



CODESYS File-Based Storage 0.9.0.0

Whitepaper

Version: <0.9.0.0">

Template: templ_tecdoc_de_V3.0.docx

Dateiname: Whitepaper-CODESYS-File-Based Storage-0900.docx

INHALT

	Seite
1 What is CODESYS File-Based Storage	3
1.1 Advantages	3
1.2 Disadvantages	3
2 File system layout	3
2.1 Project layout	3
2.2 File layout	4
2.2.1 POU, DUTs, and GVLs	4
2.2.2 Library Manager	7
2.2.3 Project information	8
2.2.4 Any other objects	9
3 Conversion of existing projects	10
4 Limitations	11
4.1 User Management	11
4.2 Project Archive	11
4.3 Online Change	11
4.4 File system	11
4.5 Validators	11
5 Scripting	12
6 License	12

1 What is CODESYS File-Based Storage

CODESYS File-Based Storage (FBS) is a new, alternative, file-based project format. The classic V3 project or library stores its data within a single compressed file (*.project or *.library). In contrast to this existing format, the FBS project is stored directly as a folder structure on the disc. Individual files are serialized in a human-readable format and can be read with any editor.

Using CODESYS File-Based Storage requires at least CODESYS V3.5 SP 20 Patch 4 or higher.

The project root consists of folders with the following extensions:

- "*.fbproj": for projects
- "*.fbprojlib": for library-projects

1.1 Advantages

- Human-readable project structure
The project and its content are human-readable on disc and can be edited.
- Simplified project creation
New projects can be set up just by creating a folder structure and the files you need

1.2 Disadvantages

- Increased storage space requirements
Since the context is serialized as plain text, the project requires more disc space than classic V3 files whose content is binary.
- Increased loading time
Reading a large number of small files takes more time than reading a single large file.

2 File system layout

2.1 Project layout

A standard V3 project gets serialized to the file system as follows. Folders or files with the prefix "." are hidden and not visible within the CODESYS V3 IDE.

Path	Explanation
StandardDemo.fbproj\	Root folder of the project
root\auxiliary\	Storage folder for auxiliary files created by plug-ins
root\sidecars\	Folder for project-relevant files, like options, precompile context, ...
root\PLC.device.xml.v3\	Folder for sub objects of the device
root\PLC.device.xml.v3\Plc Logic.plclogic.xml.v3\	Folder for sub objects of the PLC logic
root\PLC.device.xml.v3\Plc Logic.plclogic.xml.v3\Application.application.xml.v3\	Folder for sub objects of the application
root\PLC.device.xml.v3\Plc Logic.plclogic.xml.v3\Application.application.xml.v3\SomeInterface.itf\	Folder for sub objects of the interface
root\PLC.device.xml.v3\Plc Logic.plclogic.xml.v3\Application.application.xml.v3\SomeInterface.itf\MethodName.meth	Method

root\PLC.device.xml.v3^\Plc Logic.plclogic.xml.v3^\Application.application.xml.v3^\PLC_PRG.prg.st^\	Folder for sub objects of the POU
root\PLC.device.xml.v3^\Plc Logic.plclogic.xml.v3^\Application.application.xml.v3^\PLC_PRG.prg.st^\Inch.prop.st	Property in ST
root\PLC.device.xml.v3^\Plc Logic.plclogic.xml.v3^\Application.application.xml.v3^\Task Configuration.taskconfig.xml.v3^\	Folder for sub objects of the task configuration
root\PLC.device.xml.v3^\Plc Logic.plclogic.xml.v3^\Application.application.xml.v3^\Task Configuration.taskconfig.xml.v3^\MainTask.task.xml.v3	Task object
root\PLC.device.xml.v3^\Plc Logic.plclogic.xml.v3^\Application.application.xml.v3^\AbstractFB.fb.st	Function block in ST
root\PLC.device.xml.v3^\Plc Logic.plclogic.xml.v3^\Application.application.xml.v3^\CFC.pou.xml	POU in CFC
root\PLC.device.xml.v3^\Plc Logic.plclogic.xml.v3^\Application.application.xml.v3^\Constants.gvl	Global variable list object
root\PLC.device.xml.v3^\Plc Logic.plclogic.xml.v3^\Application.application.xml.v3^\ISomeInterface.itf	Interface object
root\PLC.device.xml.v3^\Plc Logic.plclogic.xml.v3^\Application.application.xml.v3^\Libraries.json	Library Manager
root\PLC.device.xml.v3^\Plc Logic.plclogic.xml.v3^\Application.application.xml.v3^\PLC_PRG.prg.st	POU in ST
root\PLC.device.xml.v3^\Plc Logic.plclogic.xml.v3^\Application.application.xml.v3^\Task Configuration.taskconfig.xml.v3	Task configuration object
root\PLC.device.xml.v3^\Plc Logic.plclogic.xml.v3	PLC logic object
root\PLC.device.xml.v3	Device object
root\Project Settings.workspace.xml.v3	Workspace object
root\ProjectInfo.json	Project Information

