

## Datenblatt CODESYS File-Based Storage

CODESYS File-Based Storage bietet ein alternatives Projektablage-Format für CODESYS. Im Gegensatz zur normalen, binären Projektablage werden damit Projekte textbasiert in einer dem Projektbaum ähnlichen Ordnerstruktur abgelegt.

CODESYS File-Based Storage ist Teil des Tool-Bundles CODESYS Professional Developer Edition.

### Produktbeschreibung

Die normale CODESYS Projektablage erzeugt eine einzige binäre Datei, deren Inhalte nicht von anderen Tools gelesen oder erzeugt werden können.

CODESYS File-Based Storage hingegen legt ein Projekt als Ordnerbaum auf der Festplatte ab, der dem Projektbaum entspricht. Pro Objekt werden eine oder mehrere Textdateien erzeugt, die von anderen Texteditoren gelesen oder gar erzeugt werden können: IEC-Quellcode als reiner Text, Projekt-Metadaten als JSON.

#### Typische Anwendungsfälle:

- **Weg in Richtung CODESYS 4** – CODESYS 4 speichert Projekte von Haus aus dateibasiert. Für Bibliotheken ist das Format identisch: eine nach `.fbslib` konvertierte Bibliothek liegt bereits in dem Format, das CODESYS 4 liest und schreibt. Für Entwicklungsprojekte folgt der Konvertierungsschritt in kommenden Versionen.
- **Automatisierung in CI/CD** – über die Skriptschnittstelle lassen sich Projekte ohne Bedienoberfläche öffnen, konvertieren und prüfen.
- **Analyse und Werkzeugunterstützung** – das gesamte Projekt kann mit gewohnten Werkzeugen durchsucht und statisch geprüft werden.

Das Format ist offen und dokumentiert. Lesender Zugriff von außerhalb CODESYS ist unterstützt, bearbeitet werden Projekte jedoch im CODESYS Development System: ob eine von außen geschriebene Änderung ein gültiges Projekt ergibt, entscheidet erst das Laden und Übersetzen in CODESYS. Dabei ist zu beachten, dass CODESYS V3 keine Inkonsistenzen im Projekt erlaubt – Projekte mit ungültigem Zustand können evtl. nicht geöffnet oder angezeigt werden.

Für die Verwendung im Kontext von Git empfehlen wir die Verwendung des kommenden CODESYS Git 2.0.0.0 oder neuer. Es kennt sowohl das Ablageformat als auch das CODESYS Projektmodell und stellt damit sicher, dass alle zu einem Projekt gehörenden Dateien gemeinsam und in einem konsistenten Zustand eingecheckt werden.

Für Objekttypen, für die es (noch) kein geeignetes Format gibt, wird das xml-basierte Exportformat verwendet.

CODESYS File-Based Storage bringt einen Satz neuer Kommandos mit, die für die Arbeit mit den dateibasierten Projekten oder Bibliotheken verwendet werden können.

### Ablageformat

- Projekttypen als Ordner: `.fbproj` (Entwicklungsprojekt), `.fbslib` (Bibliothek) sowie `.fbslibpack` (Bibliothekspaket als Einzeldatei)
- Programmierobjekte als einzelne Dateien nach dem Schema `<Name>.<Klasse>.<Sprache>`, zum Beispiel `Timer.fb.st`; Objekte mit Unterobjekten werden zu Container-Verzeichnissen
- Programme, Funktionsblöcke, Funktionen, Methoden, Aktionen, Transitionen, Schnittstellen, globale Variablenlisten, DUTs (Struct, Enum, Union, Alias) und Eigenschaften werden als lesbarer Strukturierter Text abgelegt
- Der Deklarationsteil ist unabhängig von der Implementierungssprache immer Strukturierter Text – Variablendeklarationen sind damit auch bei grafisch implementierten POU's sauber vergleichbar
- Bibliotheksverwaltung, Projektinformationen, Projekteinstellungen und Ordner-Metadaten als eingerücktes JSON
- Deterministische Ausgabe: UTF-8 ohne BOM, LF-Zeileneenden, Tabulator-Einrückung, stabile Sortierung, keine Zeitstempel und keine absoluten Pfade – dasselbe Projekt erzeugt immer dieselben Dateien
- Verlustfreier Rückfall: Objekttypen ohne eigene Repräsentation werden als `.xml.v3` abgelegt, nicht ladbare Objekte bleiben unverändert erhalten

