

# System Requirements Specification

## **(Pflichtenheft)**

(TINF18C, SWE I Praxisprojekt 2019/2020)

*Project:* **DD2AML Converter**

*Customer:* **Rentschler & Ewertz**  
Rotebühlplatz 41  
70178 Stuttgart

*Supplier:* by Lara Mack und Nora Baitinger - Team 3  
(Nora Baitinger, Antonia Wermerskirch, ~~Carl Beese~~, Lara Mack, Bastiane Storz)  
Rotebühlplatz 41  
70178 Stuttgart

Version	Date	Author	Comment
0.1	07.09.2018		created
0.2	01.11.2019	Wermerskirch	added Introduction
0.3	02.11.2019	Mack	added Product Data
0.4	04.11.2019	Baitinger	added Product Requirements
0.5	04.11.2019	Mack	added Non-Functional Requirements
0.6	04.11.2019	Wermerskirch	added Product Environment and Use Cases
0.7	06.03.2020	Storz	Team change / deleted ESI
1.0	24.03.2020	Baitinger	Complete overhaul
1.0	07.05.2020	Baitinger	Final Version

# CONTENTS

<b>1. Introduction .....</b>	<b>3</b>
1.1. Product Environment.....	3
1.2. Use Cases .....	5
1.2.1. <UC.001> File conversion CLI.....	6
1.2.2. <UC.002> File conversion GUI .....	7
1.2.3. <UC.003> Library usage .....	8
<b>2. Product Requirements.....</b>	<b>9</b>
2.1. /LF10/File format analysis .....	9
2.2. /LF20/File validation.....	9
2.3. /LF30/Conversion .....	9
2.4. /LF40/AML Packager .....	10
2.5. /LF50/ Library .....	10
2.6. /LF60/ Command Line Interface .....	11
2.7. /LF70/ Graphical User Interface .....	11
2.8. /LF80/ Error handling .....	12
2.9. /LF90/ Installation wizard .....	12
2.10. /LF100/ CAEX Version .....	12
<b>3. Product Data .....</b>	<b>13</b>
3.1 /LD10/AML-DD .....	13
3.2 /LD20/AMLX Package .....	13
3.3 /LD30/File types.....	13
<b>4. Non-Functional Requirements .....</b>	<b>14</b>
4.1 /NF10/System Enviroment.....	14
4.2 /NF20/Portable Program .....	14
4.3 /NF30/License .....	14
<b>5. Figures .....</b>	<b>15</b>
<b>6. References .....</b>	<b>16</b>
<b>7. Glossar .....</b>	<b>17</b>

# 1. Introduction

The goal of this project is to develop software that supports the conversion of a Fieldbus-specific device description file such as IODD and CSP+ to an AutomationML/CAEX V2.15 file or AutomationML/CAEX V3.0. For this purpose, an already created conversion tool named GSD2AML will be extended with the two mentioned conversion rules. The main function of the software is to convert the given device description (DD) file into the corresponding AML file. For this to be possible, the converter must automatically recognize the respective input format so that it can be converted according to the conversion rules. The software should include a library module that allows the software to be used in other projects to perform the conversions, a command-line program that directly converts the file, and a graphical user interface for better usability that has the same features but is more user-friendly.

