

# PMS.Cliptest - Manual

Version 23.3.1.0



MAR Rostock GmbH - 09.08.2023

© 2023 PMS.Cliptest is a registered Trademark of MAR Rostock GmbH. Microsoft, SQL Server and Windows 10 are registered trademarks of the Microsoft Corporation. All other trademarks are in legal ownership of their registered proprietors.

## Software version management

| Version | Date       | Status  | Range of Validity | Creator           | Note   |
|---------|------------|---------|-------------------|-------------------|--|
| 1.0.0.0 | 17.05.2016 | Release | MAR / Global      | Schuster / Seidel | Software release V1.0.0.0  |
| 1.0.1.0 | 31.05.2016 | Release | MAR / Global      | Schuster / Seidel | Small changes, for details see changelog                                   |
| 1.1.0.1 | 18.10.2016 | Release | MAR / Global      | Schuster / Seidel | Improve MasterDataManager user experience, Add Production Server Interface |
| 1.1.1.0 | 24.10.2016 | Release | MAR / Global      | Schuster / Seidel | Add SAP Import to MasterDataManager  |

|          |            |         |              |                   |   |
|----------|------------|---------|--------------|-------------------|---|
| 1.2.0.0  | 04.11.2016 | Release | MAR / Global | Schuster / Seidel | Add Backup feature  |
| 1.2.9.0  | 12.05.2017 | Release | MAR / Global | Schuster / Seidel | Refactoring Scanner Interface   |
| 1.3.0.0  | 01.06.2017 | Release | MAR / Global | Schuster / Seidel | Adding module list scan   |
| 2.0.0.0  | 21.05.2019 | Release | MAR / Global | Seidel            | Currently in release process  |
| 2.1.0.0  | 19.07.2016 | Release | MAR / Global | Schuster / Seidel | Adding Clip Groups, Extended operator header, Localization  |
| 2.2.2.0  | 15.01.2020 | Release | MAR / Global | Schuster / Seidel | Improve user experience   |
| 20.1.1.0 | 11.03.2020 | Release | MAR / Global | Schuster / Seidel | Different colors for cliptest and emptytest, New backends, performance and usability improvements                         |
| 20.2.1.0 | 06.05.2020 | Release | MAR / Global | Schuster          | PMS.Cliptest.Eventing extension   |
| 20.3.1.0 | 14.07.2020 | Release | MAR / Global | Schuster / Seidel | Added PMS.Ret device for Eventing. Communication is based on CAN Bus, Switch Outputs and receive Input Signals            |
| 20.3.2.0 | 26.08.2020 | Release | MAR / Global | Schuster / Seidel | Improved PMS.Ret eventing device  |
| 20.4.1.4 | 18.12.2020 | Release | MAR / Global | Schuster / Seidel | Minor Bugfixes  |
| 21.1.1.0 | 14.01.2021 | Release | MAR / Global | Schuster / Seidel | Added Opc.Ua.Client device for Eventing, Added BrainBox device, Improve Eventing configuration, Added Eventing Conditions |
| 21.1.1.1 | 15.04.2021 | Release | MAR / Global | Schuster / Seidel | Improve configuration of statistical data   |
| 22.3.1.0 | 29.09.2022 | Release | MAR / Global | Schuster / Seidel | Added Dashboard-Feature, CsvFileBackend, Changed color scheme, Removed name 2016  |
| 22.4.1.0 | 03.11.2022 | Release | MAR / Global | Schuster / Seidel | Redesign with Hamburger Menu. Preparation for newer .Net (Core) Frameworks. Fixed "Keep" option for Scans.                |
| 22.4.2.0 |            | Release | MAR / Global | Schuster / Seidel |   |

|          |            |         |                 |                      |   |
|----------|------------|---------|-----------------|----------------------|---|
| 23.2.1.0 | 12.05.2023 | Release | MAR /<br>Global | Schuster /<br>Seidel | Complete technology-switch to .Net 7, first (draft)version of Web-Interface, new Eventing Device HellermannTytonCpk |
| 23.2.2.0 | 02.06.2023 | Release | MAR /<br>Global | Schuster /<br>Seidel | Several Bugfixes  |
| 23.3.1.0 | 11.07.2023 | Release | MAR /<br>Global | Schuster /<br>Seidel | Bugfixes in multiple production interfaces  |

## Document version management

| Version | Date       | Status  | Range of<br>Validity | Creator              | Document   |
|---------|------------|---------|----------------------|----------------------|--|
| 1.0.0.0 | 17.05.2016 | Release | MAR /<br>Global      | Schuster /<br>Seidel | Publishing of Document V1.0.0.0  |
| 1.0.1.0 | 31.05.2016 | Release | MAR /<br>Global      | Schuster             | Documentation for menu option Manage Master Data added                       |
| 1.0.2.0 | 21.07.2016 | Release | MAR /<br>Global      | Schuster             | Dongle Licensing modified  |
| 1.0.2.1 | 24.08.2016 | Release | MAR /<br>Global      | Schuster             | Hardware requirements modified   |
| 1.1.0.1 | 18.10.2016 | Release | MAR /<br>Global      | Schuster             | MasterDataManager and ServiceConfigurator changes                            |
| 1.1.1.0 | 24.10.2016 | Release | MAR /<br>Global      | Schuster             | MasterDataManager SAP Import   |
| 1.2.0.0 | 04.11.2016 | Release | MAR /<br>Global      | Schuster /<br>Seidel | Add Backup feature   |
| 1.2.9.0 | 12.05.2017 | Release | MAR /<br>Global      | Schuster             | Refactoring Scanner Interface  |
| 1.3.0.0 | 01.06.2017 | Release | MAR /<br>Global      | Schuster             | Adding module list scan  |
| 1.3.0.1 | 06.06.2017 | Release | MAR /<br>Global      | Schuster             | Adding version upgrade instructions  |
| 1.3.0.2 | 15.06.2017 | Release | MAR /<br>Global      | Schuster             | MasterData – PinStatus file add hint for LED number and address combinations |
| 1.3.0.3 | 21.01.2019 | Release | MAR /<br>Global      | Seidel               | Refactored references according to production interfaces                     |
| 1.4.0.0 | 27.02.2019 | Release | MAR /<br>Global      | Seidel               | Added chapter dialogs, refreshed screenshots                                 |

|          |            |         |              |                   |  |
|----------|------------|---------|--------------|-------------------|--|
| 2.1.0.0  | 19.07.2019 | Release | MAR / Global | Schuster          | Added chapter clip groups and localization   |
| 2.2.2.0  | 15.01.2020 | Release | MAR / Global | Schuster          | Improved user experience   |
| 2.2.3.0  | 10.03.2020 | Release | MAR / Global | Seidel            | Added chapter Label confirmation and dialog section printing confirmation            |
| 20.1.1.0 | 11.03.2020 | Release | MAR / Global | Schuster          | Different colors for cliptest and emptytest, New backends, extend printing interface |
| 20.2.1.0 | 06.05.2020 | Release | MAR / Global | Schuster / Seidel | PMS.Cliptest.Eventing extension  |
| 20.3.1.0 | 14.07.2020 | Release | MAR / Global | Schuster / Seidel | Added chapter 6.5 PMS.Ret eventing configuration                                     |
| 20.3.2.0 | 26.08.2020 | Release | MAR / Global | Schuster / Seidel | Refreshed chapter 6.5  |
| 21.1.1.0 | 14.01.2020 | Release | MAR / Global | Schuster / Seidel | Added Opc.Ua.Client device, Added BrainBox device                                    |
| 21.1.1.1 | 15.04.2021 | Release | MAR / Global | Schuster / Seidel | Improve configuration of statistical data  |
| 22.3.1.0 | 29.09.2022 | Release | MAR / Global | Schuster / Seidel | Added Dashboard-Feature, CsvFileBackend, Changed color scheme, Removed name 2016     |
| 22.4.1.0 | 03.11.2022 | Release | MAR / Global | Schuster / Seidel | Redesign with Hamburger Menu. Preparation for newer .Net (Core) Frameworks.          |
| 22.4.2.0 |            | Release | MAR / Global | Schuster / Seidel |  |
| 23.2.1.0 | 12.05.2023 | Release | MAR / Global | Schuster / Seidel | Added chapter 5.4 PMS.Cliptest.Web   |
| 23.2.2.0 | 02.06.2023 | Release | MAR / Global | Schuster / Seidel | Added HellermannTyton CPK eventing device (cable tie plier)                          |
| 23.3.1.0 | 11.07.2023 | Release | MAR / Global | Schuster / Seidel | Added chapter 9.1.2 Scan Data and 5.1.6 Statistic Files Structure                    |

## Referenced Documents

| ID | Document                                      |
|----|---|
| 1  | PMS_PrintProcessor_Documentation_v1.5.3.0.pdf |

## Table of Content

- [PMS.Cliptest - Manual](#)
  - [Software version management](#)
  - [Document version management](#)
  - [Referenced Documents](#)
  - [Table of Content](#)
  - [1 Preamble](#)
    - [1.1 License Royalty](#)
    - [1.2 License Activation](#)
      - [1.2.1 License Key](#)
      - [1.2.2 USB Dongle](#)
      - [1.2.4 PMS.Cliptest DEMO](#)
  - [1.3 Designated use of PMS.Cliptest](#)
    - [1.4 System structure](#)
    - [1.5 Program Modes](#)
      - [1.5.1 Without Production Server](#)
      - [1.5.2 With Production Server](#)
  - [2 Program Runtime Requirements](#)
    - [2.1 Hardware Requirements](#)
      - [2.1.1 PMS.Cliptest.Service](#)
      - [2.1.2 PMS.Cliptest.ViewClient](#)
      - [2.1.3 PMS.Cliptest.MasterDataManager](#)
    - [2.2 Software Requirements](#)
  - [3 Installation guide](#)
    - [3.3 Installing PMS.Cliptest](#)
      - [3.3.1 Deployment options](#)
        - [3.3.1.1 Classic use case](#)
        - [3.3.1.2 Standalone use case](#)
      - [3.3.2 Setup Steps](#)
    - [3.4 Upgrading PMS.Cliptest](#)
  - [4 Program Master Data Requirements](#)
    - [4.1 PinStatus file](#)
      - [4.1.1 Clip Groups](#)
    - [4.2 Coordinates file](#)
    - [4.3 Variants file](#)
    - [4.4 ClipList file](#)
    - [4.5 Modules files](#)
  - [5 User Interface Overview](#)
    - [5.1 PMS.Cliptest.ViewClient](#)
      - [5.1.1 Menu](#)
      - [5.1.2 Overview View](#)
      - [5.1.3 Settings View](#)
        - [5.1.3.1 General](#)
        - [5.1.3.2 Processing](#)

- [5.1.3.2.1 Workplaces to act for / Name by Facility](#)
      - [5.1.3.2.2 Test Parameter](#)
      - [5.1.3.2.3 Test Processing Settings](#)
      - [5.1.3.2.4 Statistic Data](#)
    - [5.1.3.3 Operators](#)
      - [5.1.3.3.1 Manage Operators](#)
    - [5.1.3.4 Scanners](#)
      - [5.1.3.4.1 COM \(RS232\) Scanner](#)
      - [5.1.3.4.2 USB Scanner](#)
    - [5.1.3.5 Scancodes](#)
    - [5.1.3.6 Monitor](#)
      - [5.1.3.6.1 Header With Layout](#)
      - [5.1.3.6.2 Dashboard](#)
    - [5.1.3.7 Display](#)
      - [5.1.3.7.1 Style Configuration](#)
      - [5.1.3.7.2 Test Panel Configuration](#)
      - [5.1.3.7.3 Desktop alerts](#)
    - [5.1.3.8 Printing](#)
  - [5.1.4 Test Panel](#)
    - [5.1.4.1 List layout](#)
    - [5.1.4.2 Image layout](#)
    - [5.1.4.3 Matrix layout](#)
  - [5.1.5 Dialogs](#)
    - [5.1.5.1 Quality Confirmation](#)
    - [5.1.5.2 Label confirmation](#)
  - [5.1.6 Statistic Files Structure](#)
    - [5.1.6.1 Xml File](#)
    - [5.1.6.2 Csv File](#)
- [5.2 PMS.ClipTest.MasterDataManager](#)
  - [5.2.1 Menu](#)
  - [5.2.2 Overview View](#)
    - [5.2.2.1 Board operations](#)
  - [5.2.3 Settings View](#)
    - [5.2.3.1 General](#)
    - [5.2.3.2 File Paths](#)
    - [5.2.3.3 CSV Settings](#)
    - [5.2.3.4 Teaching](#)
    - [5.2.3.5 Scanner](#)
  - [5.2.4 Manage Master Data](#)
    - [5.2.4.1 Backup](#)
      - [5.2.4.1.1 Automatic Backup](#)
  - [5.2.5 Import Master Data](#)
  - [5.2.6 SAP Import](#)
    - [5.2.6.1 Import files](#)
      - [ZCrea file](#)
      - [ZCrea Module file](#)

- [OGC Master Data file](#)
    - [5.2.6.2 Workflow](#)
    - [5.2.6.3 Traceability data export](#)
    - [5.2.7 Module Management](#)
    - [5.2.8 Clip Exchange View](#)
    - [5.2.9 Board Teaching View](#)
    - [5.2.10 Board Designer View](#)
      - [5.2.10.1 Clip Positioning](#)
      - [5.2.10.2 Matrix Configuration](#)
  - [5.3 PMS.ClipTest.ServiceConfigurator](#)
    - [5.3.1 ServiceConfigurator options](#)
    - [5.3.2 Configuration](#)
      - [5.3.2.1 General](#)
      - [5.3.2.2 Backend](#)
        - [5.3.2.2.1 SQL Backend](#)
        - [5.3.2.2.2 File Backend / Demo Backend](#)
      - [5.3.2.3 Production Server](#)
      - [5.3.2.4 Receivers](#)
      - [5.3.2.5 Scan Routing](#)
        - [5.3.2.5.1 Example Configurations](#)
      - [5.3.2.6 Printing](#)
      - [5.3.2.7 Eventing](#)
        - [5.3.2.7.1 Device Context Menu](#)
      - [5.3.2.8 Licensing](#)
  - [5.4 PMS.ClipTest.Web](#)
    - [5.4.1 Requirements](#)
    - [5.4.2 Options](#)
      - [5.4.2.1 Status](#)
      - [5.4.2.2 Logs](#)
      - [5.4.2.3 Reporting](#)
        - [5.4.2.3.1 Test Statistic](#)
        - [5.4.2.3.2 Ranking of clip events](#)
      - [5.4.2.4 Tests](#)
      - [5.4.2.5 Grid Layout](#)
      - [5.4.2.6 Image Layout](#)
      - [5.4.2.7 Printing](#)
      - [5.4.2.8 Receiver](#)
      - [5.4.2.9 Eventing](#)
      - [5.4.2.10 Counters](#)
      - [5.4.2.11 Administration](#)
      - [5.4.2.12 Login/Logout](#)
- [6 PMS.ClipTest.Eventing](#)
  - [6.1 Data Types](#)
  - [6.2 ClipTest](#)
    - [6.2.1 Events](#)
      - [6.2.1.1 Application Started](#)