In the above table, paths marked with a backslash '\' at the end are folders and therefore not visible in the file system.

2.2 File layout

Depending on the kind of object and implementation language, the content will be serialized in a new human-readable format. Currently this is only implemented for objects based on Structured Text (ST) language.

2.2.1 POU, DUTs, and GVLs

POUs, DUTs, and GVLs are represented by files, as outlined below. "Name" represents the POU name.

"Child" objects (like properties, methods, ...) sit in a folder bearing the name of the parent + "^".

For single language POUs, the language is added as part of a two-part extension.

Name.prg.st	Program "Name", written in ST, including declaration and implementation part
Name.gvl	The global variable list "Name"
Name.fb.xml.v3^\PropName.prop.st	The property "PropName" of the FB "Name"
Name.fb.xml.v3^\ActionName.action.st	The action "ActionName" of the POU "Name" (which must be CFC/SFC/FBD/LD in this case), written in ST

Name.fb.xml.v3^\TransName.transition.st	The transition "TransName" of the POU "Name" (which must be CFC/SFC/FBD/LD in this case), written in ST
ISomeInterface.itf	Interface declaration
ISomeInterface.itf^\MethodName.meth	The method "MethodName" of the interface "ItfName"
SomeEnum.enum	Declaration of an ENUM data type
PostalAddress.struct	Declaration of a STRUCT data type
SomeUnion.union	Declaration of a UNION data type
Temperature.alias	Declaration of an IEC ALIAS for another data type

PLC_PRG.prg.st

```

1  __METADATA__
2  {
3    "v3Meta": {
4      "objectGuid": "9a9badf8-6b26-4674-bdb3-050acd691c67",
5      "objectTypeGuid": "6f9dac99-8de1-4efc-8465-68ac443b7d08",
6      "embeddedObjectTypeGuids": [
7        "a9ed5b7e-75c5-4651-af16-d2c27e98cb94",
8        "3b83b776-fb25-43b8-99f2-3c507c9143fc"
9      ],
10     "properties": {},
11     "subObjects": {}
12   }
13 }
14 __DECLARATION__
15 PROGRAM PLC_PRG
16 VAR
17   switchOn: BOOL;
18   lightOn: BOOL;
19   blinker: Blinker;
20   interval: TIME := T#1S;
21 END_VAR
22 __IMPLEMENTATION__
23 blinker(enable:=switchon,interval:=interval);
24 lightOn:= blinker.output;

```

Inch.prop.st

```

1  __METADATA__
2  {
3    "v3Meta": {
4      "objectGuid": "2e8ea291-f78d-46eb-994a-536f663b64c2",
5      "objectTypeGuid": "5a3b8626-d3e9-4f37-98b5-66420063d91e",
6      "embeddedObjectTypeGuids": [
7        "a9ed5b7e-75c5-4651-af16-d2c27e98cb94"
8      ],
9      "properties": {},
10     "subObjects": {
11       "set": {
12         "objectGuid": "33bab876-4f14-4898-ad89-bd68d84335f8",
13         "objectTypeGuid": "792f2eb6-721e-4e64-ba20-bc98351056db",
14         "embeddedObjectTypeGuids": [
15           "a9ed5b7e-75c5-4651-af16-d2c27e98cb94",
16           "3b83b776-fb25-43b8-99f2-3c507c9143fc"
17         ],
18         "properties": {},
19         "subObjects": {}
20       },
21       "get": {
22         "objectGuid": "42dcd8f6-e874-46ae-8d3a-c8629397b692",
23         "objectTypeGuid": "792f2eb6-721e-4e64-ba20-bc98351056db",
24         "embeddedObjectTypeGuids": [
25           "a9ed5b7e-75c5-4651-af16-d2c27e98cb94",
26           "3b83b776-fb25-43b8-99f2-3c507c9143fc"
27         ],
28         "properties": {},
29         "subObjects": {}
30       }
31     }
32   }
33 }
34 __DECLARATION__
35 PROPERTY Inch : REAL
36 __SETTERDECLARATION__
37 VAR
38   cm : real;
39 END_VAR
40 __SETTERIMPLEMENTATION__
41 cm := inch * 2.54;
42 __GETTERDECLARATION__
43 VAR
44   cm : real;
45 END_VAR
46 __GETTERIMPLEMENTATION__
47 Inch := cm / 2.54;