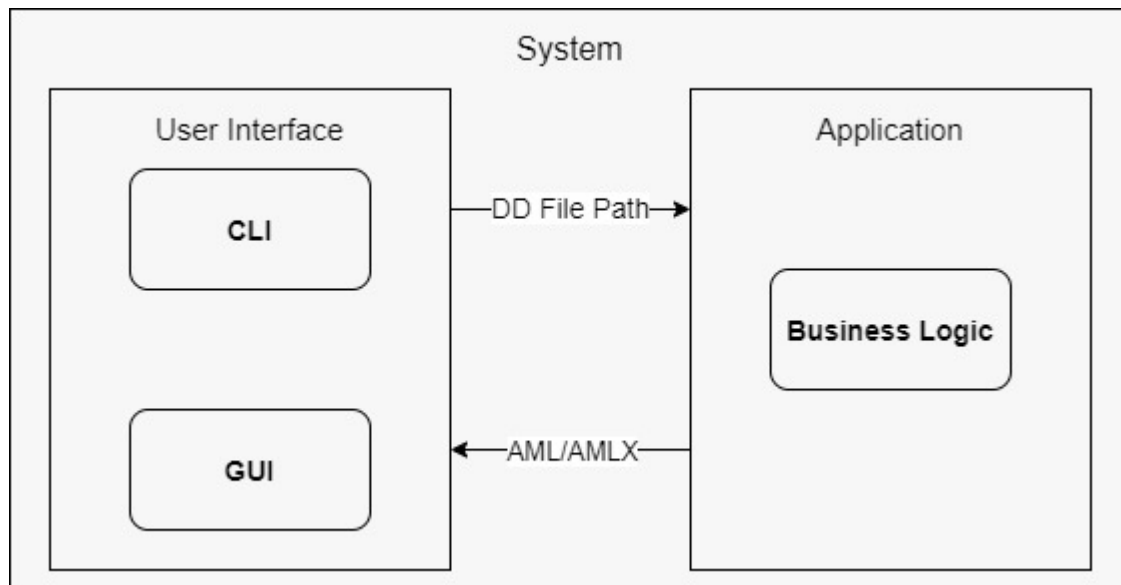
## 1.1. Product Environment

The software is mainly used by manufacturers and other companies who need to manage various device description files. The device description files of the types GSD, IODD and CSP+ are supported, whereby an independent program only for the conversion of GSD files already exists. Now it will be extended by the transformation rules mentioned above.

An IODD (IO Device Description) file describes the sensors and actuators of a system or components. It also contains information about identity, parameters, process data, communication and more. It is written in XML format, like AML, which ensures a conversion. CSP+ (Control & Communication Profile) is used to describe components of the CC-Link Family or those which are compatible with them.

When working with different device description files, it may be useful to normalize them to make it easier to continue working with them. In this case, the files could be converted to AML, an exchange format developed as an open standard. During conversion, all important data and information are read from the device description file and stored in AML. AutomationML (AML) is short for Automation MarkUp Language and is used to describe parts of automation plants as objects. These objects can consist of multiple other objects and be part of a larger assembly of objects. That way AML can be used to describe a single screw or an entire robot with the necessary level of detail. AML makes use of various standards to describe plant components.

1. CAEX (Computer-Aided Engineering Exchange) to describe attributes of objects and their relations in a hierarchical structure. This is called a system topology. In this respect, CAEX forms the overarching integration framework of AutomationML.
2. COLLADA to describe the geometry and 3D models of objects
3. COLLADA also integrates motion planning. It describes the connections and relations of moveable objects, which is called Kinematics
4. PLCopen XML describes the logic. Internal behavior and states of objects, action-sequences and I/O connections are implemented via this format.



*Figure 1- Product Environment*

## 1.2. Use Cases

The software includes a library that can be integrated into other software, a command line interface and a graphical user interface, whereby the library and the program available as CLI or GUI are two independent programs.