- [6.2.1.2 Clip Received](#)
  - [6.2.1.3 Cliptest Finished](#)
  - [6.2.1.4 Cliptest Started](#)
  - [6.2.1.5 Emptytest Finished](#)
  - [6.2.1.6 Emptytest Started](#)
  - [6.2.1.7 Http Message Received](#)
  - [6.2.1.8 Scan Received](#)
  - [6.2.1.9 Service Status Updated](#)
  - [6.2.1.10 Test Changed](#)
- [6.2.2 Actions](#)
  - [6.2.2.1 Change Clip Status](#)
  - [6.2.2.2 Handle Scan](#)
  - [6.2.2.3 Hide Message](#)
  - [6.2.2.4 Import Master Data](#)
  - [6.2.2.5 Log Event data](#)
  - [6.2.2.6 Print](#)
  - [6.2.2.7 Request Service Status](#)
  - [6.2.2.8 Reset Counter](#)
  - [6.2.2.9 Show Message](#)
  - [6.2.2.10 Start Cliptest](#)
  - [6.2.2.11 Start Emptytest](#)
  - [6.2.2.12 Stop Test](#)
- [6.2.3 Conditions](#)
  - [6.2.3.1 Is Test Running For Board](#)
  - [6.2.3.1 Is Test not Skipped](#)
- [6.3 Timer](#)
  - [6.3.1 Events](#)
    - [6.3.1.1 Scheduler](#)
    - [6.3.1.2 Timer Elapsed](#)
  - [6.3.2 Actions](#)
    - [6.3.2.1 Start Timer](#)
    - [6.3.2.2 Stop Timer](#)
- [6.4 Audio Player](#)
  - [6.4.1 Actions](#)
    - [6.4.1.1 Play File](#)
    - [6.4.1.2 Play System Sound](#)
    - [6.4.1.3 Text To Speech\\*](#)
- [6.5 PMS.Ret](#)
  - [6.5.1 Events](#)
    - [6.5.1.1 Clip Received](#)
  - [6.5.2 Actions](#)
    - [6.5.2.1 Set Output](#)
- [6.6 Opc Ua Client](#)
  - [6.6.1 Opc Ua Server Browser](#)
  - [6.6.2 Events](#)
    - [6.6.2.1 Item Changed](#)



- [6.6.2.2 Item Read](#)
    - [6.6.3 Actions](#)
      - [6.6.3.1 Read Value](#)
      - [6.6.3.2 Write Value](#)
  - [6.7 Brain Box](#)
    - [6.7.1 Events](#)
      - [6.7.1.1 Input Changed](#)
    - [6.7.2 Actions](#)
      - [6.7.2.1 Pulse Output](#)
      - [6.7.2.2 Set Output](#)
  - [6.8 HellermannTyton Cpk](#)
    - [6.8.1 Events](#)
      - [6.8.1.1 Binding\\_Clip\\_Received](#)
      - [6.8.1.2 Binding\\_Faulted](#)
      - [6.8.1.3 Binding\\_Too\\_Much](#)
      - [6.8.1.4 Data Received](#)
    - [6.8.2 Actions](#)
      - [6.8.2.1 Set Binding\\_Clips](#)
      - [6.8.2.2 Reset Binding\\_Clips](#)
      - [6.8.2.3 Send Command](#)
- [7 First Program Start](#)
  - [7.1 Service Configuration](#)
  - [7.2 Master Data Import](#)
  - [7.3 ViewClient](#)
- [8 Troubleshooting](#)
- [9 Appendix I - Printing Interface](#)
  - [9.1 General Data](#)
    - [9.1.1 Test Data](#)
    - [9.1.2 Scan Data](#)
    - [9.1.3 Status Detail Data](#)
  - [9.2 List Data](#)
    - [9.2.1 Variant Parameters](#)
    - [9.2.2 Scans](#)
    - [9.2.3 Critical Events](#)
    - [9.2.4 Clips](#)
- [10 Appendix II - Available Clip States](#)
  - [Clip Test](#)
  - [Empty Test](#)
- [11 Appendix III – Localization](#)
  - [11.1 Adding a new language](#)
  - [11.2 Xliff tools](#)
- [12 Appendix IIII – RegEx Examples](#)

## 1 Preamble

## 1.1 License Royalty

PMS is a registered trademark of MAR Rostock GmbH. A license must be purchased for one to use the full functionality of **PMS.Cliptest**. Fill out the registration document and send it to your local distributor to obtain a license.


## 1.2 License Activation

There are two possibilities for adding new licenses to the **PMS.Cliptest.Service**.

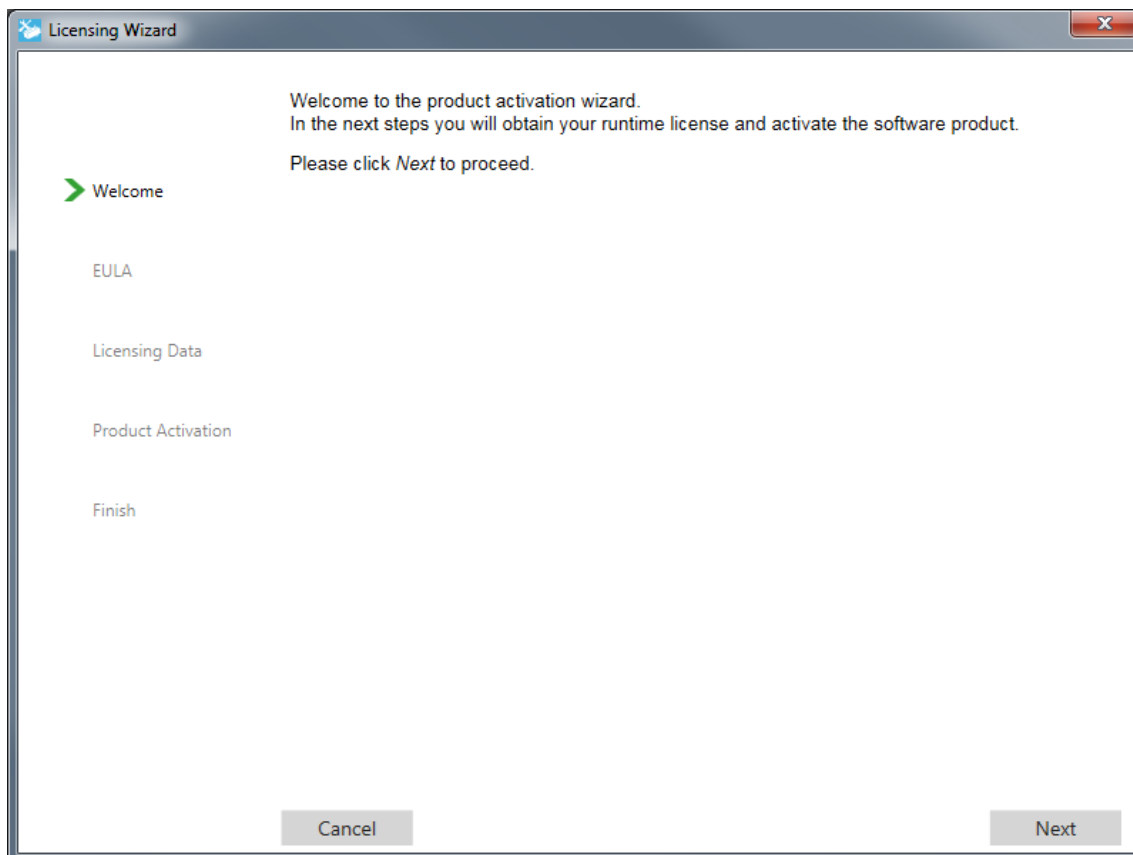
A license can be added to the local machine or a USB license dongle can be used. To use a dongle license, the dongle must be plugged in into the machine where the **PMS.Cliptest.Service** should be executed. The process for adding new local licenses is described in chapter [1.2.1 License Key](#).

### 1.2.1 License Key

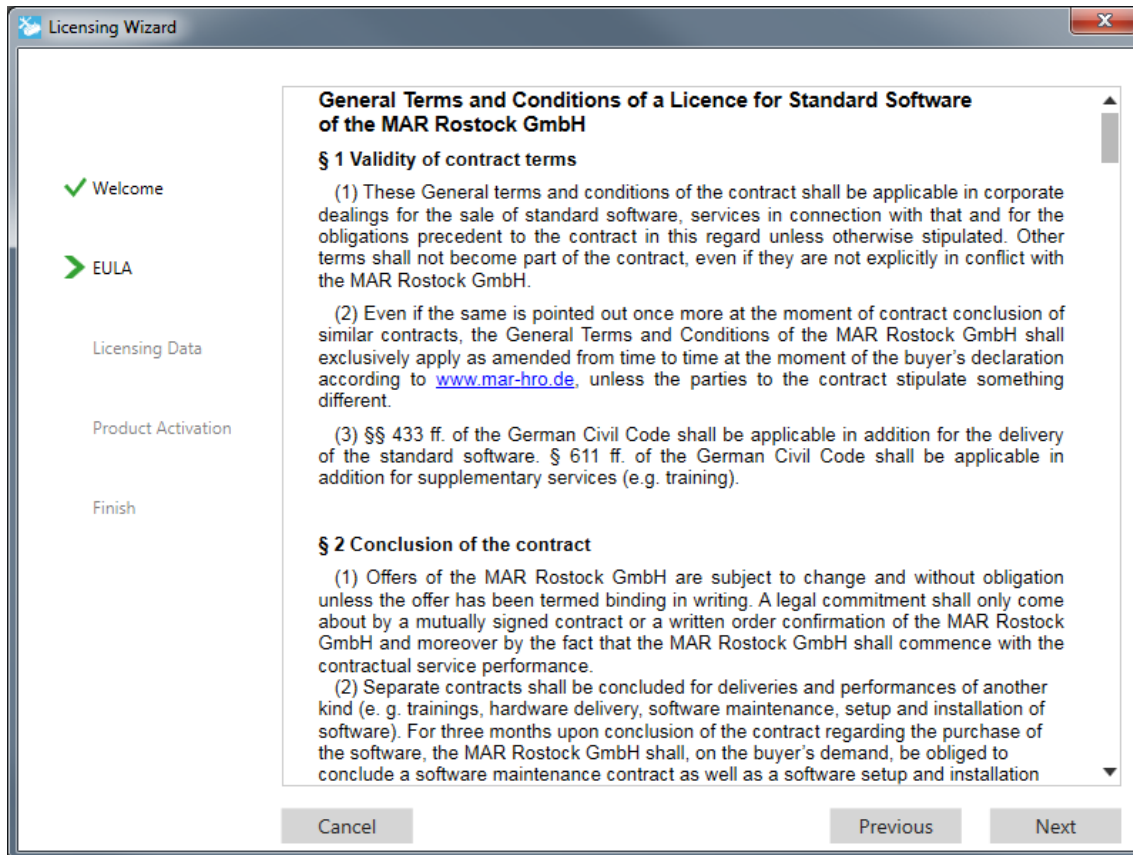
To add licenses with a license key, the **PMS.Cliptest.ServiceConfigurator** must be used. If the security option is enabled, a password prompt will be sent before proceeding to start the **ServiceConfigurator** (For further information see chapter [5.3 PMS.Cliptest.ServiceConfigurator](#)). At the tab page **Licensing** you will find the currently installed licenses and are able to add new local licenses.

 Figure 1: PMS.Cliptest.ServiceConfigurator Licensing

The button **Add New Licences** starts a wizard to add new licenses. The screenshot below shows the initial dialog of the wizard.



Click next to proceed to the wizard. In the next step, a prompt to read the license terms and conditions will be sent and one must agree to proceed.



In the following step, a license key must be added. The corresponding software product (module) to your license key must then be selected, the license key entered and the module must be added to the list. After all the license keys for the **PMS.Cliptest.Service instance** are added **Next** must be clicked to proceed.

**Licensing Wizard**

To proceed the activation procedure, you have to generate your unique **installation id** of each software module.  
Please fill in your **license keys** for each **module**, which you have got in reply on your software order, and proceed after entering all license keys by clicking *Next*.

☒ Welcome  
☒ EULA  
☒ **Licensing Data**  
 Product Activation  
 Finish

Module:

License key:

| MODULE               | LICENSE ID     |
|----------------------|----------------|
| PMS.ClipTest.Service | 0123-4567-89AB |

All added license keys must be activated in the following step. There are two possibilities to activate a license key. If a computer is connected to the internet, the automatic activation function can be used. Otherwise, the Website can be used for activation (<http://pac.mar-hro.de>). The license key and the generated Installation id can then be used for activation, the confirmation box checked and "submit" clicked. The activation Key must be entered manually. Repeat this procedure for each entered license key by clicking the **Next license** button. If a problem arises during the license key activation process, it can be skipped.

Licensing Wizard

Now the software can be activated to associate the **installation id** and the **license key**.

Therefore, you will need an **activation key**. This activation key can be obtained manually by our website (<http://pac.mar-hro.de>) or automatically by pressing *Automatic activation* (an internet connection is necessary).  
If you choose the manually method, please go to our product activation website and fill out the activation form (you need the **license id** and the **installation id** for this). After submitting you'll get the **activation key**. Enter the **activation key** into the field *Activation Key* on this window. After activating all licenses click *Next* to proceed.

**Remarks:** If you have provided a contact e-mail address on your software order you will receive an additional email with the summary of the product activation information.