```

Constants.gvl

```

1  __METADATA__
2  {
3    "v3Meta": {
4      "objectGuid": "e0127516-efba-475f-ab2c-85e9f3157db1",
5      "objectTypeGuid": "ffbfa93a-b94d-45fc-a329-229860183b1d",
6      "embeddedObjectTypeGuids": [
7        "a9ed5b7e-75c5-4651-af16-d2c27e98cb94"
8      ],
9      "properties": {},
10     "subObjects": {}
11   }
12 }
13 __DECLARATION__
14 {attribute 'qualified_only'}
15 VAR_GLOBAL
16   iHugo : INT := 42;
17   iOtto : INT := 4711;
18 END_VAR

```

2.2.2 Library Manager

Library managers of projects get fixed file names and extensions. Their content will be serialized as JSON.

Libraries.json

```

1  {
2    "references": {
3      "Standard": {
4        "$type": "Placeholder",
5        "placeholder": "Standard",
6        "defaultResolution": "Standard, * (System)",
7        "allowUnqualifiedAccess": true
8      }
9    },
10   "placeholderOverrides": {
11     "Standard": "Standard, 3.5.18.0 (System)"
12   },
13   "v3Meta": {
14     "objectName": null,
15     "objectGuid": "c4177c08-cc2c-48a0-acf5-ac0af68e20df",
16     "objectTypeGuid": "adb5cb65-8e1d-4a00-b70a-375ea27582f3",
17     "embeddedObjectTypeGuids": [],
18     "properties": {},
19     "subObjects": {}
20   },
21   "libraryParameters": {},
22   "v3SystemLibraries": [
23     {
24       "namespace": "_3S_LICENSE",
25       "libraryReference": {
26         "$type": "Placeholder",
27         "placeholder": "3SLicense",
28         "defaultResolution": "3SLicense, 0.0.0.0 (3S - Smart Software Solutions GmbH)",
29         "resolverGuid": "97c3b452-d9fa-4ac2-9d0c-3d420aa6d95b",
30         "allowUnqualifiedAccess": true
31       }
32     },
33     {
34       "namespace": "BPLog",
35       "libraryReference": {
36         "$type": "Placeholder",
37         "placeholder": "BreakpointLogging",
38         "defaultResolution": "Breakpoint Logging Functions, * (3S - Smart Software Solutions GmbH)",
39         "allowUnqualifiedAccess": true
40       }
41     },
42     {
43       "namespace": "DED",
44       "libraryReference": {
45         "$type": "Placeholder",
46         "placeholder": "CAA Device Diagnosis",
47         "defaultResolution": "CAA Device Diagnosis, 3.5.20.0 (CAA Technical Workgroup)",
48         "allowUnqualifiedAccess": true
49       }
50     },
51     {
52       "namespace": "IoStandard",
53       "libraryReference": {
54         "$type": "Placeholder",
55         "placeholder": "IoStandard",
56         "defaultResolution": "IoStandard, 3.5.17.0 (System)",
57         "allowUnqualifiedAccess": true,
58         "linkAllContent": true
59       }
60     }
61   ]
62 }
```

2.2.3 Project information

Project information objects get fixed file names and extensions, too. Their content will be serialized as JSON.

ProjectInfo.json

```
1  {
2    "title": "StandardDemo",
3    "version": "0.0.0.1",
4    "author": "Creator",
5    "company": "Demo Inc.",
6    "defaultNamespace": "ThisIsDemo",
7    "v3Meta": {
8      "objectGuid": "11c0fc3a-9bcf-4dd8-ac38-efb93363e521",
9      "objectTypeGuid": "085afe48-c5d8-4ea5-ab0d-b35701fa6009",
10     "embeddedObjectTypeGuids": [],
11     "properties": {},
12     "subObjects": {}
13   },
14   "customProperties": {
15     "DemoInt": "42",
16     "DemoText": "HelloWorld",
17     "Project": ""
18   }
19 }
```