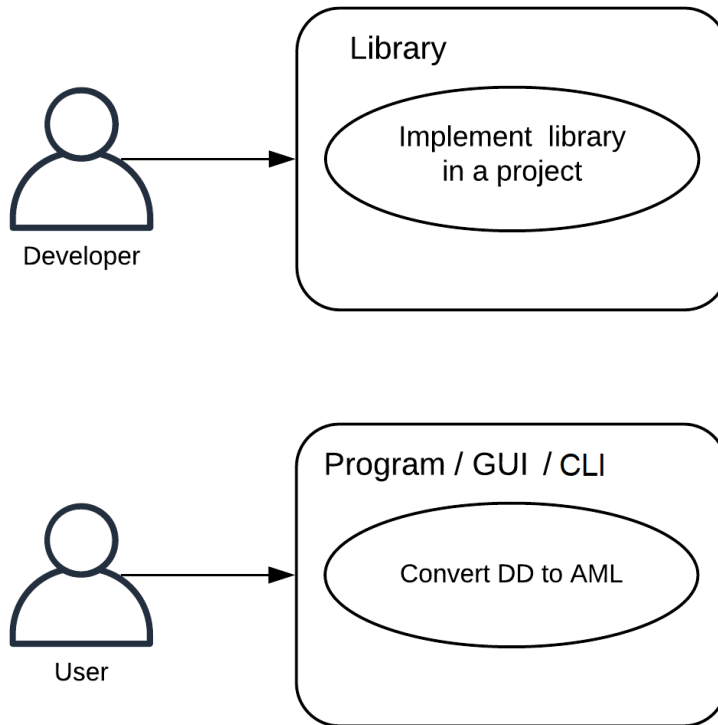


Figure 2 - Use Case Overview Diagram

### 1.2.1. <UC.001> File conversion CLI

<b>Related Business Process:</b>	<BP.001>: File conversion
<b>Use Cases Objective:</b>	User wants to convert his DD file to an AMLX file using the command line interface.
<b>System Boundary:</b>	Program itself is the system boundary.
<b>Precondition:</b>	<ul style="list-style-type: none"> <li>• The program must be installed on the device.</li> <li>• "dd2aml" needs to be registered as an environment variable</li> <li>• The DD file must be without errors.</li> </ul>
<b>Postcondition on success:</b>	The command line interface should not be closed before the conversion is finished.
<b>Involved roles:</b>	User and DD2AML-Converter
<b>Triggering Event:</b>	Typing dd2aml file with a valid FILEPATH in the command line.