License 1 of 1       

Module:

License key:

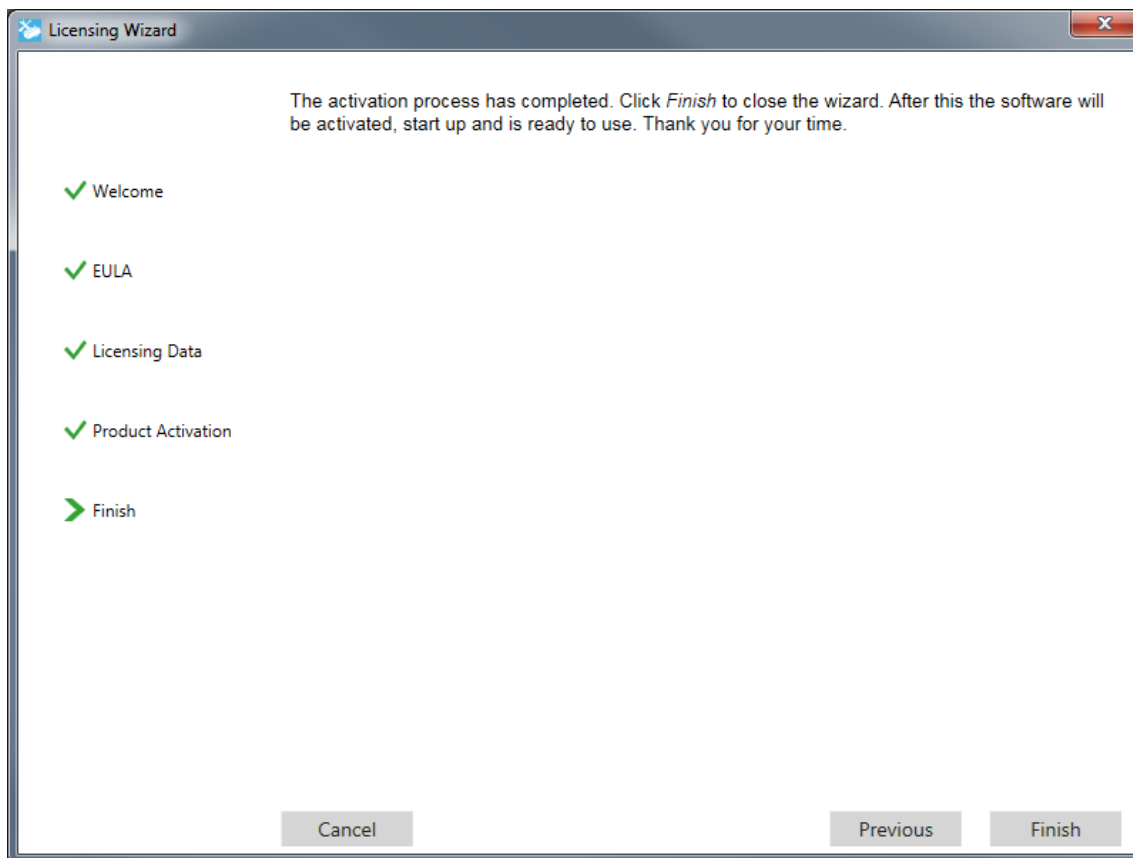
Installation id:

Activation key:

Cancel    Previous    Next

Progress: Welcome, EULA, Licensing Data, Product Activation, Finish

If all licenses are activated (or skipped), **Next** must be clicked to proceed. The last step of the activation process is completed when the **Finish** button is clicked.



### 1.2.2 USB Dongle

The second possibility to add new licenses to the **PMS.Cliptest.Service** is by using the USB dongle license system. Licenses can be added from multiple USB dongles. The **PMS.Cliptest.Service** will fetch the licenses from all plugged in dongles. The available licenses are shown in the **PMS.Cliptest.ServiceConfigurator** on the tab page **Licensing**.

**Attention** The license file (*pms.lic*) must remain on the dongle. Do not modify, copy or override this file. Any modifications of the license file can invalidate all stored licenses.

### 1.2.4 PMS.Cliptest DEMO

The **PMS.Cliptest.Service** can be started in a DEMO mode, by adding a demo license with the license wizard. This license expires after 50 days. The DEMO mode has no limitation regarding the functionality.

## 1.3 Designated use of PMS.Cliptest

**PMS.Cliptest** was developed to test the correct assembling of orders or order types on assembly boards. For further descriptions, imagine an assembly line where products are manufactured by assembling modules. **PMS.Cliptest** will verify the correct installation of modules to enhance the quality of the produced products.

Therefore, the assembly boards must have been equipped with radio receiver clips which will be pressed when a module is installed. **PMS.Cliptest** uses multiple radio receivers to obtain the current states of all clips (pressed / unpressed) and is now able to check this data against the configured master data. If all necessary clips are pressed (depending on master data), the result of the test will be ok. If the test time is over and not all clips are pressed correctly the test result will be not ok.

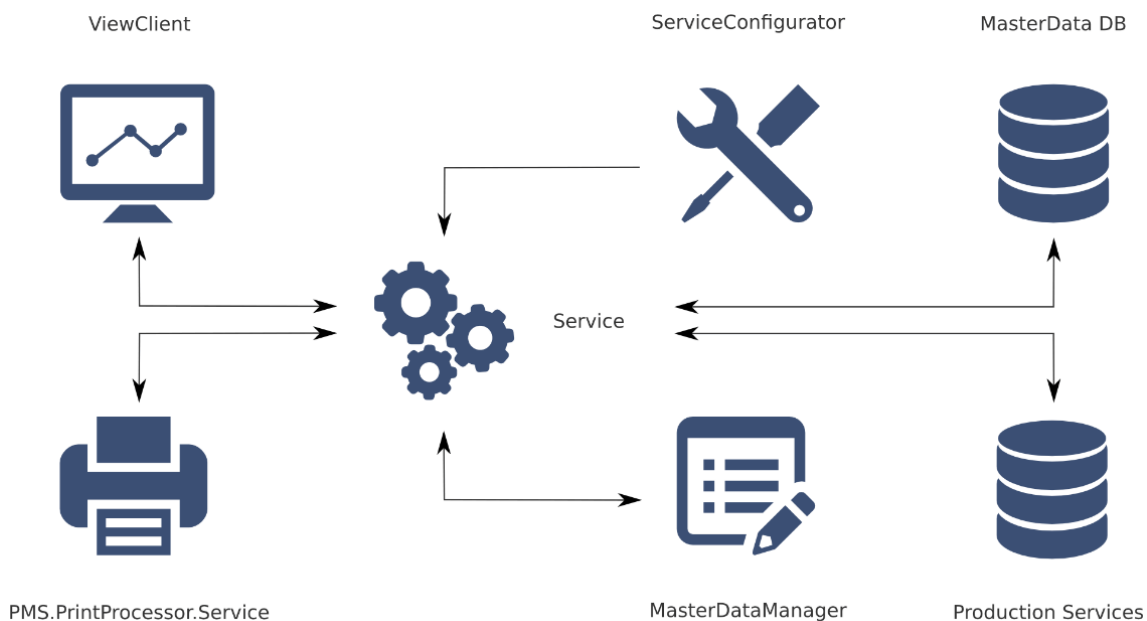
There are mainly two types of tests that can be processed by **PMS.Cliptest**:

**Clip Test:** A Clip Test is the main test of **PMS.Cliptest**. The test can run some time (is configurable) and shows the clips that should be pressed and which clips are already pressed to obtain a positive test result on a panel. If all clips are correctly pressed during a running test, this test is automatically finished with a positive test result. If the test time is over and not all clips are pressed correctly, the test result will be negative.

**Empty Test:** An Empty Test can be used to ensure that all clips on a board are currently unpressed. This may be used to grant an initial state of the board before start assembling a new product on there. This test can also run some time (configurable) and shows all clips of the board and their states on a display. If all clips are unpressed while a running empty test, this test is automatically finished with a positive test result. If the test time is over and not all clips are unpressed, the test result will be negative.

## 1.4 System structure

The PMS.Cliptest application is divided into several parts. The main part is the **PMS.Cliptest.Service** (short: Service). It handles the test logic, the master data import and the communication to the **PMS.PrintProcessor.Service** (short: PrintProcessor). The configuration of the Service can be changed by usage of the **PMS.Cliptest.ServiceConfigurator** (short: ServiceConfigurator). The **PMS.Cliptest.ViewClient** (short: ViewClient) or the **PMS.Cliptest.MasterDataManager** (short: MasterDataManager) can be used for the interaction with the service. All print jobs can be handled by the **PMS.PrintProcessor.Service**.



The radio receivers, which transmit the clip press events to the **Service**, are directly connected to it.

## 1.5 Program Modes

**PMS.Cliptest** can be used in various combinations.

### 1.5.1 Without Production Server

In this configuration, **PMS.Cliptest** acts as a single workplace instance. Start commands for clip- or empty-tests are received from a scanner connected to the **ViewClient**. The scanned barcodes identify the assembly boards necessary for the test.

In this configuration, **PMS.Cliptest** acts as a single workplace instance. Start commands for clip- or empty-tests are received from a scanner connected to the ViewClient. The scanned barcodes identify the assembly boards that are necessary for the test.

### 1.5.2 With Production Server

The production server can be a higher level manufacturing execution system (e.g. MES) to which **PMS.Cliptest** is able to communicate with.

In this configuration, **PMS.Cliptest** uses the local database to handle master data but uses also the configured production server interface for the communication to any supported production system. Start commands for clip tests or empty tests as well as the list of clips for a test can be pre-generated and received from the production server.

## 2 Program Runtime Requirements

### 2.1 Hardware Requirements

If all applications are currently not running on the same computer, a 100/1000MBit Network card is required.

- X86/X64-Platform with 1,0 Ghz (min. 2,0 Ghz recommended)
- 2 GB RAM (4 GB recommended)
- 500 MB free disc space for **PMS.Cliptest** and distributed Database
- 1000 MB free disc space for **PMS.Cliptest**
- recommended screen resolution 1920 x 1080

#### 2.1.1 PMS.Cliptest.Service

- When running with RS232 receivers, at least one free COM-port (RS232) is required for each receiver.

#### 2.1.2 PMS.Cliptest.ViewClient

- If running with a RS232 barcode scanner an additional free COM-Port (RS232)
- If running with a USB barcode scanner a free USB Port

#### 2.1.3 PMS.Cliptest.MasterDataManager

- A free COM-Port (RS232) for barcode-scanner is required for scanning addresses from clips.

### 2.2 Software Requirements

- Microsoft Windows 10 (Version 1903) (**PMS.Cliptest** was tested to work with this operating system)
- Microsoft .net Framework 4.7.2 (If the required version of the .net Framework is not installed, the setup shows an error. The setup for the .net Framework is required at the installation medium)
- Access to the default communication ports (for further information see chapter [5.3.2.1 General](#))

|                     |       |     |                 |  |
|---------------------|-------|-----|-----------------|--|
|                     |       |     |                 |  |
| Basic communication | 45181 | TCP | <b>required</b> |  |
| Web Interface       | 45183 | TCP | optional        |  |

## 3 Installation guide

### 3.3 Installing PMS.Cliptest

This chapter describes how to install **PMS.Cliptest** with the delivered **PMS.Cliptest** Setup. It is important to ensure that a Microsoft SQL Server (e.g. Express version) is installed.



### 3.3.1 Deployment options

This section describes possible installation scenarios with an explanation of the required licenses.

#### 3.3.1.1 Classic use case

The typical installation scenario has two computers, one for the Empty Test and one for the Clip Test. All applications (Service, ServiceConfigurator, MasterDataManager,ViewClient) must be installed on one machine. Only the ViewClient can be installed on the second machine. At least one **PMS.Cliptest.Service** license and one **PMS.Cliptest.ViewClient** license are needed for this scenario.

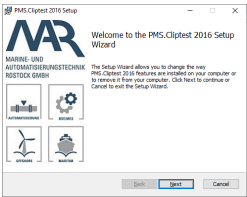
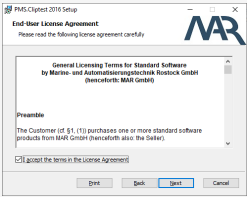
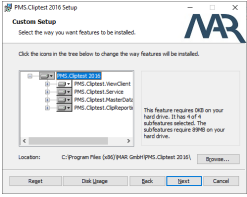
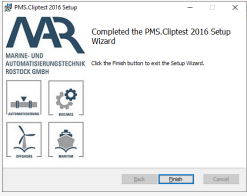
#### 3.3.1.2 Standalone use case

In the standalone use case, all applications must be installed on a single machine. At least one **PMS.Cliptest.Service** license is needed for this configuration to be used.

### 3.3.2 Setup Steps

Click on "PMS.Cliptest Setup.exe" to start the setup wizard. The setup wizard automatically detects whether the Microsoft .net Framework is installed. If it cannot be found, it must be installed manually.

Follow the steps above:

| Dialogue  | Steps  |
|---|--|
|   | Click Next to proceed to the license agreement   |
|  | Accept the License agreement and press Next to proceed to the feature dialog   |
|  | Choose in this step the all components which you want to install. By Default, the setup installs the PMS.Cliptest.Service, PMS.Cliptest.ViewClient and the PMS.Cliptest.MasterDataManager. You can customize the installation by selecting single components. The install location can be changed by selecting PMS.Cliptest. |
|  | After the installation is completed close the dialog by clicking the finish button. The PMS.Cliptest.Service is not started automatically without a reboot. The PMS.Cliptest.ServiceConfigurator can be used to start the PMS.Cliptest.Service (See chapter 5.3). Upgrading PMS.Cliptest.                                    |

### 3.4 Upgrading PMS.Cliptest

During the upgrade progress to newer versions of **PMS.Cliptest**, a database upgrade might be required. If you have an older version of **PMS.Cliptest** installed, the database must be upgraded before the new version can be used. The following steps must be followed to upgrade the database.

1. Create a backup of the current master data with the MasterDataManager (ref. see chapter [5.2.4.1 Backup](#))
2. Uninstall the current version of **PMS.Cliptest**
3. Install the new version of **PMS.Cliptest**
4. Create a new database with the ServiceConfigurator (the old database can be deleted if update was successful). Save the changes and restart the service
5. Restore the backup with the MasterDataManager

In some cases it is also possible to reconfigure the service application

## 4 Program Master Data Requirements

This chapter describes the necessary master data that is required for the program to be run effectively. The master data must be prepared as CSV-files and can be imported by using the master data import. Refer to the following sub-chapters for detailed descriptions about the necessary CSV-File data.

The structure of the files is defined as CSV (comma separated variables). A comma or semicolon is used to separate the data of columns inside the file. The default separator is a comma. To set a new separator, refer to chapter [5.2.3.3 CSV Settings](#).

The column headers can also be written into file as the first line. The default comment token is # To set a new comment token, Refer to chapter [5.2.3.3 CSV Settings](#) for detailed information.