### Konvertierung und Migration

- Explizite Konvertierung binärer Projekte und Bibliotheken (`.project` / `.library`) in das Ordnerformat – als Befehl im Entwicklungssystem oder über die Python-Skriptschnittstelle
- Migration im Bestand, sobald ein Objekttyp oder eine Sprache nachträglich native Unterstützung erhält: es ändert sich nur die Ablage auf der Festplatte, Identität und Semantik der Objekte bleiben unberührt

### Automatisierung

- Skriptschnittstelle für die CODESYS Python-Skript-Engine: Projekte und Bibliotheken ohne Bedienoberfläche öffnen, binäre Projekte konvertieren und die mitgelieferten Validatoren auf bekannte Problemfälle ausführen – nutzbar als Qualitätsschranke in CI/CD-Pipelines

## Allgemeine Informationen

### Lieferant:

CODESYS GmbH  
 Memminger Straße 151  
 87439 Kempten  
 Deutschland

### Support:

Technischer Support ist bei diesem Produkt nicht enthalten. Um technischen Support zu erhalten, erwerben Sie bitte ein CODESYS Support Ticket.

<https://support.codesys.com>

### Artikelname:

CODESYS File-Based Storage

### Artikelnummer:

2101000019

### Vertrieb/Bezugsquelle:

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

### Lieferumfang:

- Package für das CODESYS Development System inklusive der Lizenzvereinbarung.

## Systemvoraussetzungen und Einschränkungen

|                                             |                                                                                                                                                                                                                                                                                                                                                                                                           |
|---------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Programmiersystem</b>                    | CODESYS Development System V3.5.22.0 oder höher                                                                                                                                                                                                                                                                                                                                                           |
| <b>Laufzeitsystem</b>                       | -                                                                                                                                                                                                                                                                                                                                                                                                         |
| <b>Unterstützte Plattformen/<br/>Geräte</b> | -                                                                                                                                                                                                                                                                                                                                                                                                         |
| <b>Zusätzliche Anforderungen</b>            | <p>Abonnement der CODESYS Professional Developer Edition</p> <p>Für die Versionierung empfohlen: das kommende CODESYS Git 2.0.0.0 oder neuer</p> <p>Für den Einsatz in CI/CD-Pipelines wird auf dem Build-Agent eine CODESYS Installation mit dem File-Based Storage Add-on und einer gültigen Lizenz benötigt. Headless bedeutet ohne interaktive Bedienoberfläche, nicht ohne CODESYS Installation.</p> |
| <b>Einschränkungen</b>                      |                                                                                                                                                                                                                                                                                                                                                                                                           |

Ein Login ohne Online-Change auf eine bestehende, laufende Applikation mit einem neuen dateibasierten Projekt kann funktionieren, aber es kann nicht garantiert werden.

Eine zuvor installierte Vorversion (< 1.0.0.0) muss deinstalliert werden. Zusätzlich muss die Datei FileBasedStorage.dll aus dem Common-Verzeichnis gelöscht werden.

Bedingt durch das dateibasierte Konzept werden beim Öffnen und Speichern viele einzelne Dateien verarbeitet statt einer einzigen Binärdatei. Die Lade- und Speicherzeiten können dadurch höher ausfallen als bei der binären Projektablage.

|                               |                                                                                                                                                                  |
|-------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Lizenzierung</b>           | siehe CODESYS Professional Developer Edition<br><br>Ohne gültige Lizenz werden dateibasierte Projekte nur lesend geöffnet, der Befehl Speichern ist deaktiviert. |
| <b>Erforderliches Zubehör</b> | Optional: CODESYS Key                                                                                                                                            |

*Bitte beachten Sie: Technische Änderungen, Druckfehler und Irrtümer vorbehalten. Es gilt der Inhalt der aktuellen Online-Version dieses Dokuments.*

Erstellungsdatum: 28.07.2026