



Data Sheet CODESYS File-Based Storage

CODESYS File-Based Storage offers an alternative project storage format for CODESYS. In contrast to normal, binary project storage, projects are stored text-based in a folder structure similar to the project tree.

CODESYS File-Based Storage is part of the CODESYS Professional Developer Edition tool bundle.

Product description

The normal CODESYS project storage creates a single binary file whose contents cannot be read or created by other tools.

CODESYS File-Based Storage, on the other hand, stores a project as a folder tree on disk that mirrors the project tree. One or more text files are created for each object, and these files can be processed or even created by other text editors: IEC source code as plain text, project metadata as JSON.

Typical use cases:

- **The path towards CODESYS 4** – CODESYS 4 stores projects file-based natively. For libraries the format is identical: a library converted to `.fbslib` is already stored in the format that CODESYS 4 reads and writes. For development projects, the conversion step will follow in upcoming versions.
- **Automation in CI/CD** – the scripting interface opens, converts and validates projects without a user interface.
- **Analysis and tooling** – the whole project can be searched and statically checked with familiar tools.

The format is open and documented. Read access from outside CODESYS is supported, but projects are edited in the CODESYS Development System: whether a change written from outside results in a valid project is decided only when CODESYS loads and builds it. Note that CODESYS V3 does not allow inconsistencies in the project – it may be impossible to open or display projects with an invalid status.

For use in the context of Git, we advise you to use the upcoming CODESYS Git 2.0.0.0 or later. It understands both the storage format and the CODESYS project model and therefore ensures that all files belonging to a project are committed together and in a consistent state.

For object types for which there is not yet a suitable storage format, the xml-based export format is used.

CODESYS File-Based Storage brings a separate set of new commands to deal with file-based projects or libraries.

Storage format

- Project types stored as folders: `.fbproj` (development project), `.fbslib` (library) and `.fbslibpack` (library package as a single file)
- Programming objects as individual files following the pattern `<Name>.<class>.<language>`, for example `Timer.fb.st`; objects with children become container directories
- Programs, function blocks, functions, methods, actions, transitions, interfaces, global variable lists, DUTs (struct, enum, union, alias) and properties are stored as readable Structured Text
- The declaration part is always Structured Text, regardless of the implementation language – variable declarations therefore compare cleanly even for graphically implemented POU's
- Library Manager, project information, project settings and folder metadata as indented JSON
- Deterministic output: UTF-8 without BOM, LF line endings, tab indentation, stable ordering, no timestamps and no absolute paths – the same project always produces the same files
- Lossless fallback: object types without a dedicated representation are stored as `.xml.v3`, objects that cannot be loaded are preserved unchanged

Conversion and migration

- Explicit conversion of binary projects and libraries (`.project` / `.library`) into the folder format – as a command in the development system or through the Python scripting interface
- In-place migration as soon as an object type or language gains native support later on: only the representation on disk changes, the identity and semantics of the objects remain untouched

Automation

- Scripting interface for the CODESYS Python scripting engine: open projects and libraries without a user interface, convert binary projects, and run the supplied validators for known problem cases – usable as a quality gate in CI/CD pipelines

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 File-Based Storage

Item number:

2101000019

Sales/Source of supply:

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

Included in delivery:

- Package for the CODESYS Development System including license agreement

System requirements and restrictions

Programming System	CODESYS Development System V3.5.22.0 or higher
Runtime System	-
Supported Platforms/ Devices	-
Additional Requirements	Subscription of the CODESYS Professional Developer Edition Recommended for version control: the upcoming CODESYS Git 2.0.0.0 or later For use in CI/CD pipelines, the build agent requires a CODESYS installation with the File-Based Storage add-on and a valid license. Headless means without an interactive user interface, not without a CODESYS installation.
Restrictions	A login without online change to a running existing application with a new file-based project may work but cannot be guaranteed.

A previously installed earlier version (< 1.0.0.0) must be uninstalled. In addition, the file FileBasedStorage.dll must be deleted from the Common directory.

Due to the file-based concept, many individual files are processed when opening and saving a project instead of a single binary file. Loading and saving can therefore take longer than with binary project storage.

Licensing	see CODESYS Professional Developer Edition Without a valid license, file-based projects open read-only and the Save command is disabled.
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: 2026-07-28