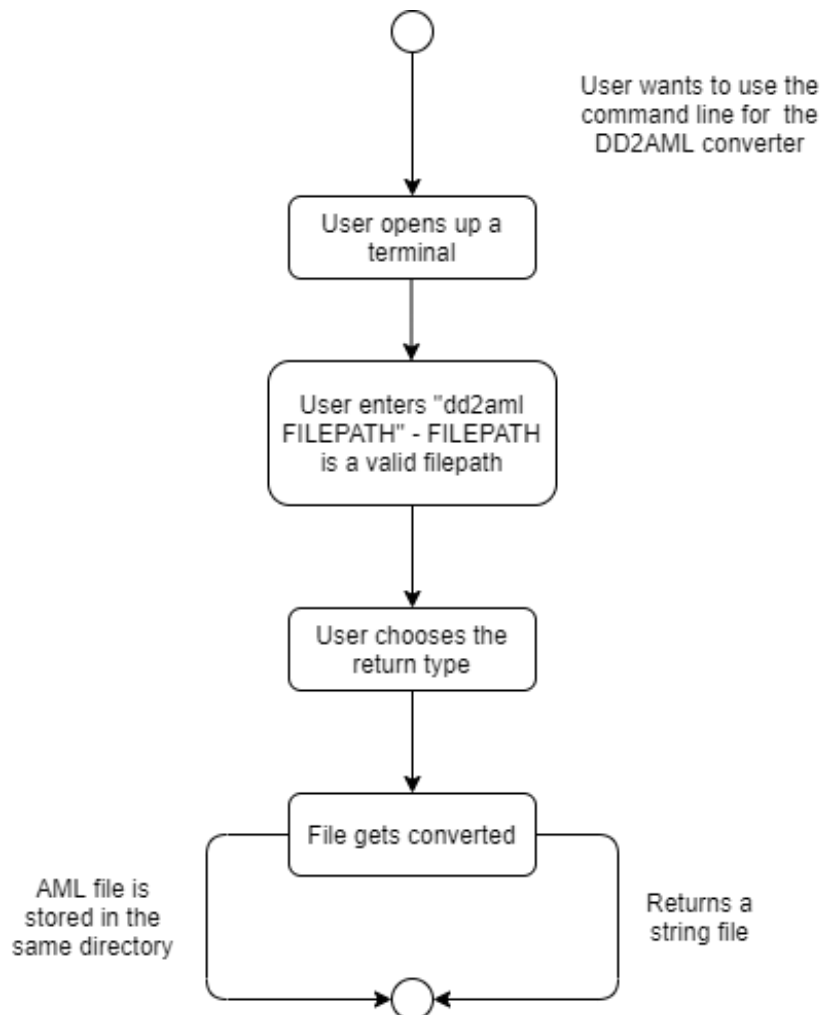


Figure 3 - File conversion CLI

### 1.2.2. <UC.002> File conversion GUI

<b>Related Business Process:</b>	<BP.001>: File conversion
<b>Use Cases Objective:</b>	User wants to convert his DD file to an AMLX file using the GUI.
<b>System Boundary:</b>	Program itself is the system boundary.
<b>Precondition:</b>	<ul style="list-style-type: none"> <li>The program must be installed on the device.</li> <li>The DD file must be without errors.</li> </ul>
<b>Postcondition on success:</b>	The program should not be closed before the conversion is finished.
<b>Involved roles:</b>	User and Converter
<b>Triggering Event:</b>	Program is started.

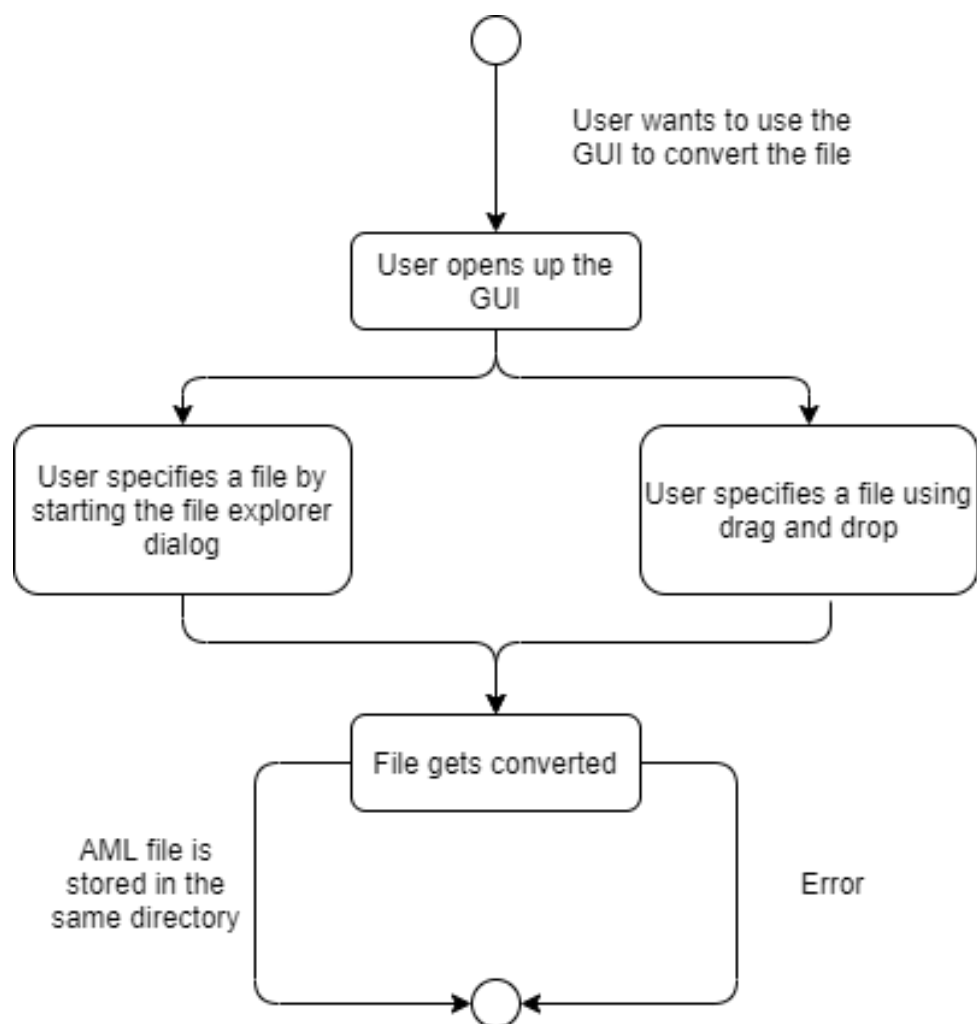


Figure 4 - File conversion GUI

### 1.2.3. <UC.003> Library usage

<b>Related Business Process:</b>	<BP.002>: Library implementation
<b>Use Cases Objective:</b>	Possibility for developers to implement this project in their software.
<b>System Boundary:</b>	Program itself is the system boundary.
<b>Precondition:</b>	The library needs to have a clearly documented interface and must be downloadable from GitHub. It needs to be published under the AGPL-license.
<b>Postcondition on success:</b>	The DD file was successfully converted into an AMLX package.
<b>Involved roles:</b>	Developer and library
<b>Triggering Event:</b>	Developer wants to implement a conversion function to his project.

