

# Moduldocumentation

(MOD)

(TINF20C, SWE I Praxisprojekt 2021/2022)

## Modul

## Runtime

*Project:* **Modelling Wizard for Devices**

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

*Supplier:* by Florian Kellermann – Team 1  
(Linus Eickhoff, Florian Kellermann, Lukas Ernst, Malte Horst, Florian Kaiser)  
Rotebühlplatz 41  
70178 Stuttgart

Version	Date	Author	Comment
V0.1	06.09.2021	Florian Kellermann	Created
V0.2	29.04.2022	Florian Kellermann	Filling in information
V0.3	01.05.2022	Florian Kellermann	Updated Information
V0.4	03.05.2022	Lukas ernst	Approved Information
V0.5	06.05.2022	Malte Horst	checked
V1.0	06.05.2022	Florian Kellermann	Improved design

# Contents

1.	Scope .....	3
2.	Definitions .....	3
3.	Module Requirements .....	4
3.1.	User View .....	4
3.2.	Requirements .....	4
3.3.	Module Context .....	4
4.	Analysis .....	5
5.	Design .....	5
5.1.	Risks .....	5
6.	Implementation .....	6
7.	Module Test .....	7
7.1.	Module Testplan .....	7
7.2.	Module Testreport .....	7
8.	Summary .....	8
9.	Appendix .....	8
9.1.	References .....	8
9.2.	Code .....	8

## 1. Scope

This module documentation explains the Runtime in further detail. It shows how the standalone application is working in the backend. In this case focusing on how the AML Editor Plugin is now running as standalone. The individual functions are tested in advance and their results are documented here. If there are any existing problems they are also documented here and possible solutions are explained in more detail

It can also serve as a programming guide, if further features should be implemented.

## 2. Definitions

**SRS** - System Requirement Specification

**GUI** – Graphical User Interface

**STP** - System Test Plan

**STR** - System Test Report

**AMLX** - AML Package

**CAEX** - Computer Aided Engineering Exchange

**URI** - Uniform Resource Identifier

## 3. Module Requirements

### 3.1. User View

This Module should provide the user the following features:

1. No longer has to install AML editor
2. Faster process of installation
3. Therefore better usability

### 3.2. Requirements

The following requirements are implemented by this module: /LF30/, /LF40/, /LF50/ and /LF60/

**/LF30/:** By using a standalone application the error handling shall be improved and therefore changed.

**/LF50/:** Display device in a readable way when loading the application.

**/LF60/:** edit device in the application

**Goal:** create a standalone application from the existing plugin for the AutomationML Editor.

### 3.3. Module Context

This module provides the backend of the Standalone application. It is responsible for making the program standalone. There are many benefits from stopping the use of the AutomationML Editor and turning the plugin into an executable.

## 4. Analysis

The main task is to run the Device Modelling Plugin in a standalone application to no longer depend on the use of the Automation ML editor. With the past version of this software you had to install Automation ML, set it up and only then were you able to import the Device Modelling plugin into the application. The plugin therefore was relying on the Automation ML Engine to read, correctly interpret and show the functionality of the plugin. The goal is to create **one** executable to solve this problem.

## 5. Design

There are three important aspects when doing so:

- 1.: The programs GUI has to be working the same way as before. Certain functionalities need to be found at the same exact locations so users from the past will also be happily using the new software. So only small usability changes are to be expected here.
- 2.: The windows forms application shall not create new errors the Automation ML Editor didn't have before. The errors from before will remain and get fixed but its not the goal to make the use of the program less efficient because of it being standalone.
- 3.: The program must be run in one executable to improve usability

### 5.1. Risks

Creating new problems and errors because the plugin might be relying on a certain functionality of the Automation ML editor must be avoided.

## 6. Implementation

To make sure everything will be running smoothly the windows forms application is importing the Automation ML Engine and then automatically importing the pre-existing plugin (with a few changes). So the actual software interpreting the .dll file will be the same as before but it will no longer need the Automation ML editor and you no longer manually will have to import the plugin. When starting the program the windows forms application will load the Automation ML Engine and automatically apply the plugin.

## 7. Module Test

In this section nearly all requirements will be tested separately on their functionality.

### 7.1. Module Testplan

Req. -ID	Functionality	Priority	Testsuite ID
<b>LF30: Error Handling</b>	Application throws errors on expected shutdowns and wrong formatting	B	TS-002
<b>LF50: Display device in a readable way</b>	Displays loaded device in GUI in a readable way for user	A	TS-002
<b>LF60: Edit device</b>	Every attribute of devices should be editable	A	Ts-002

### 7.2. Module Testreport

Req. -ID	Pass/Fail	When failed: Observation	Reviewer
<b>LF30: Error Handling</b>	Pass		Linus Eickhoff
<b>LF50: Display device in a readable way</b>	Pass		Linus Eickhoff
<b>LF60: Edit device</b>	Pass		Malte Horst

## 8. Summary

All of the module requirements have been achieved. The plugin is now imported automatically into a windows forms application that can run on windows. The plugin is then imported and executed based on the AML Engine. By doing it this way the AutomationML Editor is no longer in use and the user only has to run one executable for the program to start and be usable.

## 9. Appendix

### 9.1. References

- [1] System Requirements Specification: [https://github.com/H4CK3R-01/TINF20C\\_ModellingWizard\\_Devices/wiki/1.-Software-Requirements--Specification](https://github.com/H4CK3R-01/TINF20C_ModellingWizard_Devices/wiki/1.-Software-Requirements--Specification)
- [2] Previous Project: <https://github.com/DekaAthlos/TINF19C-ModellingWizard>
- [3] System Test Plan: [https://github.com/H4CK3R-01/TINF20C\\_ModellingWizard\\_Devices/blob/47d2ba67fc73ebc080f303f0e29ca2260d8c7d88/PROJECT/STP/TINF20C\\_STP\\_Team\\_1.pdf](https://github.com/H4CK3R-01/TINF20C_ModellingWizard_Devices/blob/47d2ba67fc73ebc080f303f0e29ca2260d8c7d88/PROJECT/STP/TINF20C_STP_Team_1.pdf)
- [4] System Test Report: [https://github.com/H4CK3R-01/TINF20C\\_ModellingWizard\\_Devices/blob/47d2ba67fc73ebc080f303f0e29ca2260d8c7d88/PROJECT/STR/TINF20C\\_STR\\_Team\\_1.pdf](https://github.com/H4CK3R-01/TINF20C_ModellingWizard_Devices/blob/47d2ba67fc73ebc080f303f0e29ca2260d8c7d88/PROJECT/STR/TINF20C_STR_Team_1.pdf)

### 9.2. Code

The source code for this module can be found at:

- [https://github.com/H4CK3R-01/TINF20C\\_ModellingWizard\\_Devices/blob/app-source-code/SOURCE/Plugin/MWData.cs](https://github.com/H4CK3R-01/TINF20C_ModellingWizard_Devices/blob/app-source-code/SOURCE/Plugin/MWData.cs)
- [https://github.com/H4CK3R-01/TINF20C\\_ModellingWizard\\_Devices/blob/app-source-code/SOURCE/Plugin/DeviceDescription.cs](https://github.com/H4CK3R-01/TINF20C_ModellingWizard_Devices/blob/app-source-code/SOURCE/Plugin/DeviceDescription.cs)