2.2.4 Any other objects

Other objects that do not have a dedicated textual representation yet will be serialized as XML. This format is also used for the native import/export features.

```

PLC.device.xml.v3

1  __V3_META__
2  {
3    "objectId": "3ee64819-e85a-4ee7-8783-36295202d99c",
4    "objectTypeGuid": "225bfe47-7336-4dbc-9419-4105a7c831fa",
5    "embeddedObjectTypeGuids": [],
6    "properties": {
7      "_3S.CoDeSys.DeviceObject.DeviceProperty": "<?xml version='1.0' encoding='utf-8'>
8        <Single xml:space='preserve' Type='{cdadfc2b-8598-4621-ad56-5b1df7db910f}' Method='IArchivable'>
9          <Single Name='DeviceIdentification' Type='{05ee0b98-6639-4276-bc32-b75578b819ef}' Method='IArchivable'>
10             <Single Name='Type' Type='int'>4096</Single>
11             <Single Name='Id' Type='string'>0000 0001</Single>
12             <Single Name='Version' Type='string'>3.5.20.40</Single><Null Name='BaseName' />
13           </Single>
14         </Single>"
15    },
16    "subObjects": {}
17  }
18  __V3_CONTENT__
19  <?xml version='1.0' encoding='utf-8'>
20  <Single xml:space='preserve' Type='{225bfe47-7336-4dbc-9419-4105a7c831fa}' Method='IArchivable'>
21    <Single Name='UniqueIdGenerator' Type='string'>4105</Single>
22    <Null Name='TypeList' />
23    <Single Name='NeedsBusCycle' Type='bool'>False</Single>
24    <Single Name='Id' Type='{05ee0b98-6639-4276-bc32-b75578b819ef}' Method='IArchivable'>
25      <Single Name='Type' Type='int'>4096</Single>
26      <Single Name='Id' Type='string'>0000 0001</Single>
27      <Single Name='Version' Type='string'>3.5.20.40</Single>
28      <Null Name='BaseName' />
29    </Single>
30    <Single Name='DefaultDeviceInfo' Type='{59a90934-c888-40db-a9df-306fcf36d75c}' Method='IArchivable'>
31      <Single Name='Name' Type='string'>CODESYS Control Win V3</Single>
32      <Single Name='Description' Type='string'>CODESYS V3 Soft-PLC for Windows with non realtime capabilities (CODESYS Control Win V3)</Single>
33      <Single Name='Vendor' Type='string'>CODESYS</Single>
34      <Single Name='OrderNumber' Type='string'>???'</Single>
35      <Array Name='Categories' Type='int'>ABAAAA=</Array>
36      <Array Name='Families' Type='string' />
37      <Single Name='Custom' Type='string'></Single>
38      <Null Name='DefaultInstanceName' />
39    </Single>
40    <Single Name='DeviceParameterSet' Type='{abc12bfe-e34e-4b2c-a058-42c6e7e03a13}' Method='IArchivable'>
41      <List2 Name='Params' />
42      <Single Name='LmGuid' Type='System.Guid'>1616e230-4766-5140-b958-ebd8e198cf3a</Single>
43      <Null Name='Sections' />
44      <Single Name='InitOnlineChange' Type='bool'>False</Single>
45      <Single Name='AlwaysMapping' Type='bool'>False</Single>
46      <Null Name='EditorName' />
47      <Single Name='AlwaysMappingMode' Type='{9d1b24fb-5845-4978-8458-564ca7830fec}'>OnlyIfUnused</Single>
48      <Single Name='CreateBitChannels' Type='bool'>False</Single>
49    </Single>
50    <Single Name='Disable' Type='bool'>False</Single>
51    <Single Name='Exclude' Type='bool'>False</Single>
52    <List Name='Connectors' Type='System.Collections.ArrayList' />
53    .
54    .
55    .
56    <Single Name='LogicalLanguageModelPositionId' Type='long'>-1</Single>
57    <List Name='SupportedLogicalBusSystems' Type='System.Collections.ArrayList' />
58    <Single Name='MappingPossible' Type='bool'>False</Single>
59    <Single Name='HidePropertiesDialog' Type='bool'>False</Single>
60    <Null Name='PlaceholderResolution' />
61    <Null Name='RedirectedPlaceholderResolution' />
62  </Single>

