**System Test Plan**  
(Systemtest Plan)

**(TINF20C, SWE I Praxisprojekt 2021/2022)**

Project: Standalone Modelling Wizard for Devices

Customer: Rentschler & Holder

Rotebühlplatz 41

70178 Stuttgart

Supplier: Team 1  
 Florian Kaiser, Florian Kellermann, Linus Eickhoff, Lukas Ernst, Malte Horst

Rotebühlplatz 41

70178 Stuttgart

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Date** | **Author** | **Comment** |
| 0.1 | 11.03.2022 |  | created |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**Contents**

1. Scope 3

2. Definitions 3

3. Product Names and Attributes 3

4. Features 3

5. Test Preparation Strategy 4

6. Test Execution Strategy 4

7. Test Equipment 5

8. Test Schedule and Budget 5

9. Test Planning 5

10. References / Standards 5

11. Appendix: Testcases 6

11.1. Testsuite <TS-001 Conversion Library> 6

11.1.1. <TC-001-001> (File Validation with valid input file) 6

11.1.2. <TC-001-001> (File Validation with invalid input file) 7

11.2. Testsuite <TS-001 Conversion Library> 8

11.2.1. <TC-002-001> (View CLI help text) 8

11.2.2. <TC-002-002> (Converting without output flag) 9

11.3. Testsuite <TS-003 GUI> 10

11.3.1. <TC-003-001> (GUI Input field verification) 10

11.3.2. <TC-003-002> (GUI Input file selection via file explorer) 10

11.3.3. <TC-003-003> (GUI Input file selection via drag and drop) 11

11.3.4. <TC-003-004> (GUI Output file path generation) 11

# Scope

The STP (System Test Plan) specifies the test strategy and test planning. It references tests to be performed to verify the accordance of the demanded features given by the SRS (System Requirements Specification) to the implemented features. The document derived from the STP is the STR (System Test Report) where additionally the results are given.

# Definitions

**TC** Testcase

**TS** Testsuite

**CLI** Command Line Interface

**GUI** Graphical User Interfac

# Product Names and Attributes

The following test objects must be verified:

|  |  |  |  |
| --- | --- | --- | --- |
| **Ref.-Id.** | **Product Number** | **Product Name** | **Product Description** |
| 1 | Build v1.0 | Standalone Modelling Wizard for Devices GUI | Windows standalone application with a GUI |

# Features

The following requirements must be verified, as long as they are not classified as “not to be tested”. This table shows the test coverage between functionality and test suites or test cases.

|  |  |  |  |
| --- | --- | --- | --- |
| **Req. - ID** | **Functionality** | **Priority** | **Testsuite ID** |
| LF10: Import | Imports file by absolute path | A |  |
| LF20: File validation | Checks whether input file is in a valid format | B |  |
| LF30: Error handling | Application throws errors on expected shutdowns and wrong formatting | B |  |
| LF40: GUI | Draws GUI for user | A |  |
| LF50: Display device in a readable way | Displays loaded device in GUI in a readable way for user | A |  |
| LF60: Edit device | Every attribute of devices should be editable | A |  |
| LF70: Create device | Creates a new and empty device | A |  |
| LF80: Export device | Loaded device is saved as to file | A |  |

# Test Preparation Strategy

The creation of tests will be application case-based. Two main application cases can be identified, the file operations and the GUI.

File operations represent the first main application case. Device files need to be loaded, validated and saved to ensure full functionality of the application for the user.

The GUI is the second main application case. Unlike the previous plugin for the AML Editor, the GUI provides a view of the loaded device with input fields in which the respective device data is displayed. These fields must be checked and features to edit and save device have to be validated.

# Test Execution Strategy

Since it is a re-implementation of an already existing software, a complete test is not necessary, but it is still useful. The test should be divided into the following phases:

1. File operations
2. Graphical User Interface

Since the file operations are needed for the application to work, these have to be tested first.

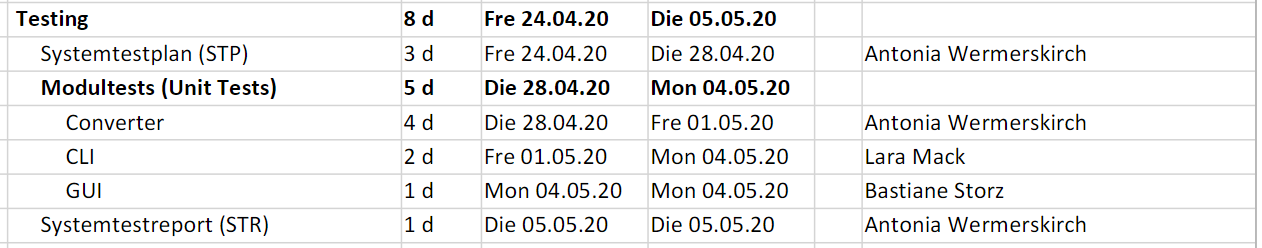
Then the GUI functionality can be tested. This includes the start of the program and the execution of the main features of the application in the GUI.