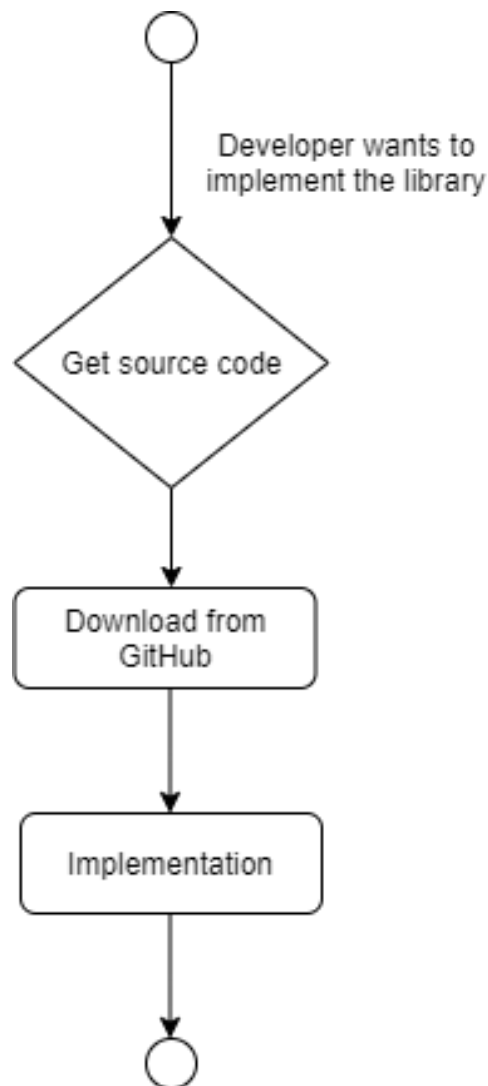


Figure 5 - Library usage



## 2. Product Requirements

The following functionalities shall be supported by the system.

### 2.1. /LF10/File format analysis

Before starting the conversion to AML, it is necessary to import the file the user wants to convert. Therefore, the user selects an IODD or CSP+ file. The component shall check if it is one of the listed formats and recognize it as one of the given formats. In case of violation, the user is asked to import another file. Otherwise, it will continue with the "File validation".

Input field	Value Range
filepath	<i>absolute location of the import file</i>

### 2.2. /LF20/File validation

The „File validation“ shall check if the import file is serializable. In case of a successful serialization, the import file is considered valid and the file shall be checked for essential data regarding the conversion. If such data is missing an error shall be reported. Otherwise, the file will be accepted.

Input field	Value Range
filepath	absolute location of the import file

### 2.3. /LF30/Conversion

The main feature is to convert IODD and CSP+ files into AML-files. Therefore, the "Conversion" shall check the certain features of each file format and convert them into the AML notation. The conversion results are either returned as a string or stored with its dependencies in an AMLX package. This procedure is described in the component "AML Packager".

Input field	Value Range
filepath	absolute location of the import file

## 2.4. /LF40/AML Packager

The “AML Packager” shall collect all files and resources and create the required AMLX. The package will contain the generated AML file and all the corresponding files. The name and the output directory of the AMLX are by default the same as those of the input file.

