.3 „StringTable“

In this table, you can define the string entries within the "General" and "Component" categories in different languages using placeholders.

Create a new placeholder by entering a new value in the "New placeholder" row in the "Placeholder" column (e.g. "Description"). The value for the placeholder is then set in English ("neutral") in the "Value" column. If you want to add an additional value, e.g. in German, activate the relevant line and press . Once you have selected a language in the window that appears, you can define the value of the placeholder in the corresponding language. You can navigate through the tables of the different languages at any time using the selection dialog.

To ensure that the predefined values in CODESYS are set in the correct language, put the placeholders in the corresponding entries via "\$placeholder name". The already defined placeholders are available via a selection dialog at the end of each string entry. Entries without placeholders appear in each language as a fixed value (e.g. "Vendor").

## 2.4 „FileTable“

In this table, you can define the file entries (e.g. "LicenseAgreement") within the "General" and "Component" categories in different languages using placeholders.

The placeholders are set in the same way as in the "StringTable" category. To ensure that the predefined values in CODESYS are also set in the correct language, put the placeholders in the corresponding file entries using "\$Placeholder name". The already defined placeholders are available via a selection dialog at the end of the line of each file entry. Entries without placeholders appear in each language as a fixed value.

## 2.5 Further settings

Add the product components (e.g. Library or DeviceDescription) to the respective component by right-clicking on "Component" > "Add Object" > "Items" > "...".

Define the target directory for sample projects or other files via the "TargetDirectoryDefinition". Create the entry by right-clicking on "General" > "Add Object" > "TargetDirectoryDefinition". By default, examples that are installed with CODESYS are stored in %USERPROFILE%\CODESYS Examples\. We recommend a folder path containing the product name and the version number of the product for storing your own product components, e.g. %USERPROFILE%\CODESYS Examples\ProductName\1.0.0.0\.

You can either install help files directly into the project directory or integrate them directly into the CODESYS Help for CODESYS Store products by right-clicking on "Component" > "Add Object" > "Items" >

"OnlineHelpFile" or "OnlineHelpMerge". The "OnlineHelpFile" object contains the help itself. "OnlineHelpMerge" defines where the new object is integrated into the existing CODESYS online help. We will be happy to provide you with an example of a merge file on request.

Please note: In the case of license-protected libraries, only libraries with one license may be included in a package. Multiple libraries with different product codes are not permitted; multiple libraries with the same product code are permitted.

### **3 Miscellaneous**

#### **3.1 Software**

The software must be executable and tested with the specified CODESYS minimum version. Calls to external systems must be documented.

#### **3.2 Updates**

Please mark your product updates with a date, a new product version, and the corresponding CODESYS version. If you would also like to make older versions available in our CODESYS Store (see version history in the Store for the respective product under the Versions tab), please also provide a brief description (release notes) of the changes made.