# Test Equipment

The following equipment must be available for testing:

* A computer with Windows 10 or higher
* The standalone Device Modelling Wizard software

# Test Schedule and Budget



The testing of the CLI begins as soon as the CLI is completed. This makes it possible to make the necessary corrections quickly. The conversion library can only be tested once the rules for one input format, but preferably both input formats, have been established. Since only minimal changes are made in the installer of the GUI, the GUI can be tested as soon as all adjustments intended for the GUI have been made.

No budget is needed for the tests, as they are all performed by hand.

# Test Planning

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Testsuite** | **Test objective** | **Testplan Creator** | **Testplan Reviewer** | **Tester** |
| TS-001 | File operations | Linus Eickhoff | Florian Kaiser |  |
| TS-002 | Graphical User Interface | Linus Eickhoff | Florian Kaiser |  |

# References / Standards

[1] [SRS TINF20C Device Modelling Wizard](https://github.com/H4CK3R-01/TINF20C_ModellingWizard_Devices/wiki/1.-Software-Requirements--Specification)

# Appendix: Testcases

## Testsuite <TS-001 File operations>

### <TC-001-001> (Loading of a valid file with validation)

|  |  |  |  |
| --- | --- | --- | --- |
| **Testcase ID:** | | TC-001-001 | |
| **Testcase Name:** | | Loading of a valid file with validation | |
| **Req.-ID:** | | LF10, LF20, LF30 | |
| **Description:** | | The test case verifies that it recognizes if a valid file has been loaded. | |
| **Test Steps** | | | |
| **Step** | **Action** | | **Expected result** |
| 1 | Install the DD2AML tool and open the CLI by typing cmd in the windows search. | | The DD2AML tool is installed on the system. The CLI is open. |
| 2 | Select a valid input file for the validation, for example:  dd2aml –input /filePathTo/Balluff-BNI\_IOL\_355\_S02\_Z013-20170315-IODD1.1.xml -v 2 | | The validation is executed successfully, and the conversion is completed correctly without error message. |
| 3 | Then open the logs of the CLI. These can be found under:  C:\Users\USERNAME\AppData\Local\DD2AML\Logs\CLI | | After replacing the USERNAME tag with the real username, the CLI folder with all logs opens. The most recent log is opened. |
| 4 | Find the log message that shows that the file was successfully deserialized. It can be found at the beginning of the log file. | | The log message “DD file was deserialized correctly.” should be found approximately in the fourth line of the log. |
|  | | | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Testdata:** | | TD-001-001 | | | |
| **Dataset** | **File** | | **Validation** | **Permission Input** | **Permission Output** |
| 1 | Balluff-BNI\_IOL\_355\_S02\_Z013-20170315-IODD1.1.xml | | valid | given | given |
|  | | | | | |

### <TC-001-002> (Loading of an invalid file with validation)

|  |  |  |  |
| --- | --- | --- | --- |
| **Testcase ID:** | | TC-001-002 | |
| **Testcase Name:** | | Loading of an invalid file with validation | |
| **Req.-ID:** | | LF10, LF20, LF30 | |
| **Description:** | | The test case verifies that errors are detected during the validation of the input file and a corresponding error message is displayed with a description of the error and line details in the log. | |
| **Test Steps** | | | |
| **Step** | **Action** | | **Expected result** |
| 1 | Install the DD2AML tool and open the CLI by typing cmd in the windows search. | | The DD2AML tool is installed on the system. The CLI is open. |
| 2 | Select a valid input file for the validation, for example:  dd2aml –input /filePathTo/BrokenBalluff-BNI\_IOL\_355\_S02\_Z013-20170315-IODD1.1.xml -v 2 | | The conversion is aborted after the failed validation. |
| 3 | Then open the logs of the CLI. These can be found under:  C:\Users\USERNAME\AppData\Local\DD2AML\Logs\CLI | | After replacing the USERNAME tag with the real username, the CLI folder with all logs opens. The most recent log is open. |
| 4 | Look at the first error message in the logs. | | The error message can be found approximately in the 5th line. Detailed information about the error, as well as line details are given. |
|  | | | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Testdata:** | | TD-001-002 | | | |
| **Dataset** | **File** | | **Validation** | **Permission Input** | **Permission Output** |
| 1 | BrokenBalluf-BNI\_IOL\_355\_S02\_Z013-20170315-IODD1.1.xml | | invalid | given | given |
|  | | | | | |