The following master data must be prepared when using **PMS.Cliptest** without any production server (Production Server Interface NONE).

- PinStatus file
- Cliplist file
- Variants file
- Coordinates file (optionally)

The following master data must be prepared when using **PMS.Cliptest** in combination with an external production server:

- PinStatus file
- Coordinates file (optionally)
- Modules file (depending on production server interface)

### 4.1 PinStatus file

The PinStatus file defines the structure of every board. It defines all LEDs (Clips) and their addresses for every board. Therefore, it defines the correlation between the physical Clip (Clip-address) on the board and the virtual Clip-Number defined in the master data for a module of an order (CAD data of the order). It is also used in runtime to retentively save the states of all radio receiver clips (ON/OFF). This file must contain all clip IDs of all assembly boards to be processed with the used instance of **PMS.Cliptest**. The combination of LED, Address and Board have to be unique, but it is possible that the same LED number with multiple addresses or the same address with multiple LED numbers are displayed on the same board. The information can be split into multiple files, e.g. one file for each board.

Necessary CSV-File structure:

| Column name        | Column type | Comment  |
|--------------------|-------------|--|
| <b>LED</b>         | Short       | Unique Clip position on Board (LED)  |
| <b>Address</b>     | String      | Unique ID of radio receiver clip (e.g. 001099BB)   |
| <b>Board</b>       | String      | Assembly board where the clip is located on  |
| <b>ForceTest</b>   | Boolean     | <b>Optionally:</b> True/False; True if clip must be tested in mode "Unpressed clips", even if the clip was already pressed before.   |
| <b>KSKType</b>     | String      | <b>Optionally:</b> The KSKType is an additional identifier to differentiate same LED/Addresses on the same Board (This option can be used for specific types of harnesses). Leave it empty if there are no duplicate LED/Addresses on the same Board. This feature can only be used in combination with the ForceTest. |
| <b>Description</b> | String      | <b>Optionally:</b> The Description is an optionally property to get more information about a Clip. Leave it empty if there is no description. This feature can only be used in combination with the KSKType.   |
| <b>ClipGroup</b>   | String      | <b>Optionally:</b> This field configures the ClipGroup. Leave it empty if the Clip is not a member of a group. This feature can only be used in combination with the option Description.   |

The file can contain three columns up to seven. If the file contains no "ForceTest" column, the related value is initialized with "False". If the file contains no "KSKType" column, the value will be empty. The "KSKType" value can only be used in combination with "ForceTest". The option "Description" can only be used in combination with "KSKType" and the option "ClipGroup" can only be used in combination with "Description".

#### 4.1.1 Clip Groups

The Clip Groups feature allows the user to assign clips to different groups. Each clip can only be a member of one group. If all clips of a group are in the state IO when a test is running, the status of the clips switch to GRPIO and the status will not change until the end of the test (Clips that are in the state NLIO will be not considered for this calculation). Changes of the clip status will be not considered in the clip state calculation. The behavior is similar to the test mode clips must be pressed once for all clips in a group. The clip groups can be configured in the master data files.

#### 4.2 Coordinates file

The coordinates file can optionally be used to store additional coordinate information for the clip positions (LEDs) on a board. This data is additionally displayed on the panel of **PMS.Cliptest**. They can be used to define an X/Y coordinate system to simplify the search for a physical clip position on a board. The configured values are not case sensitive. All letters will be converted to upper letters. The information can be split into multiple files, e.g. one file for each board.

If the Matrix layout is required for the **PMS.Cliptest.ViewClient**, the requirements for the coordinates (see chapter [5.1.4.3 Matrix layout](#)) can be examined.

Necessary CSV-File structure:

| Column name  | Column type | Comment        |
|--------------|-------------|----------------|
| <b>Board</b> | String      | Assembly board |

|                |         |  |
|----------------|---------|--|
| <b>KSKType</b> | String  | Reserved for future usage (you can enter arbitrary data)             |
| <b>LED</b>     | Integer | Clip position on assembly board wherefore the X/Y information is for |
| <b>X Axis</b>  | String  | X- Information for clip position                                     |
| <b>Y Axis</b>  | String  | Y- Information for clip position                                     |

### 4.3 Variants file

The variants file is used to define the different order types, which are produced with **PMS.Cliptest**. Additional parameters that can be used for the label are also defined. Every variant must have related entries in the Clip list, defining the necessary LEDs in this variant. The information can be split into multiple files, e.g. one file for each board.

Necessary CSV-File structure:

| Column name        | Column type | Comment   |
|--------------------|-------------|---|
| <b>Variant</b>     | String      | Name of the Variant                                   |
| <b>KSKType</b>     | String      | KSK Type of the variant                               |
| <b>Comment</b>     | String      | <b>Optionally:</b> Comment for the variant            |
| <b>Parameter_1</b> | String      | <b>Optionally:</b> User defined value for the variant |
| <b>Parameter_2</b> | String      | <b>Optionally:</b> User defined value for the variant |
| <b>Parameter_N</b> | String      | <b>Optionally:</b> User defined value for the variant |

### 4.4 ClipList file

The ClipList defines necessary clips (LEDs) for the defined variants and the given assembly boards. This file must contain all clips (LEDs) on the assembly board, which must exist (must be pressed) while the given variant is tested. The information can be split into multiple files, e.g. one file for each board.

Necessary CSV-File structure:

| Column name    | Column type | Comment   |
|----------------|-------------|---|
| <b>KSK</b>     | String      | Name of the Variant (see variants table)  |
| <b>Board</b>   | String      | Assembly board where the order / order type should be produced on   |
| <b>LED</b>     | Integer     | Clip position which must be pressed to correctly assembly the order / order type  |
| <b>Comment</b> | String      | Description for an order / order type to be displayed in PMS.Cliptest panel (should always be the same for the same order / order type) |

### 4.5 Modules files

The Modules define necessary clips (LEDs) for a module. This file must contain all modules with their respective clips for a harness. The information can be split into multiple files, e.g. one file for type of harness.

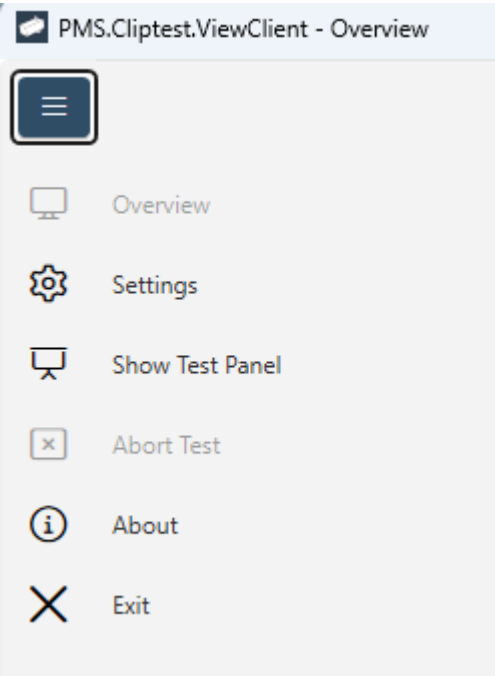
# 5 User Interface Overview

## 5.1 PMS.Cliptest.ViewClient

The main window of **PMS.Cliptest.ViewClient** (short: ViewClient) contains different views. By default, not all tab pages are displayed (simple mode). The expert mode with all configuration options can be activated in the Menu with a toggle switch on the bottom. All options which are not displayed in the simple mode can be described in the documentation.

### 5.1.1 Menu

Access to all features of the **PMS.Cliptest.ViewClient** is provided in the menu on the left side. The menu options are available via the icons. The menu can be expanded to display more information about the options. To expand or collapse the menu, click on the top icon (three horizontal lines).



The click expands or collapses the menu bar. The requested button must be clicked to switch between different views. To close the menu one can switch to another view or the top button.

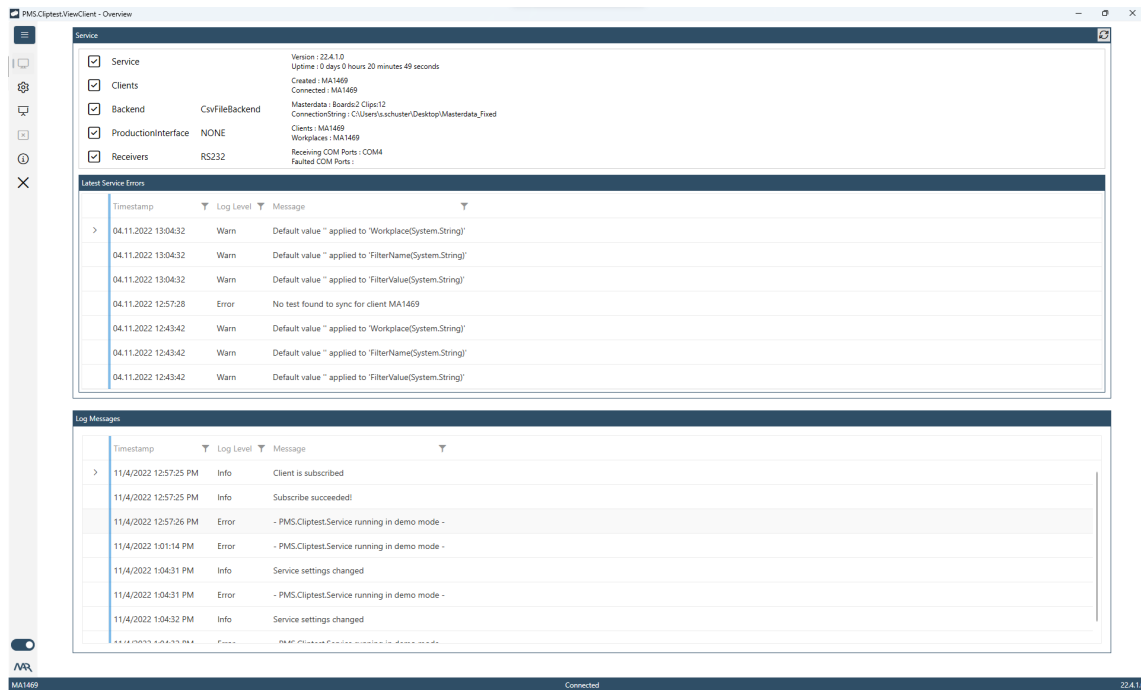
| Button          | Action  |
|-----------------|---|
| Overview        | Use this button to navigate to the overview.  |
| Login / Logout  | If the option "Using Operators" is checked, can be switched with these buttons.                         |
| Show Test Panel | To open the configured Test Panels click on this button.  |
| Abort Test      | If a test is running, it can be canceled by scanning the configured abort barcode or using this button. |
| About           | Information about the product and licensing can be found in the above view.                             |

Exit

Administrator privileges are required to close the PMS.Cliptest (configurable).ViewClient.

### 5.1.2 Overview View

This is the initial view of **ViewClient**. The current state of the application can be seen here.



On top of the overview, information about the status of the **PMS.Cliptest.Service** can be seen. Information about the connected clients, the internal **PMS.Cliptest** database, the configured production server, the clip receiver and the latest server errors is displayed.

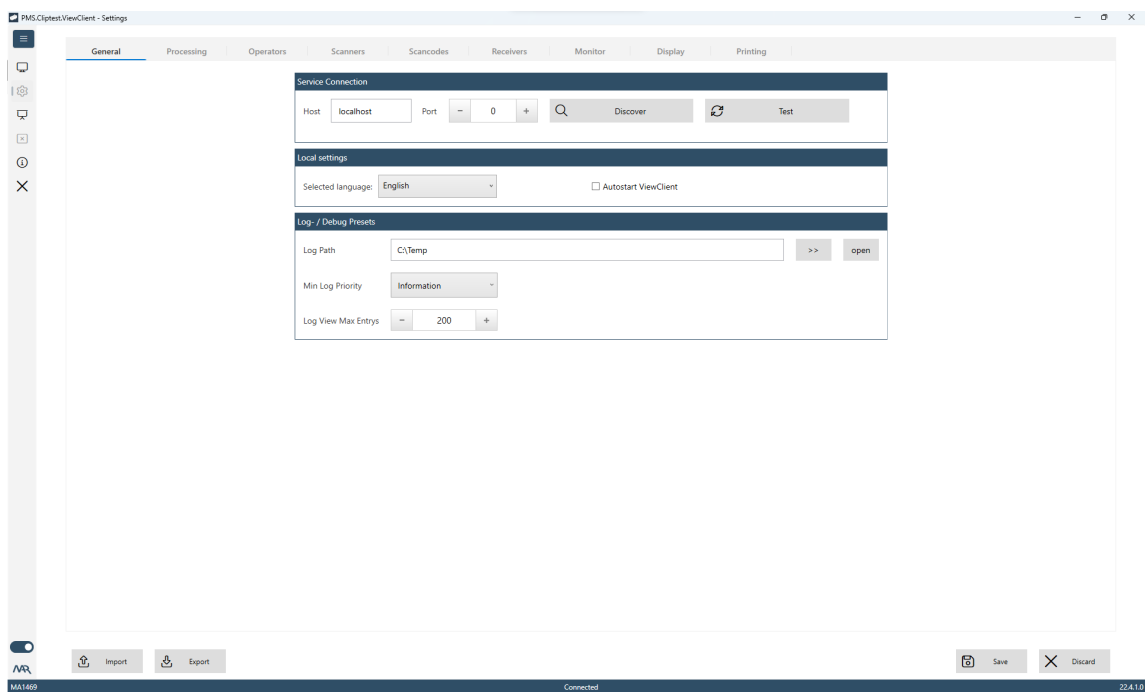
The log window on the bottom informs one about events and errors occurred in the **PMS.Cliptest** workflow. It also shows information messages, depending on the configured log level.

### 5.1.3 Settings View

The configuration can be opened by using the settings in the menu option. By default, the password protection for the settings is disabled (for further information see chapter [5.1.3.3 Operators](#)). When this option is enabled, the configured password is requested. The default user after installation of **PMS.Cliptest** is "Admin" with the password "0".