```

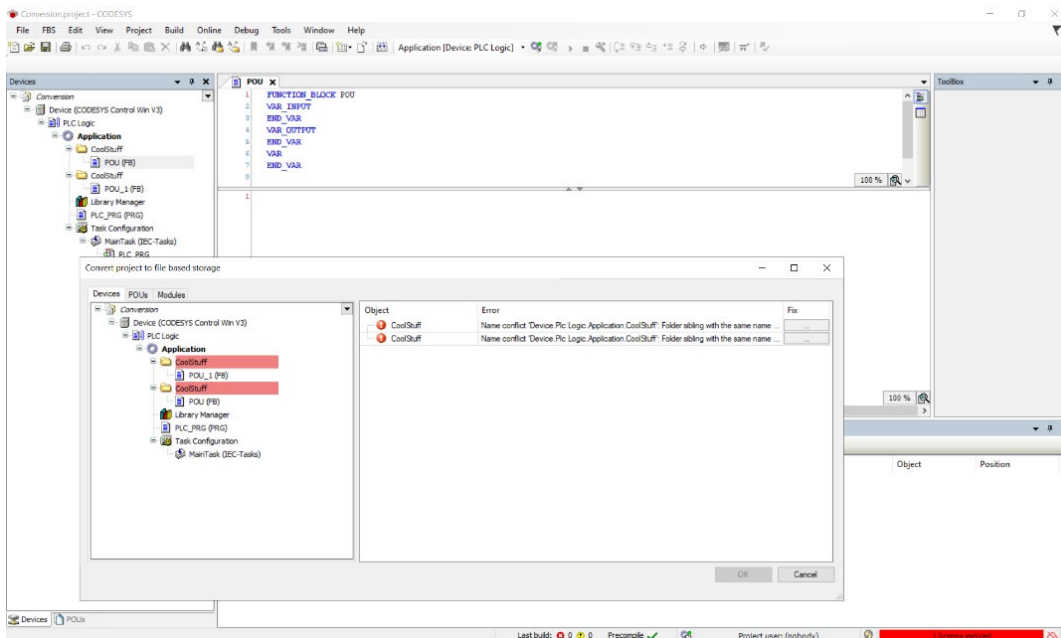
Changes with CODESYS File-Based Storage 1.0.0.0

Within the next version 1.0.0.0, based on CODESYS V3.5 SP 22, the internal format will change: The elimination of the current divider (__DECLARATION__, __METADATA__) will reduce the required content. As a consequence, the files will be smaller and more lightweight.

3 Conversion of existing projects

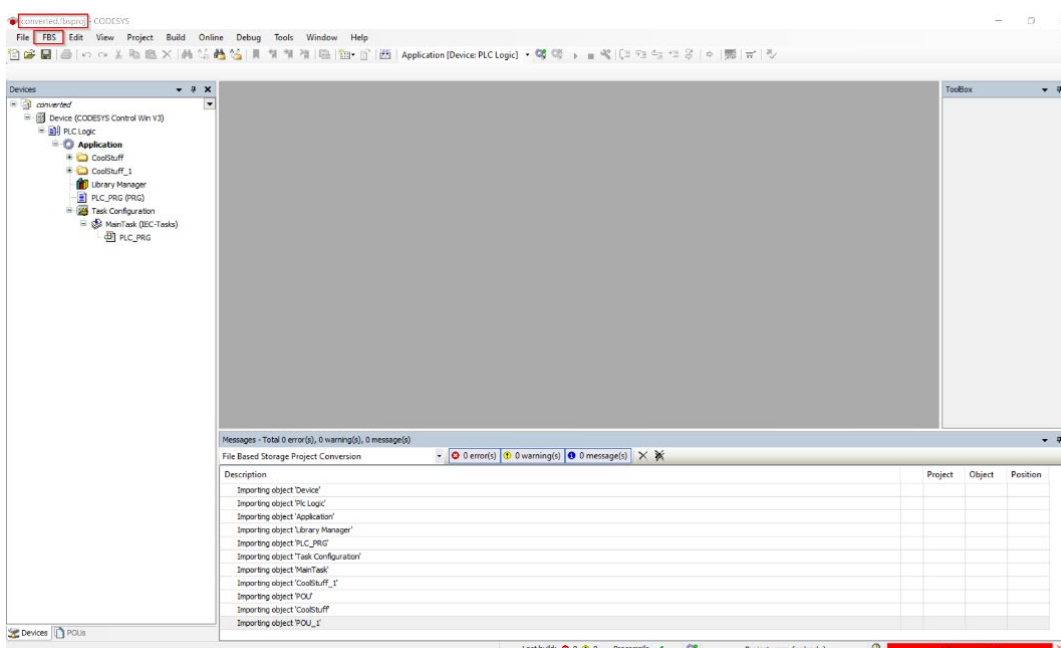
Existing V3 projects can be converted to a file-based project format. During conversion, a check is performed to ensure the conversion can be completed without errors.