### <TC-001-003> (Export of a valid device to file with validation)

|  |  |  |  |
| --- | --- | --- | --- |
| **Testcase ID:** | | TC-001-003 | |
| **Testcase Name:** | | Export of a valid device to file with validation) | |
| **Req.-ID:** | | LF20, LF30, LF80 | |
| **Description:** | | The test case verifies that a correctly formatted device can be validated and exported to a file | |
| **Test Steps** | | | |
| **Step** | **Action** | | **Expected result** |
| 1 | Install the DD2AML tool and open the CLI by typing cmd in the windows search. | | The DD2AML tool is installed on the system. The CLI is open. |
| 2 | Select a valid input file for the validation, for example:  dd2aml –input /filePathTo/BrokenBalluff-BNI\_IOL\_355\_S02\_Z013-20170315-IODD1.1.xml -v 2 | | The conversion is aborted after the failed validation. |
| 3 | Then open the logs of the CLI. These can be found under:  C:\Users\USERNAME\AppData\Local\DD2AML\Logs\CLI | | After replacing the USERNAME tag with the real username, the CLI folder with all logs opens. The most recent log is open. |
| 4 | Look at the first error message in the logs. | | The error message can be found approximately in the 5th line. Detailed information about the error, as well as line details are given. |
|  | | | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Testdata:** | | TD-001-002 | | | |
| **Dataset** | **File** | | **Validation** | **Permission Input** | **Permission Output** |
| 1 | BrokenBalluf-BNI\_IOL\_355\_S02\_Z013-20170315-IODD1.1.xml | | invalid | given | given |
|  | | | | | |

### <TC-001-004> (Export of an invalid device to file with validation)

|  |  |  |  |
| --- | --- | --- | --- |
| **Testcase ID:** | | TC-001-001 | |
| **Testcase Name:** | | Loading of a valid file with validation | |
| **Req.-ID:** | | LF10, LF20, LF30 | |
| **Description:** | | The test case verifies that errors are detected during the validation of the loaded device and a corresponding error message is displayed with a description of the error and line details in the log. This happens before saving the device as file. | |
| **Test Steps** | | | |
| **Step** | **Action** | | **Expected result** |
| 1 | Install the DD2AML tool and open the CLI by typing cmd in the windows search. | | The DD2AML tool is installed on the system. The CLI is open. |
| 2 | Select a valid input file for the validation, for example:  dd2aml –input /filePathTo/Balluff-BNI\_IOL\_355\_S02\_Z013-20170315-IODD1.1.xml -v 2 | | The validation is executed successfully, and the conversion is completed correctly without error message. |
| 3 | Then open the logs of the CLI. These can be found under:  C:\Users\USERNAME\AppData\Local\DD2AML\Logs\CLI | | After replacing the USERNAME tag with the real username, the CLI folder with all logs opens. The most recent log is opened. |
| 4 | Find the log message that shows that the file was successfully deserialized. It can be found at the beginning of the log file. | | The log message “DD file was deserialized correctly.” should be found approximately in the fourth line of the log. |
|  | | | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Testdata:** | | TD-001-001 | | | |
| **Dataset** | **File** | | **Validation** | **Permission Input** | **Permission Output** |
| 1 | Balluff-BNI\_IOL\_355\_S02\_Z013-20170315-IODD1.1.xml | | valid | given | given |
|  | | | | | |

## Testsuite <TS-003 GUI>

### <TC-003-001> (GUI Input field verification)

|  |  |  |  |
| --- | --- | --- | --- |
| **Testcase ID:** | | TC-003-001 | |
| **Testcase Name:** | | GUI Input field verification | |
| **Req.-ID:** | | LF40, LF60 | |
| **Description:** | | Run converter application with GUI and test the editing of a device. Input fields for data need to be validated. | |
| **Test Steps** | | | |
| **Step** | **Action** | | **Expected result** |
| 1 | Install the DD2AML Software and open the GUI. | | The software is installed and the GUI window opens. |
| 2 | Try to start the conversion by pressing the “Convert” button at the bottom centre. | | Conversion not possible, because "Convert" button stays deactivated. |
|  | | | |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Testdata:** | | TD-003-001 | | | | |
| **Dataset** | **Input File** | | **Validation** | **Permission Input** | **Permission Output** | **Output File** |
|  |  | |  |  |  |  |
|  | | | | | | |