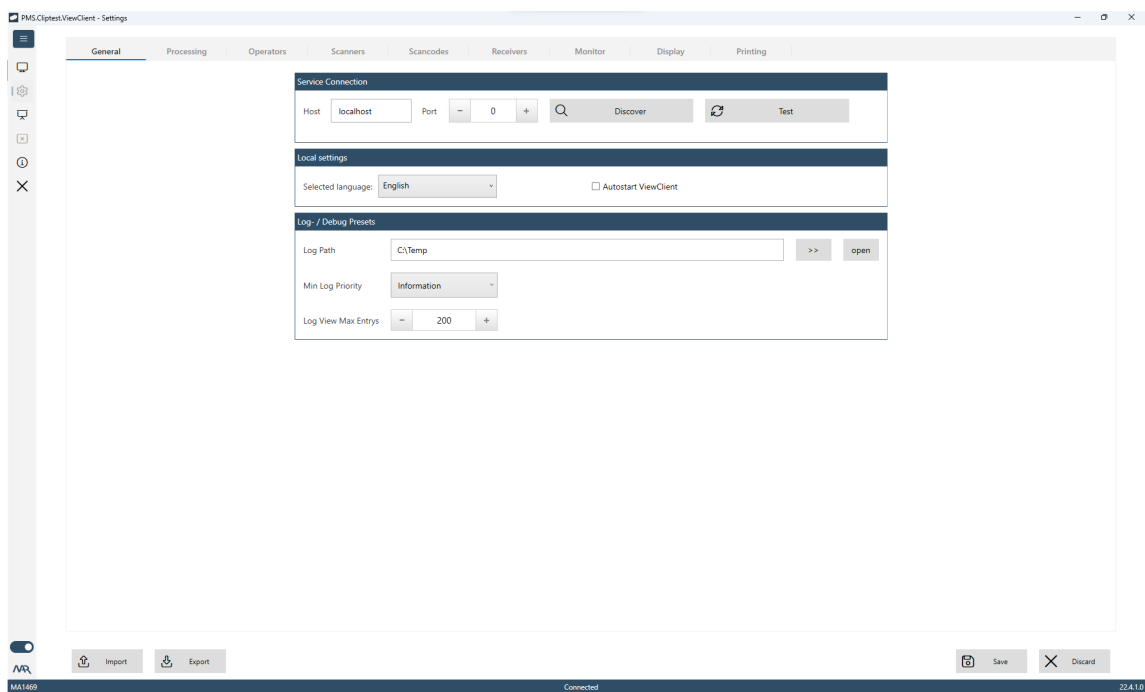
On the bottom of the settings view, current changes of the settings can be saved or discarded. Outstanding changes are signaled with a \* behind to program name on top of the settings window.

To Transfer the configuration from one client to another, **Import** and **Export** features can be used. The export feature enables the user to export the settings as an xml file, which can be imported on another client with the import Feature. The option **Workplace to Act for /Name By facility** will be not transferred because this setting should be unique for each instance of **PMS.Cliptest**.



### 5.1.3.1 General

The **General** page is used to configure the connection to the service, the language of the user interface and the log-/ debug-presets. Only the hostname (or IP-Address) and the port are required for the Service connection. If the user is not sure what setup options are available, the **Discover** button, which searches for all **PMS.Cliptest.Service** applications in the network, can be used. The input can be verified with the **Test** button. The language of the user interface can be selected in the second group box. The changes will be applied after the settings are saved. Additional information about the localization in the chapter 11 (Appendix III – Localization) will then be provided. If the **PMS.Cliptest.ViewClient** application needs to be started automatically on computer start, the option **Autostart ViewClient** must be checked.



The configurable options are described in the following table

| Log- /<br>Debug<br>Resets | Description  |
|---------------------------|--|
| Log Path                  | Enter a Path where PMS.ClipTest.ViewClient should save Log-Files. The default value is "{InstallationPath}/Logs".  |
| Min Log<br>Priority       | Select the minimum priority of Log-Events.   |
| Log View<br>Max Entries   | Enter the count of maximum entries to be shown in log-window. If log-window contains more entries than entered count the oldest entries will be removed from log-window. The default value is 200. |

### 5.1.3.2 Processing

The **Processing** tab page is used to configure various settings concerning the process flow of **PMS.ClipTest**. The program modes for running the **ViewClient** can be set. This page is divided into three sections.

#### 5.1.3.2.1 WORKPLACES TO ACT FOR / NAME BY FACILITY

The section Workplaces to act for configure the internal workplace name of **PMS.ClipTest.ViewClient**. For example, use "ASS01" as assembly line 1. The default value is the same as the ViewClientId. If different instances of **PMS.ClipTest.ViewClient** which are using the same database are in operation, different names for each instance must be configured to avoid interactions.

If the program is connected to a production server, this name must correspond to machine names configured in the server. The server will request the tests for the respective work places.

#### 5.1.3.2.2 TEST PARAMETER

In the **Test Parameter** section, clip conditions and the timings can be configured.



The following table describes each configurable option

- Test mode
  - Configures the mode, which defines under which condition a clip is recognized as OK while running a clip test. The mode is described in the corresponding chapter. The check boxes below the combo box display the three conditions which must be fulfilled to recognize a clip as OK corresponding to the chosen test mode.
- Min Clip Press Time
  - The Min Clip Press Time defines how long a clip must be pressed until it is recognized as pressed. This parameter affects the sub status of a clip during the pre-process (if clip is pressed before the test is started). The sub states are combined to one final status when the test is finished
- Maximum ClipTest time
  - Configures the maximum time a clip test is running. If this time has elapsed after starting a clip test the test is automatically completed. It is important to remember that a clip test will also be automatically completed if all necessary clips to be pressed for current processing order / order type (Cliplist) are ok. The default value is 600 seconds. Set this option to 0 if the test should be not completed automatically. If this option and the option Maximum EmptyTest time are configured with 0, the status strip on the operator window will be hidden.
- Maximum EmptyTest time
  - Configures the maximum time an empty test is running. If this time has elapsed after starting an empty test the test is automatically completed. It is important to remember that an empty test will also be automatically completed if all configured clips of the assembly board are not pressed. The default value is 20 seconds. Set this option to 0 if the test should not be completed automatically. If this option and the option Maximum ClipTest time are configured with 0, the status strip on the operator window will be hidden.

The Test mode affect the test process of the clip test. The empty test is not affected. A differentiation can be made to examine the conditions under which a clip is recognized as IO while running a clip test.

Each clip has two different sub states, which are combined to one final state, when the clip test is finished: status pre-process (clip pressed before the test is started) and status while the running clip test.

|   | Status pre-process | Status clip test | Clip pressed while test is completed |
|---|--------------------|------------------|--------------------------------------|
| Test Mode 1 - Unpressed clips               | X                  | X                | X                                    |
| Test Mode 2 - All clips pressed once        | 0                  | X                | X                                    |
| Test Mode 3 - All clips must remain pressed | 0                  | 0                | X                                    |

As shown in this table, one or several conditions can cause an IO status depending on the chosen test mode. In the first test mode an IO status from the pre-process OR an IO status from the clip test OR if the clip is pressed in the moment when the test is finished result in a final IO. The sub states (status pre-process, status clip test) depend on the latency period. The following conditions will cause an IO status.

- Pre-process
  - If the clip is pressed before the test is started and remains pressed until the latency period is exceeded, this sub status will be recognized as IO. When the test is started, **PMS.Cliptest** loads the sub states of the pre-process for all contained clips. Only clip events, which were raised after the generation of the clip list, are taken into consideration.

- ClipTest
  - If the clip is pressed while the test is running, this sub status will be recognized as IO. The latency period is not taken into consideration.

Following the three modes will be described and how the mentioned sub states are combined to calculate the final status of a clip.

#### 1. Un-pressed clips

- **PMS.ClipTest.Service** collects every clip event in the local database, even for those clips which are not contained by an actual running test and even if no test is actually running. This log provides the information if a clip was pressed before the clip test was started (pre-process). In this mode the log is taken into consideration to calculate the final status of a clip during the running test. If a clip was pressed before starting the test and it fulfills also the configured latency period (duration, how long the clip must remain pressed during the pre-process), the final clip status during the test is IO, even if the clip is not pressed during the test. The final clip status will be IO under the following conditions
  - ☒ Clip was pressed in the preprocess (before clip test) and remained pressed until the latency period was IO
  - ☒ **OR** this clip was pressed during the running clip test without taking into consideration the latency period. **If the clip was pressed in the pre-process and remained pressed until the test was started, it will not be recognized as IO until it was released and pressed again.**
  - ☒ **OR** the clip is pressed, explicitly when the clip test is completed (pressed during or until the end of the test).

#### 2. All clips must be pressed once (during the running clip test).

- In this mode the clip must be pressed once during the running clip test and must remain pressed until the latency period was fulfilled to achieve an IO status for the clip. The final clip status will be IO under the following conditions.
  - ☒ The clip was pressed during the running clip test without taking into consideration the latency period. **If the clip was pressed in the pre-process and remained pressed until the test was started, it will not be recognized as IO until it was released and pressed again**
  - ☒ **OR** the clip is pressed, explicitly when the clip test completes (pressed while or until the end of the test).

#### 3. All clips must remain pressed.

- In this mode the clip must be pressed until the end of the test to achieve an OK status for the clip. The clip will be missing, if the clip was pressed during the test but gets un-pressed before the test was finished.

Furthermore, the first two modes take into consideration the ForceTest flag from the master data: Single clips can be forced to be tested even if they were already pressed before the test or during the test in these two modes. Therefore, an additionally flag "ForceTest" must be added to the imported master data PinStatus.csv. If this flag is activated (True), the clip is required to be pressed until the test was finished.

For a better understanding, the final status for a clip differentiated for all three modes is shown in the following table.

In the following table the latency period is set to 10 seconds.

| Process   |           |                     |                   | Resulting Sub-States |                 |                              | Config     | Final Status in Testmodes |              |              |
|-----------|-----------|---------------------|-------------------|----------------------|-----------------|------------------------------|------------|---------------------------|--------------|--------------|
| Configure | Requester | PreProcess Duration | ClipTest Duration | PreProcess Status    | ClipTest Status | Clip pressed until end of te | Force Test | 1                         | 2            | 3            |
| true      | true      | 7                   | 0                 | Missing              | Missing         | false                        | true       | Missing                   | Missing      | Missing      |
| true      | true      | 12                  | 0                 | OK                   | Missing         | false                        | true       | Missing                   | Missing      | Missing      |
| true      | true      | 7                   | 15                | Missing              | OK              | false                        | true       | Missing                   | Missing      | Missing      |
| true      | true      | 12                  | 22                | OK                   | OK              | false                        | true       | Missing                   | Missing      | Missing      |
| true      | true      | 7                   | 0                 | Missing              | Missing         | true                         | true       | OK                        | OK           | OK           |
| true      | true      | 12                  | 0                 | OK                   | Missing         | true                         | true       | OK                        | OK           | OK           |
| true      | true      | 7                   | 23                | Missing              | OK              | true                         | true       | OK                        | OK           | OK           |
| true      | true      | 12                  | 23                | OK                   | OK              | true                         | true       | OK                        | OK           | OK           |
| true      | true      | 7                   | 0                 | Missing              | Missing         | false                        | false      | Missing                   | Missing      | Missing      |
| true      | true      | 12                  | 0                 | OK                   | Missing         | false                        | false      | OK                        | Missing      | Missing      |
| true      | true      | 7                   | 1                 | Missing              | OK              | false                        | false      | OK                        | OK           | Missing      |
| true      | true      | 12                  | 2                 | OK                   | OK              | false                        | false      | OK                        | OK           | Missing      |
| true      | true      | 7                   | 0                 | Missing              | Missing         | true                         | false      | OK                        | OK           | OK           |
| true      | true      | 12                  | 0                 | OK                   | Missing         | true                         | false      | OK                        | OK           | OK           |
| true      | true      | 7                   | 12                | Missing              | OK              | true                         | false      | OK                        | OK           | OK           |
| true      | true      | 12                  | 12                | OK                   | OK              | true                         | false      | OK                        | OK           | OK           |
| true      | true      | 12                  | 0                 | OK                   | OK              | true                         | false      | OK                        | OK           | OK           |
| false     | true      | -                   | -                 | unconfigured         | unconfigured    | unconfigured                 | -          | unconfigured              | unconfigured | unconfigured |
| true      | false     | 12                  | 22                | unnecessary          | unnecessary     | true                         | -          | unnecessary               | unnecessary  | unnecessary  |
| true      | false     | 7                   | 1                 | OK                   | OK              | true                         | false      | OK                        | OK           | unnecessary  |
| true      | false     | 12                  | 12                | unnecessary          | unnecessary     | false                        | false      | unnecessary               | unnecessary  | OK           |
| true      | false     | 7                   | 15                | OK                   | unnecessary     | true                         | false      | OK                        | unnecessary  | unnecessary  |

### 5.1.3.2.3 TEST PROCESSING SETTINGS

The general options which influences the behavior of the test processing can be configured In the **Test Processing Settings** section.

The following table describes each configurable option.

- EmptyTest before ClipTest necessary
  - Requires that an empty test with result IO has to be performed for the given board before a clip test can be started. After every ClipTest, a new empty test has to be done.
- Start EmptyTest after Cliptest
  - If this option is enabled, the EmptyTest will be started directly, when the ClipTest is finished, automatically starts the respective EmptyTest if the corresponding process status was set by the production server.
- Load only clips for actual workplace
  - If this option is enabled, ViewClient will only load the clip list for the actual running work place. The work place must be provided by the server. With help of this option, it is possible to use several clip tests on one line, so that each work place can test a different range of clips.
- Using Operators
  - If this option is checked, a scan can only be performed if an operator is logged into the program. A change of the variant is only possible with a logged in operator.
- Enable Quality Confirmation
  - This option is used to require a qualified user to confirm unusual critical events. A dialog at the end of each test will ask for confirmation or the test will be finished with NIO.

### 5.1.3.2.4 STATISTIC DATA

In the **Statistic Data** section, the statistical data for creation can be configured.

- Export of test data
  - The detailed status of a clip when the test was completed can be exported to the **PMS.Cliptest** backend, to the selected production server interface or to a file. Therefore, the corresponding check box must be activated (XML or CSV) and the path must be setup. To save the test data to a backend, one of the available options must be selected.

- Export ClipLog
  - This function is only available if **PMS.ClipTest.ClipReporting** is licensed. All clip events related to the running clip test are collected and traced. After the clip test is finished or aborted, these clip events are stored permanently. The information contains a unique ID for the performed test, including the number of presses per clip involved in this test. The information can be saved into the **PMS.ClipTest** backend or the production server interface.

### 5.1.3.3 Operators

The use of **Operators** can be activated in the settings tab **Processing**. With the use of operators, the operator must perform a log in to start a process by scanning or to change the order variant.