Input field	Value Range
filepath	absolute location of the import file
outputfilepath (optional)	location where the resulting AMLX package shall be stored
filename (optional)	filename of resulting AMLX package

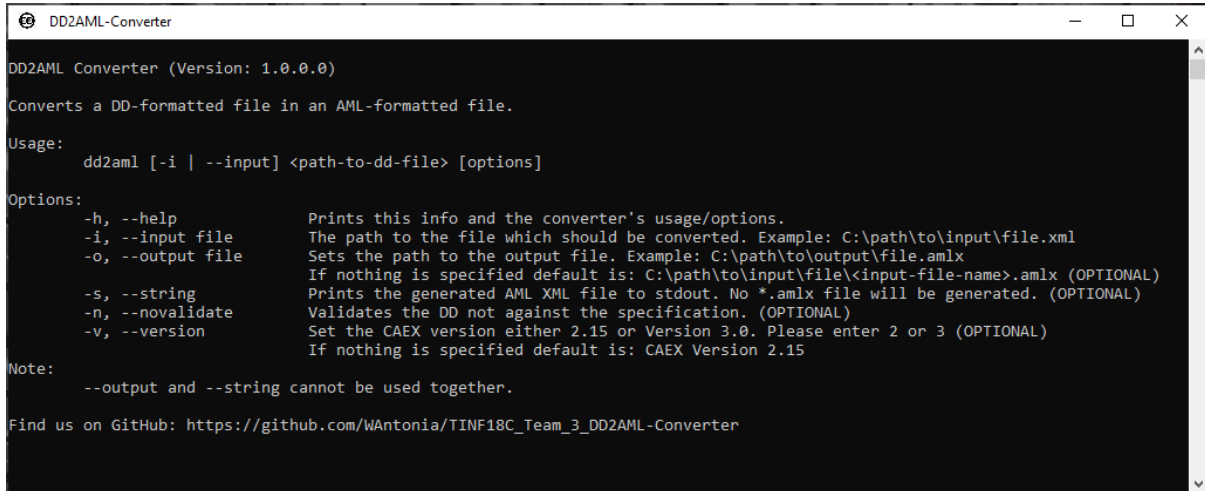
## 2.5. /LF50/ Library

The Library shall be used on its own, so developers can use it in their projects. So after converting the given DD-file, the resulting AML file will be either returned as a string or stored with its dependencies in an AMLX package.

Input field	Value Range
filepath	absolute location of the import file
return type	String or AMLX package

## 2.6. /LF60/ Command Line Interface

The user can start the conversion with the command-line interface to interact with the program. The conversion will use the API of the library. The command-line tool shall provide the requests to the user for the needed inputs and provide error information to the user. The resulting AML file will be either returned as a string or stored with its dependencies in the AMLX package.



```
DD2AML Converter (Version: 1.0.0.0)
Converts a DD-formatted file in an AML-formatted file.

Usage:
  dd2aml [-i | --input] <path-to-dd-file> [options]

Options:
  -h, --help                Prints this info and the converter's usage/options.
  -i, --input file           Sets the path to the file which should be converted. Example: C:\path\to\input\file.xml
  -o, --output file          Sets the path to the output file. Example: C:\path\to\output\file.xml
                             If nothing is specified default is: C:\path\to\input\file\<input-file-name>.amlx (OPTIONAL)
  -s, --string               Prints the generated AML XML file to stdout. No *.amlx file will be generated. (OPTIONAL)
  -n, --novalidate           Validates the DD not against the specification. (OPTIONAL)
  -v, --version              Set the CAEX version either 2.15 or Version 3.0. Please enter 2 or 3 (OPTIONAL)
                             If nothing is specified default is: CAEX Version 2.15

Note:
  --output and --string cannot be used together.

Find us on GitHub: https://github.com/WAntonia/TINF18C\_Team\_3\_DD2AML-Converter
```

Figure 6 - GUI

## 2.7. /LF70/ Graphical User Interface

The second option for the user to interact with the program is the GUI tool. It also uses the API of the library for the conversion. It shall provide the user with the necessary requests to start the conversion and give error information when needed.