### <TC-003-002> (GUI Load file via file explorer)

|  |  |  |  |
| --- | --- | --- | --- |
| **Testcase ID:** | | TC-003-002 | |
| **Testcase Name:** | | GUI Input file selection via file explorer | |
| **Req.-ID:** | | LF10, LF20, LF40, LF50 | |
| **Description:** | | The test case verifies that only the permitted file formats can be selected as input via file explorer. Afterwards, the device from the file has to be displayed correctly in the GUI.  Permitted file formats: .xml | |
| **Test Steps** | | | |
| **Step** | **Action** | | **Expected result** |
| 1 | Install the DD2AML Software and open the GUI. | | The software is installed and the GUI window opens. |
| 2 | Click on the "..." button at the end of the input text field. | | The file explorer opens in a new window. |
| 3 | Click on "Files" in the lower right corner directly above the buttons for open and cancel | | A drop-down menu opens showing that only file suffix with .xml or .cspp are allowed. |
|  | | | |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Testdata:** | | TD-003-002 | | | | |
| **Dataset** | **Input File** | | **Validation** | **Permission Input** | **Permission Output** | **Output File** |
|  |  | |  |  |  |  |
|  | | | | | | |

### <TC-003-004> (GUI Creation and editing of a new device)

|  |  |  |  |
| --- | --- | --- | --- |
| **Testcase ID:** | | TC-003-004 | |
| **Testcase Name:** | | GUI Creation and editing of a new device | |
| **Req.-ID:** | | LF40, LF60, LF70 | |
| **Description:** | | The test case verifies whether a new, empty device can be created and edited in the editor afterwards. | |
| **Test Steps** | | | |
| **Step** | **Action** | | **Expected result** |
| 1 | Install the DD2AML Software and open the GUI. | | The software is installed and the GUI window opens. |
| 2 | Select a valid file of IODD, CSP+ or GSD format in the Input text box. | | As soon as the file including file path is in the input field, an output file is suggested for the same directory. The output file has the suffix .amlx and does not have the file format of the input file in its name. |
|  | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Testdata:** | | TD-003-004 | |
| **Dataset** | **Input File** | | **Output File** |
| 1 | .\Balluff-BNI\_IOL\_355\_S02\_Z013-20170315-IODD1.1.xml | | .\Balluff-BNI\_IOL\_355\_S02\_Z013-20170315.amlx |
| 2 | . \0x1099\_BNI CIE-508-105-Z015\_2.0\_en.cspp | | .\0x1099\_BNI CIE-508-105-Z015\_2.0\_en.amlx |
| 4 | .\GSDML-V2.33.xml | | .\V2.33.amlx |
|  | | | |

### <TC-003-004> (GUI Export of a loaded device)

|  |  |  |  |
| --- | --- | --- | --- |
| **Testcase ID:** | | TC-003-004 | |
| **Testcase Name:** | | GUI Export of a loaded device | |
| **Req.-ID:** | | LF40, LF80 | |
| **Description:** | | The test case verifies whether a loaded device in the application can be exported and saved as a file. | |
| **Test Steps** | | | |
| **Step** | **Action** | | **Expected result** |
| 1 | Install the DD2AML Software and open the GUI. | | The software is installed and the GUI window opens. |
| 2 | Select a valid file of IODD, CSP+ or GSD format in the Input text box. | | As soon as the file including file path is in the input field, an output file is suggested for the same directory. The output file has the suffix .amlx and does not have the file format of the input file in its name. |
|  | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Testdata:** | | TD-003-004 | |
| **Dataset** | **Input File** | | **Output File** |
| 1 | .\Balluff-BNI\_IOL\_355\_S02\_Z013-20170315-IODD1.1.xml | | .\Balluff-BNI\_IOL\_355\_S02\_Z013-20170315.amlx |
| 2 | . \0x1099\_BNI CIE-508-105-Z015\_2.0\_en.cspp | | .\0x1099\_BNI CIE-508-105-Z015\_2.0\_en.amlx |
| 4 | .\GSDML-V2.33.xml | | .\V2.33.amlx |
|  | | | |