On this settings section, the authorization features can also be enabled or disabled. By default, these options are disabled. With the option **Require login to access settings**, the settings can be protected against unauthorized changes. If closing the application without privileges shouldn't be allowed, enable the option **Require login to close application**.

To login an operator, the log in button in the menu must be pressed. If an operator is logged on, the operator name is displayed in the status bar on the bottom of the main windows. The Logoff button is located in the menu and will log off the logged in operator. After a logout, a new operator can be logged in.

If no operator was logged in and a scan was performed, the program will automatically open the log in box. If cancel is pressed and no operator is logged in, the scan is canceled.

#### 5.1.3.3.1 MANAGE OPERATORS

To Add a new operator, a name and a password must be entered and the add button must be clicked. To delete an existing operator, it must be selected from the list, its name entered and the delete button pressed. To change the password for an operator, it must be selected or its name as well as a new password entered and the change password button clicked.

### 5.1.3.4 Scanners

The Scanner tab page is used to configure the scanner settings. This scanner is used to scan the assembly board barcodes to start a ClipTest or an EmptyTest for a special board or any other data.

The **PMS.ClipTest** ViewClient application supports two different types of scanners. The scanner can be connected via the COM-interface (physical or virtual COM-Port) or via USB as a keyboard. The COM-Interface must be configured with specific options (see chapter Fehler! Verweisquelle konnte nicht gefunden werden. for further information). The USB scanner uses an input box on the operator window to handle the data. This option can also be used for testing scenarios.

#### 5.1.3.4.1 COM (RS232) SCANNER

This section describes all required configuration options for a COM-port scanner which is connected via RS232.

| Value         | Description  |
|---------------|--|
| Name          | A unique name that describes the scanner   |
| Prefix Length | Number of characters which are used for the address stamping. The default value is 0 (no address stamping enabled) |
| COM port      | Enter the COM-Port you have connected your scanner to e.g. COM1  |
| Baudrate      | Select the baudrate the scanner works with. The default value is 9600  |
| Parity        | Enter the parity the scanner works with. The default value is NONE   |

|                 |   |
|-----------------|---|
| Stopbits        | Enter the count of stopbits the scanner works with. The default value is one  |
| Databits        | Enter the count of databits the scanner works with. The default value is eight data bits  |
| Scan Terminator | Scanners send the scanned value with a terminator in data stream. This terminator marks the end of the scanned data. To make PMS.Cliptest.ViewClient, correctly evaluated scans of the connected scanner this terminator must be set. Normal scanners do use a CRLF (Carriage Return Line Feed) as terminator so this is the default value. |

Since special chars like CR (carriage return) cannot be entered via keyboard, the ASCII-list can be used. The decimal values of the chars to be used as a terminator, e.g. "13" for CR or "10" for LF must be used. The ASCII-table can be used if one is not sure. The default value is CRLF and corresponds to ASCII{13} + ASCII{10}.

#### 5.1.3.4.2 USB SCANNER

This section describes all required configuration options for an USB scanner.

| Value           | Description  |
|-----------------|--|
| Name            | A unique name that describes the scanner   |
| Prefix Length   | Number of characters, which are used for the address stamping. The default value is 0 (no address stamping enabled)  |
| Scan Terminator | Scanners send the scanned value with a terminator in data stream. This terminator marks the end of the scanned data. To make <b>PMS.Cliptest.ViewClient</b> correctly evaluate scans of the connected scanner this terminator must be set. Normal scanners do use a CRLF (Carriage Return Line Feed) as terminator so this is the default value. |

Since special chars like CR (carriage return) cannot be entered via keyboard, the ASCII-list can be used. The decimal values of the chars can be used as a terminator, e.g. "13" for CR or "10" for LF. An ASCII-table can be used if one is not sure. The default value is CRLF and corresponds to ASCII{13} + ASCII{10}.

#### 5.1.3.5 Scancodes

In the section **scan code configuration**, the format of the scanned messages and the corresponding attributes. The scan codes are grouped in sequences. A scan sequence contains a type, the name of the sequence and the scan codes. Based on the type of the sequence, the action to be performed when all scans are fulfilled is decided by Cliptest. It is possible to have multiple sequences with the same type. The available types and some example data is listed in the following table.

| Value     | Description  |
|-----------|--|
| Cliptest  | This type defines a sequence of scans, which are required to start a cliptest. By default, a board scan and an order scan are required in this sequence. |
| Emptytest | This type defines a sequence of scans, which are required to start an emptytest. By default only a board scan with a prefix is required.                 |
| Optional  | This type is for scans, which are not required for starting a test. For example, the ABORT scan or additional scan data can be provided.                 |

The scan code sequences can be managed with the buttons on the right side of the sequences list. After selecting a scan code sequence, the configuration view for the sequence will appear below the list of sequences. On top of the sequence the type of the sequence can be changed. The name property is only used to allow the user to differentiate

between the sequences. The scan codes of a sequence can be managed right next to the scan code list. The option to add or remove a scan code and sort the scan codes is provided there. The sort feature is required for customs scans, which should be accessible in the printing feature. To configure a scan code, it must be selected from the list and the configuration view will appear.

The **Type** of a scan code defines how cliptest should handle the scanned data. There are several types described in the following section.

- Board
  - This type is used to identify a board scan independent from the requested test type. If your assembly boards may be called BR001 to BR999, enter "^ (BR)" for the Scan Pattern. If the connected scanner scans for example a "BR002"-code the board property for a test will be "BR002". The default value for the sequence type Cliptest is "^ (BR)". In case of sequence type Emptytest it is "^ (LE)". Additionally a Prefix for the Emptytest must be configured to differentiate between the scan sequences.
- Order
  - After scanning an order, the current order is changed to the scanned one. The scanned order must be configured in the master data tables Variants and Cliplist. If the order is not configured, **PMS.Cliptest.ViewClient** shows an error in the main program window. If the option Prefix is configured, it is cut from the scan code before the variant is processed. In this case, the master data must be provided without the prefix.
  - Examples
    - If the Prefix "OR" is configured and an order scan starts with "OR", then the "OR" is cut from the scan code. Scanning "OR123456" would be interpreted as an order scan referencing to order "123456".
    - If the Prefix is empty and an order scan starts with "OR", then the "OR" is not cut from the scan code. Scanning "OR123456" would be interpreted as an order scan referencing to order "OR123456".
- Abort
  - Enter a special code to abort a running test. If this field is set to "ABORT" and an "ABORT"-code is scanned from the connected scanner the currently running Cliptest will be completed. The default value is "ABORT".
- Modules
  - The modules scan contains a list of modules, which should be tested during a ClipTest. The Format of a single module can be described with the scan pattern. The module list needs additional master data, which can be managed with the MasterDataManager. For further information about the configuration see 5.2.7 Module Management
- Custom
  - To store additional information during a test, the Custom scan code can be used. The additional information can be used for printing. The order of the scan codes is described in the order of the bookmarks for printing.

The **Scan Pattern** describes with a regular expression the format of the barcode. If the scanned data should be trimmed at the beginning, one can configure the **Prefix**. The configured value will be removed at the start of scanned data. The **Separator** is only used for the type Modules. The configured Separator will be used to split the single data items from the scanned data. The option **Keep** allows the user to scan a scan code once and use the data until the next scan of this type. Do not use this option for all scan codes in a sequence this can cause unintended behaviors. A short description of regular expressions can be found with the help dialog. This dialog can be opened with the question mark symbol in the header of the scan code configuration.




### 5.1.3.6 Monitor

The **Monitor** tab page is used to configure the monitors where each panel of **PMS.ClipTest.ViewClient** should be shown. There are two different window types available. The first option is *HeaderWithLayout* and the second option is *Dashboard*. The configuration of both options are explained in the following chapters.

If a layout for a wrong monitor is setup up, select the window type *None* to remove a layout from a monitor.

#### 5.1.3.6.1 HEADER WITH LAYOUT

This window type has two sections. On top of the window is one of two available headers shown. The Default header contains information about the **OrderNumber**, the **OrderType**, the **Board** and the **result** of the current test.

|             |   |            |   |
|-------------|---|------------|---|
| OrderNumber |  | OrderType  |  |
| Board BR100 |   | Result NIO |  |

The Extended header displays information about the current test (**OrderNumber**, **Board**, **#Wrong**), client specific information (**#All**, **#User**, **User**) and general information (**Project Name**, **DateTime**). The field **#Wrong** displays during a ClipTest the number of clips, which are in the state **Needless pressed** (NLNIO), **Needless unknown** (NLUKN) or **Unconfigured** (UNCNF). During an EmptyTest the field displays the number of clips, which are in the state **Missing** (NIO) or **Unconfigured** (UNCNF). The field **#All** displays the number of all completed tests of the selected Client Id. Depending on the selected Production Server, the name of the current project can be displayed. If the feature using operators (see 0 for further information) is activated, the logged on operator name will be displayed in the field **User**. The field **#User** counts the completed tests of the current operator. This counter is reset after changing the user. If no operator is logged on, the counter displays the number of completed test since the last start of the **PMS.ClipTest.Service**.

|             |             |   |              |                     |   |
|-------------|-------------|---|--------------|---------------------|---|
| OrderNumber | Board BR100 |  | Project Name | 27.06.2019 10:57:41 |  |
| #Wrong 6    | #All 0      |   | User         | #User 0             |  |

Below the header is the test panel located. for the panel, a choice can be made between three different layouts types for each screen. The first type is the column layout. It displays a list of all clips and can be used without any preparation. The user is provided with separate filter options in each column. The matrix and the image layout require some additional configuration. The configuration of these layout types must be done in the **PMS.ClipTest.MasterDataManager**. For more information, see chapter 5.2.10

To setup a layout for a monitor, the monitor must be selected and after that one of the possible layouts can be chosen. After saving the settings, the new layout is applied.

When in doubt about configuring the requested monitor, the Identify Monitors button can be used. By default, the first monitor shows a test panel with only one column.

#### 5.1.3.6.2 DASHBOARD

The Dashboard layout enables the full flexibility of the layout design to the user. Different controls can be added to the panel. The location and size of the controls can be configured during the design process.

When this window type is selected the operator window opens with a white screen. To customize the layout, open the context menu with a right click to access the action buttons. For starting the customization use the *Edit* action.

The following table shows all available controls which can be added to the panel.

| Control | Description | Configuration |
|---------|-------------|---------------|
|---------|-------------|---------------|

|                   |  |   |
|-------------------|--|---|
| List Layout Flex  | List of clips which can be filtered, sorted and grouped (see <i>HeaderWithLayout</i> ). Offers enhanced flexibility to the user with more functions like grouping. | The configuration of the list is accessible via the context menu of the list header. To save any configuration it is required to use the "Save" option in the context menu of the Dashboard window. |
| List Layout Speed | Layout of clips which can be filtered and sorted. Applies UI changes very fast in high update scenarios.   | Can be configured with it's configuration dialog  |
| Image Layout      | Layout with clips which can be placed on a background image (see <i>HeaderWithLayout</i> )   | The configuration is accessible via the settings tab page display. To save any configuration it is required to use the "Save" option in the context menu of the Dashboard window.                   |
| Progress          | Progress bar which displays the remaining time of the current test   | Can be configured with it's configuration dialog  |
| Scan              | Provides input field for the USB scanner   | No configuration available  |
| Animation         | Animation of the clip pressing or releasing  | No configuration available  |
| Text              | Displays static text or dynamic test information via bookmarks   | The displayed content, the foreground and background color can be configured  |
| Image             | Displays images from a configurable path   |   |
| MAR Logo          | Image control which displays the MAR logo  |   |
| Next Scan         | Displays the next required scan  |   |
| Counter           | Displays counter values for completed test per session, user or client   | The counter type and the count conditions can be configured.  |
| Chart             | Displays a doughnut chart with slices of the clip states.  |   |
| Web               | Displays a website without refreshing the data (website is responsible to keep the data up to date). Only Microsoft Internet Explorer features supported.          | The URL of the website can be configured.   |

The configuration of the dashboard can be transferred between clients via the import and export actions in the context menu. The export contains the layout of the screen and all relevant configurations. External resources e.g. images will be not included in the export.

The dashboard view supports some shortcuts to simplify the usage of the actions. The following table describes all supported shortcuts.

| Shortcut | Action                              |
|----------|-------------------------------------|
| F2       | Enable or disable edit mode         |
| Ctrl + S | Save settings of the current window |



|     |                                       |
|-----|---------------------------------------|
| F5  | Synchronize current test with service |
| F11 | Displays or hide the title bar        |

The configuration of the dashboard view is possible without any extra license. But for using the view in the production, a PMS.Cliptest.Dashboards license is required. Without the license the view can be tested for 30 minutes (duration of successfully completed tests).

### 5.1.3.7 Display

The **Display** tab page is used to configure the settings of the test panel of **PMS.Cliptest.ViewClient**.

#### 5.1.3.7.1 STYLE CONFIGURATION

In the Style Configuration section, the display style of the clips on the operator windows is changed. To configure a style, select first one of the clip state on top of the configuration. One is able to configure the clip states **OK** (IO), **Missing** (NIO), **Needless** (NLNIO), **Unconfigured** (UNCNF) and **Group Ok** (GRPIO). For each of them, the font size, the diameter for the image layout and the foreground and background colors can be configured. If required, different style for Cliptests and Emptytests can be chosen. In the preview, the configured values for each test type can be seen. If the option is disabled, the options **Use different styles for Cliptest and Emptytest** the configured values for cliptests can be also used for emptytests. The emptytest style is not deleted when this option is disabled.

#### 5.1.3.7.2 TEST PANEL CONFIGURATION

All options for the test panel can be configured in this section. The options **Test Information Font Size** and **Clip Animation Size** configures the sizes for the test header (see chapter [5.1.3.6.1 Header](#))

The display performance of boards with many clips and constantly changing states can be slowed down by the grouping and filtering options. To avoid these display issues, the **Clip change interval** can be configured. If this option is set to 0, all clip updates will be performed immediately including filtering and grouping. With a higher value the grouping and filtering actions will be delayed and a group of updates are displayed at one time. The color changes of the clips states will be not affected by this option.

If viewing the clip updates in the live mode is not required, the option **Disable clip updates in live mode** must be checked.

The option **Enable filter and zoom modification** allows everyone to change the filtering, grouping and zooming settings in the test panel

#### 5.1.3.7.3 DESKTOP ALERTS

The **PMS.Cliptest** ViewClient application can notify the user if there are any errors or issues during working with the application e.g. invalid scan data. In the right bottom corner a desktop alert will appear if any issue arises. The display time can be configured. If the option is set to 0, the desktop alert is disabled.