The following example demonstrates the conversion of a project consisting of two sibling folders with the same name, that the file system does not support.



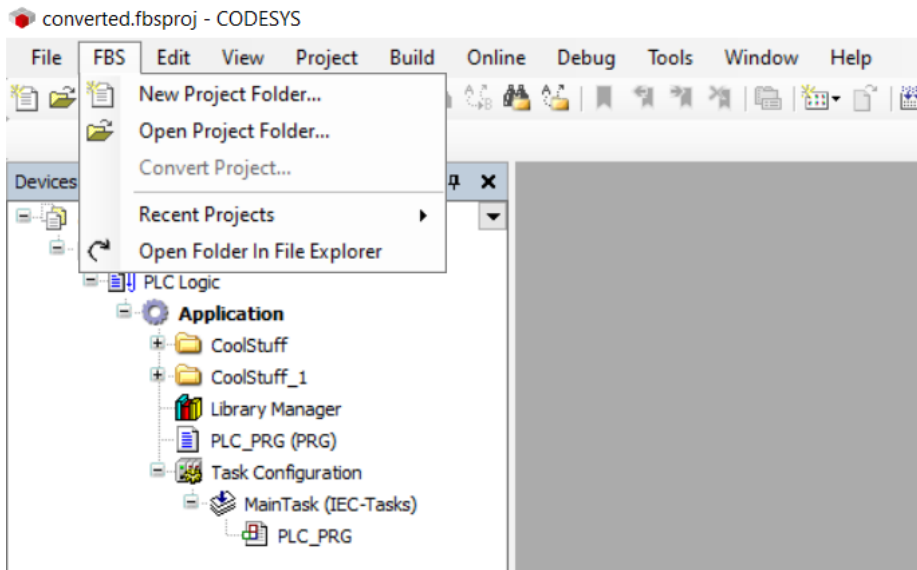
Depending on the error that occurs, quick fixes may also be available. For the above case, one of the affected folders will get a new name.

During conversion, the source project will be closed and reopened as a new file-based project.



The title bar displays the new project extension.

The CODESYS File-Based Storage add-on brings a new set of commands for creating/opening/converting projects. The functionality of the commands will be improved in the next version.



4 Limitations

4.1 User Management

Since CODESYS File-Based Storage stores objects in a readable format, we recommend not to use User Management.

4.2 Project Archive

Creating a project archive on a file-based project does not include the file-based project yet. Instead, a V3 project file will be included into the archive.

4.3 Online Change

A login without online change to an existing, running V3 application with a new file-based project may work, but cannot be guaranteed.

4.4 File system

CODESYS File-Based Storage uses folders and files to store the objects of the project. This structure results in file system limitations that are imposed by the operating system and must be observed.

For example NTFS, the default file system of Windows:

- Maximal length of file name: 255 Unicode characters
- Maximal length of path: 32,760 Unicode characters, each path component consisting of a maximum of 255 characters

CODESYS allows to have more than one folder with the same name at the same location in one project. On the file system, this is not allowed.

External changes

File-Based Storage does not react to external modifications within the file system. After modification, a complete project reload is required.

4.5 Validators

During conversion of classic V3 projects or during daily work on FBS projects, so-called validators prevent invalid and inconsistent data and structures. They block the current or undo the last operation and report warnings/errors during project conversion and loading.

5 Scripting

Create a file-based project via command:

```
# Create "empty" FBS project via command
args = (r"C:\Temp\SomeNewProject.fbsproj")
system.commands["filebasedstorage", "new"].execute(args)
```

Open a file-based project via command:

```
# Open FBS project via command
args = (r"C:\Temp\SomeProject.fbsproj")
system.commands["filebasedstorage", "open"].execute( args )
```

6 License

CODESYS File-Based Storage requires the prior installation of a CODESYS Professional Developer Edition license.

A license-free read-only mechanism is not implemented. **If you wish to share a project with someone who does not have a license installed, you need to convert the File-Based Storage project back to V3 format ("File>Save Project as").**