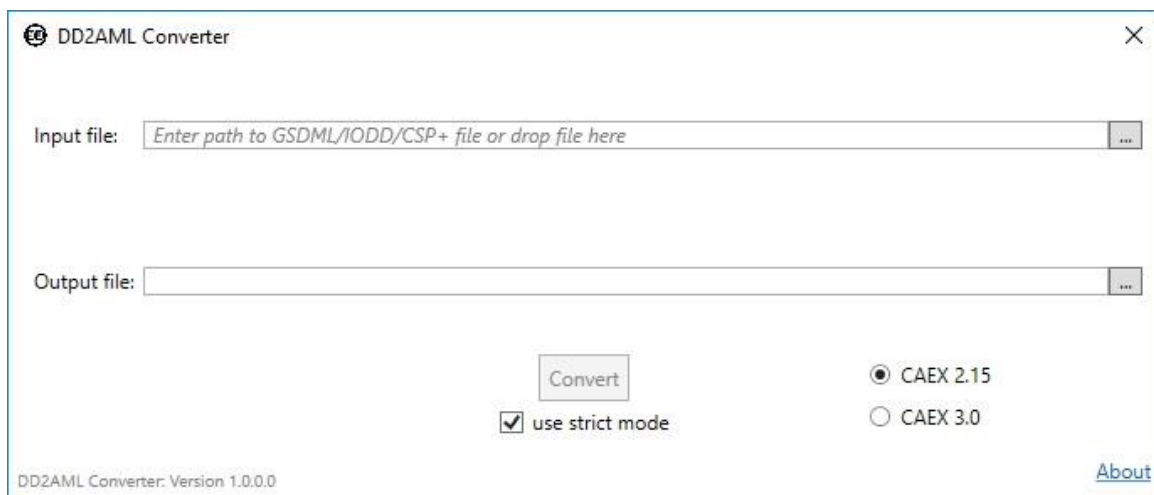


Figure 7 - GUI

## **2.8. /LF80/ Error handling**

The software shall be able to provide the user with understandable error messages. Therefore, an error mechanism is implemented. It should be comprehensible what type of problem occurred, so users and other developers who use the library can handle errors easily.

## **2.9. /LF90/ Installation wizard**

The software shall be installed using an installation wizard (File Conversion GUI).

## **2.10. /LF100/ CAEX Version**

The user shall decide which version of the AutomationML/CAEX he or she wants to have. So the software needs to give the opportunity to choose between the versions 2.15 and 3.0. The conversion into both AML/CAEX versions must be supported by the Library.

### **3. Product Data**

This section describes the initial Logical Data Model from the previously described functionalities and the context level diagram. The data requirements describe the (business) data needed by the application system via its interfaces.

#### **3.1 /LD10/AML-DD**

The system shall create an AML root file with versioning header and a "SystemUnitClass" with the descriptions, identification and interface information of the Profinet Device, also an external Reference to the original file.

#### **3.2 /LD20/AMLX Package**

The system shall store an AMLX file containing the AML root file, the original file and the device picture file.

#### **3.3 /LD30/File types**

The system shall work with a valid GSD, IODD or CSP+ file.

## **4. Non-Functional Requirements**

This section describes the non-functional requirements for the product.

### **4.1 /NF10/System Enviroment**

The software requires Windows 7 or higher. It also requires the Framework to be installed in Version x or higher. The framework is automatically included in the required Windows version.

### **4.2 /NF20/Portable Program**

The software shall exist as a portable program. This can be achieved by packing the necessary files and the runnable program in a ZIP file. During the installation no registry entry is required. So the installation is only performed for the current local user. Basically it is not an installation, but rather the unpacking of the program files on the PC.

### **4.3 /NF30/License**

The software shall be published under the AGPL v3.0 license.[1]

## 5. Figures

Figure 1- Product Environment .....	4
Figure 2 - Use Case Overview Diagram .....	5
Figure 3 - File conversion CLI .....	6
Figure 4 - File conversion GUI .....	7
Figure 5 - Library usage .....	8
Figure 6 - GUI .....	11
Figure 7 - GUI .....	11

## 6. References

[1] [Free Software Foundation: GNU AFFERO GENERAL PUBLIC LICENSE Version 3, 19 November 2007. retrieved 05.11.2019](#)



## 7. Glossar

**AML** Automation Markup Language is an open standard data format for storing and exchanging plant planning data.

**AML DD** AML Device Description

**AMLX** AML Package

**CAEX** Computer-Aided Engineering Exchange

**CLI** Command Line Interface

**CSP+** Control and Communication System Profile

**GUI** Graphical User Interface

**GSD** General-Station-Description

**IODD** Input/Output Device Description

**PN** Profinet

**ZIP** ( *zipper* ) is an archive file format that supports lossless data compression