Scanner read 'TESTSCAN' with type 'Unknown' and prefix "

#### 5.1.3.8 Printing

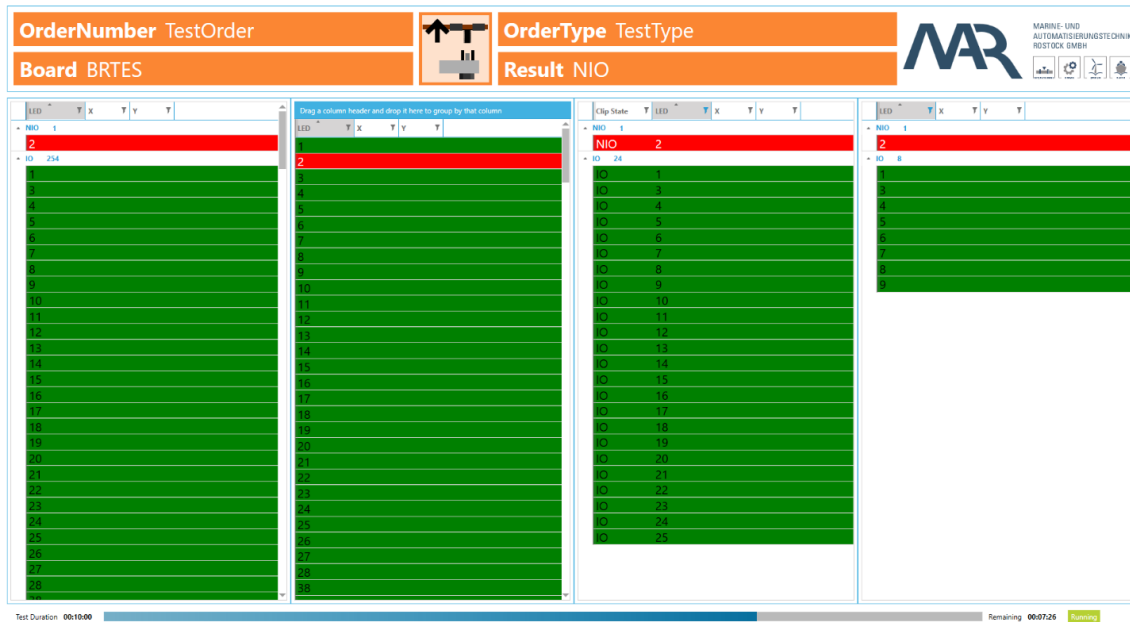
The **Printing** tab page is used to configure the print options of **PMS.ClipTest.ViewClient**. One may want to print labels after completing a test or when starting a test. Refer to chapter 9 for further descriptions

| Settings             | Description  |
|----------------------|--|
| Is Enabled           | Enables the Print Option for the defined printing event. The default value is disabled.  |
| Name                 | A name for better differentiation between your printing events can be defined.   |
| PrintProcessor Name  | Choose the PrintProcessor for the print job. The list of available PrintProcessors must be configured in the service using the ServiceConfigurator.  |
| Printer Name         | The printer for label printing can be chosen here. This is a list of known printers from the selected PrintProcessor.  |
| Label Name           | Select the template for printing the data. <b>PMS.ClipTest</b> is using labels designed with TFormer. This list contains all available labels for the selected PrintProcessor.   |
| Test Type            | Select the test type for the printing event. Possible values are ClipTest, EmptyTest and All.  |
| Test Status          | Select the test status for the printing event. Possible values are Idle, Created, Finished and All.  |
| Test Result          | Select the test result for the printing event. Possible values are IO, NIO and ALL.  |
| Copies               | This option is used for the count of copies. The default value is 1.   |
| Combined Test Result | The flag "CombinedTestResult" is used only for empty test labels (EmptyTest_Finished). If the flag is set, then the result of the empty test is combined with the last clip test result. That means, if you configure a result OK for the empty test label and check the "CombinedTestResult" flag, then the label is only printed, if the previous clip test and the empty test were OK for the given board. If you choose the result NIO, the label will be printed if the previous clip test or the empty test were not OK for board. |
| Confirm Printout     | Enable this option if you need to perform a scan-validation of your printed barcode. After this print event a dialog will appear (see chapter <a href="#">5.1.5.2 Label confirmation</a> ) asking for a value to scan. You can use valid bookmarks in the value configuration surrounded by square   |

brackets. For example the value "[Order]" will ask for the order number of the test. You can also use combinations of multiple bookmarks and static text without brackets for example: "[Board] finished with result [Result]".

#### 5.1.4 Test Panel

A **PMS.Cliptest.ViewClient** can display multiple test panels. The amount of panels depends on the count of monitors that are connected to the computer. The layout and the position of the test panels can be configured in the settings tab **Monitor**.



The test panel contains three main parts. Information about the current test is displayed on the top of each test panel. In the middle, the selected layout can be seen. The bottom contains the remaining time for the current test.

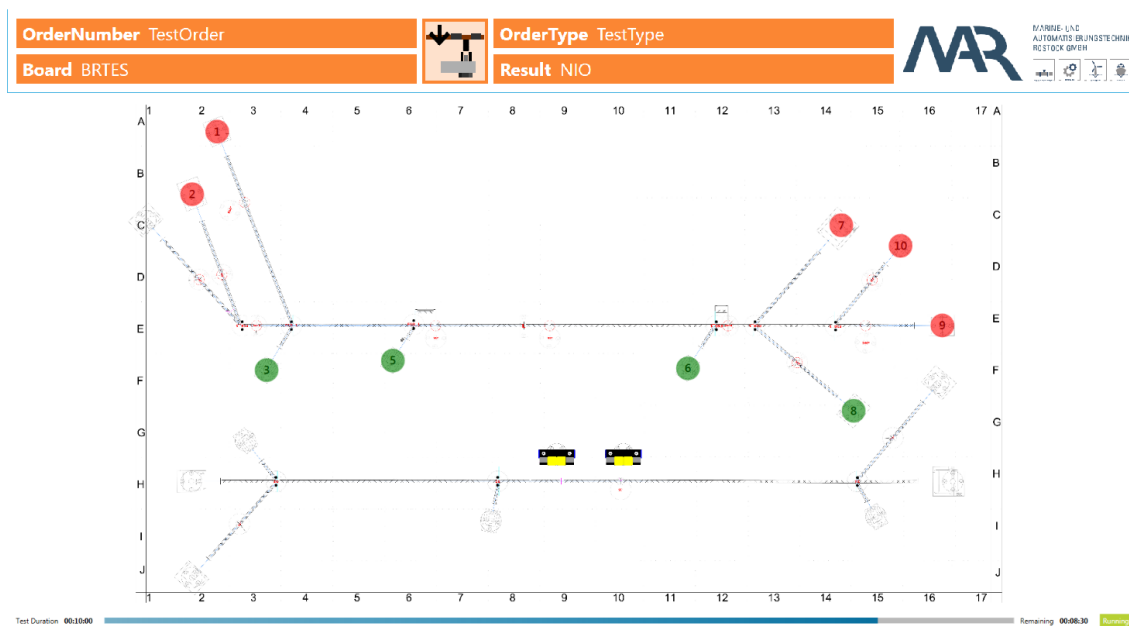
##### 5.1.4.1 List layout

The list layout displays all clips in a list with a different number of columns. Each column of this layout can be configured separately. To change the displayed clips, the grid view headers enable sorting, filtering and grouping is enabled. To save or discard the changes or reset to defaults use the context menu (right mouse click) of the grid view header. The settings for each grid view are handled separately (one save action is required for each changed grid view or you can use the save option for the window). Attention: the save option will save all unsaved changes from the settings. E.g. if the color of the clips is changed, it is not necessary to save this operation and press the save changes button in the context menu, as all other changed settings will be applied. If the Load Defaults option is chosen, the configuration will be reset, but not automatically saved. The grouping panel and editing options for the displayed columns can also be shown in the context menu. The context menu is opened by using a right mouse button click on the grid view column headers. To disable the context menu, uncheck the option **Enable filter and zoom modification** in the settings tab **Display**.

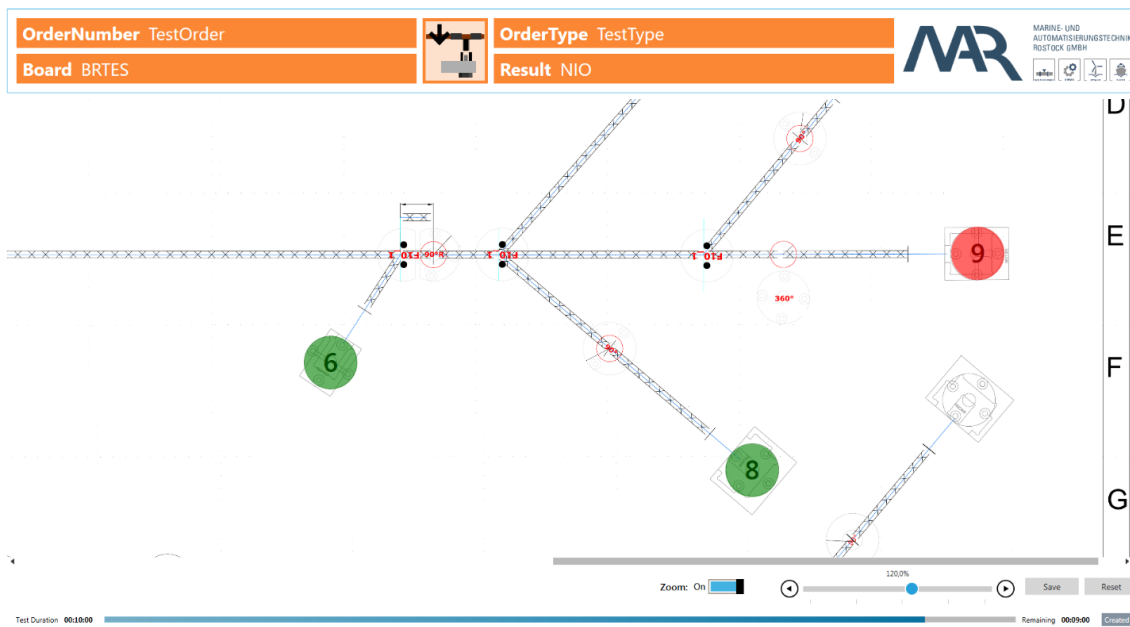
| LED |   |  |    |
|-----|---|--|----|
| 500 |   |  | 6  |
| 499 |   |  |    |
| 498 |   |  |    |
| 497 |   |  | 3  |
| 496 | T |  | 2  |
| 495 | T |  | 1  |
| 494 | S |  | 26 |

#### 5.1.4.2 Image layout

The image layout displays all clips on a configurable board layout. The style of the clips is configurable in the settings.



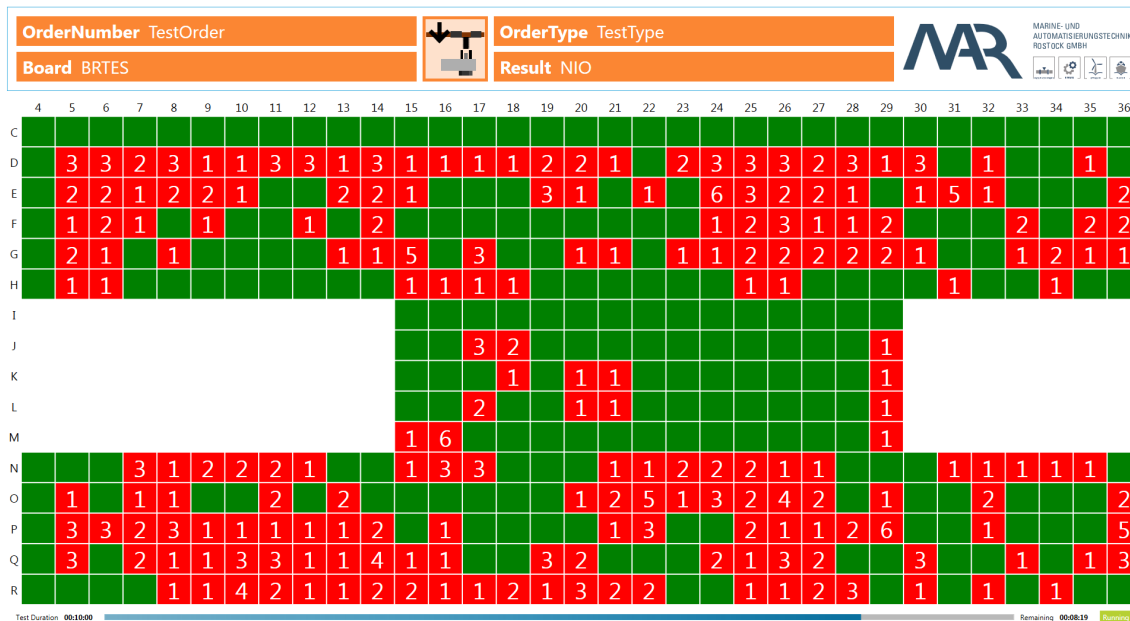
Zooming and scrolling single parts of the layout is enabled. To open the zoom control, the background must be double clicked. The zoom control is placed on the right bottom corner. Set up the requested section and **Save** the configuration for the next test.



**Important:** To use this layout, each board must be prepared with the **PMS.Cliptest.MasterDataManager**

#### 5.1.4.3 Matrix layout

The matrix layout displays all clips in a grid. Each cell shows the count of the missing clips. The style of the cells is configurable in the settings.



**Important:** To use this layout, each board must be prepared with the **PMS.Cliptest.MasterDataManager**.

#### 5.1.5 Dialogs

This chapter provides an overview about possible dialogs which might appear in different scenarios.

##### 5.1.5.1 Quality Confirmation

This dialog appears when quality related issues arise during a test. Currently the state change of any clip from state NeedlessPressed to NeedlessNotPressed during a test is treated as quality issue. **PMS.ClipTest** tracks such events and if the option "Enable Quality Confirmation" is enabled this dialog will appear on all application windows. The dialog will ask for a user of group "Quality" to login and confirm all issues forcing this user to also add a comment to each issue. If the user doesn't confirm all of the issues, the test will be finished with result NIO.

Quality Related Events

Please confirm the following issues during the test

Username

QualityUser

Password

•

| <input type="checkbox"/> Text   | Timestamp             | Comment                     |
|---|-----------------------|-----------------------------|
| <input type="checkbox"/> Clip 4 changed state from NeedlessPressed to NeedlessNotPressed            | 2/27/2019 10:36:15 AM | Masterdata wrong            |
| <input type="checkbox"/> Clip 5 changed state from NeedlessPressed to NeedlessNotPressed            | 2/27/2019 10:36:22 AM | Masterdata wrong            |
| <input type="checkbox"/> Clip 6 changed state from NeedlessPressed to NeedlessNotPressed            | 2/27/2019 10:36:29 AM | Masterdata wrong            |
| <input checked="" type="checkbox"/> Clip 3 changed state from NeedlessPressed to NeedlessNotPressed | 2/27/2019 10:36:33 AM | Clip broken -fixed on board |
| <input checked="" type="checkbox"/> Clip 3 changed state from NeedlessPressed to NeedlessNotPressed | 2/27/2019 10:36:36 AM | Clip broken -fixed on board |
| <input checked="" type="checkbox"/> Clip 3 changed state from NeedlessPressed to NeedlessNotPressed | 2/27/2019 10:36:38 AM | Clip broken -fixed on board |

Comment

Clip broken -fixed on board

Apply

Abort Test


Confirm


5.1.5.2 Label confirmation

This dialog appears if the option "Confirm Printout" is enabled in a printing event. It is supposed to verify the printed barcode. Therefore the value in the configuration should be set to the same value which is the barcode content. If the scan doesn't match the configured value the dialog won't close and the barcode can be scanned again. The print button will print the label again in case the printer has a problem.

Please scan printed barcode to confirm

Please scan the label to confirm





Print

5.1.6 Statistic Files Structure

This chapter explains the content of the exported statistic files by PMS.ClipTest.ViewClient.

5.1.6.1 Xml File

The xml file starts with general test information under the "Information" xml node followed by information for each clip under "ClipsToTest" xml node. General test information is structured as follows:

| Node | Description                            |
|------|--|
| Id   | Internal test id for reference purpose |

|                |   |
|----------------|---|
| BoardName      | Name of the tested board  |
| Workplace      | Workplace of the test   |
| VariantName    | Name of the tested Variant in case of a Cliptest  |
| Comment        | Variant comment from masterdata   |
| Operator       | Logged in operator who started the test   |
| Parameter      | List of known variant parameters from masterdata  |
| VariantName    | Name of the tested Variant in case of a Cliptest  |
| Type           | EmptyTest or ClipTest   |
| Steering       | Steering value of a test. Supplied by some production interfaces.   |
| KSK            | Harness which was tested  |
| TestMode       | Test mode of the test   |
| LatencyPeriod  | Minimum number of seconds clips need to be pressed during pre-process to be IO. 0 if no minimum duration is required. |
| TestStartTime  | Timestamp when test started   |
| TestFinishTime | Timestamp when test finished  |
| Result         | Result of the test, can be IO or NIO  |
| Status         | Status of the test - usually Finished   |
| Aborted        | If true test was aborted otherwise false  |
| CriticalEvents | Shows the list of Critical Events during the test   |
| ErrorCode      | Additional information regarding the test result  |
| TimedOut       | If test ran out of time this value is true, otherwise false   |
| TimeOutSeconds | Number of seconds of the configured test timeout  |

Clip information is structured as follows:

| Node             | Description                                 |
|------------------|---|
| Id               | Internal clip id for reference purpose      |
| SwitchKey_Number | Physical address of the clip                |
| SwitchKey_Status | True if pressed, false if not pressed       |
| ForceTest        | ForceTest value from masterdata (PinStatus) |
| LED              | Led number of the clip (masterdata)         |
| Description      | Clip description from masterdata            |

|                         |  |
|-------------------------|--|
| Name                    | Clip name from masterdata  |
| X_Axis                  | X coordinate from masterdata   |
| Y_Axis                  | Y coordinate from masterdata   |
| Configured              | False if clip is unconfigured (required by test-list but not in masterdata for board) otherwise true                 |
| Requested               | True if clip is part of the test-list  |
| PreProcess_Pressed      | Used in test-mode Unpressed Clips. True if clip was pressed/IO before the test started otherwise false.              |
| PreProcess_Duration     | Used in test-mode Unpressed Clips. Number of seconds how long the clips was pressed before the test started.         |
| PreProcess_Status       | Used in test-mode Unpressed Clips. Test-Status of the clip before the test started.                                  |
| ClipTest_Pressed        | True if clip was pressed during the test otherwise false   |
| ClipTest_Duration       | Number of seconds how long clip was pressed during test (maximum)  |
| ClipTest_Status         | Logical Test-Status during the test  |
| Status                  | Relevant Test-Status of the clip where PreProcess_Status and ClipTest_Status are combined according to the test-mode |
| PreProcess_PressTime    | Used in test-mode Unpressed Clips. Contains the timestamp when the clip was pressed before the test started.         |
| ClipTest_PressTime      | The timestamp when the clip was pressed (latest) during the test   |
| ClipTest_TriggerCount   | Number how often the clip was pressed during the test  |
| PreProcess_TriggerCount | Used in test-mode Unpressed Clips. Number how often the clip was pressed before the test started.                    |
| ClipGroup               | Name of the assigned clip group from masterdata  |

#### 5.1.6.2 Csv File

| Node             | Description                                 |
|------------------|---|
| ID               | Internal test id for reference purpose      |
| Workplace        | Workplace of the test                       |
| Board            | Name of the tested board                    |
| KSK              | Harness which was tested                    |
| SwitchKey_Number | Physical address of the clip                |
| SwitchKey_Status | True if pressed, false if not pressed       |
| ForceTest        | ForceTest value from masterdata (PinStatus) |



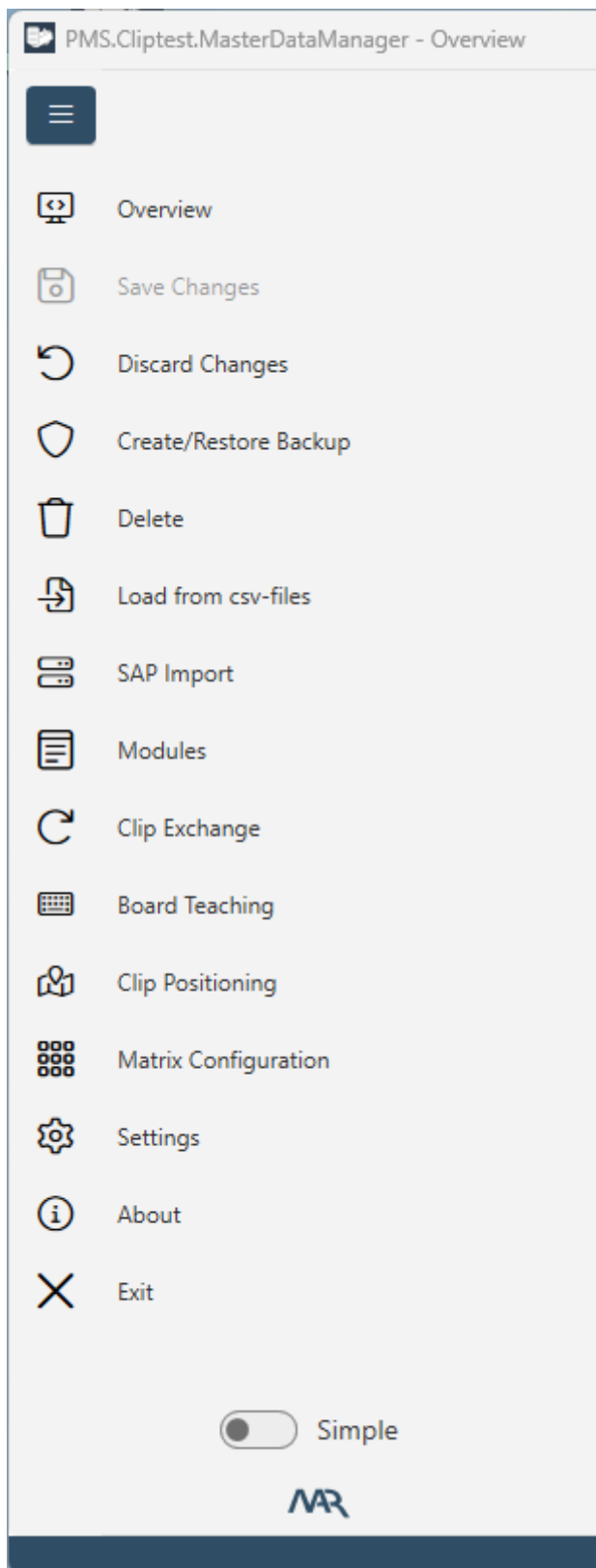
|                         |  |
|-------------------------|--|
| LED                     | Led number of the clip (masterdata)  |
| Description             | Clip description from masterdata   |
| X_Axis                  | X coordinate from masterdata   |
| Y_Axis                  | Y coordinate from masterdata   |
| Configured              | False if clip is unconfigured (required by test-list but not in masterdata for board) otherwise true                 |
| Requested               | True if clip is part of the test-list  |
| PreProcess_Pressed      | Used in test-mode Unpressed Clips. True if clip was pressed/IO before the test started otherwise false.              |
| PreProcess_PressTime    | Used in test-mode Unpressed Clips. Contains the timestamp when the clip was pressed before the test started.         |
| PreProcess_Duration     | Used in test-mode Unpressed Clips. Number of seconds how long the clips was pressed before the test started.         |
| PreProcess_Status       | Used in test-mode Unpressed Clips. Test-Status of the clip before the test started.                                  |
| ClipTest_Pressed        | True if clip was pressed during the test otherwise false   |
| ClipTest_PressTime      | The timestamp when the clip was pressed (latest) during the test   |
| ClipTest_Duration       | Number of seconds how long clip was pressed during test (maximum)  |
| ClipTest_Status         | Logical Test-Status during the test  |
| Status                  | Relevant Test-Status of the clip where PreProcess_Status and ClipTest_Status are combined according to the test-mode |
| StartTime               | Timestamp when test started  |
| FinishTime              | Timestamp when test finished   |
| PreProcess_TriggerCount | Used in test-mode Unpressed Clips. Number how often the clip was pressed before the test started.                    |
| Cliptest_TriggerCount   | Number how often the clip was pressed during the test  |
| TestType                | EmptyTest or ClipTest  |
| TestResult              | Result of the test, can be IO or NIO   |
| Aborted                 | If true test was aborted otherwise false   |
| Name                    | Clip name from masterdata  |

## 5.2 PMS.Cliptest.MasterDataManager

The main window of **PMS.Cliptest.MasterDataManager** (short: MasterDataManager) contains different views. The following chapters describe all views.

### 5.2.1 Menu

Access is provided to all features of the ***MasterDataManager*** in the menu. To open the main menu, the left top corner on the icon must be clicked (three horizontal lines). To switch between different views the required icon must be clicked. To close the menu one can switch to another view or use the top icon of the menu.



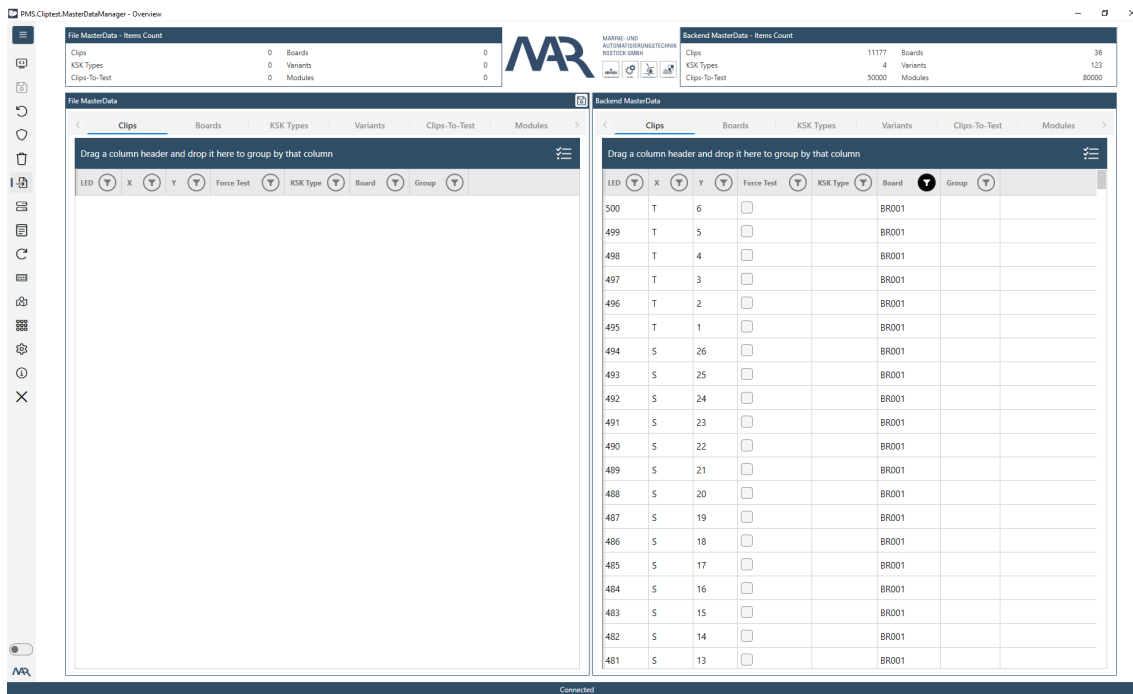
The master data can be modified with the button **Master Data**. A click on it opens a submenu. In this submenu the user is able to go back to the Overview, save or discard outstanding changes, delete the master data or create /

restore a backup from the complete master data (including configurations and traceability data). The menu option **Load from csv files** can be used to import new master data from csv-files. The features **Clip Exchange** and **Board Teaching** are only available if you have a license for the MasterDataManager Extensions. The **Board Designer** is used to configure the image and the matrix layouts of the test panel.

| Button                | Action   |
|-----------------------|--|
| Overview              | Use this button to navigate to the overview.   |
| Save Changes          | Saves the outstanding changes of the master data   |
| Discard Changes       | Reverts the outstanding changes of the master data   |
| Create/Restore backup | Opens the backup feature of the master data manager  |
| Delete                | Deletes the existing master data from the backend  |
| Load from csv-files   | Imports the master data from the configured csv files  |
| SAP Import            | Opens the SAP import feature which can be used to create variants from SAP master data       |
| Modules               | Opens the management view for modules  |
| Clip Exchange         | Opens the clip exchange view (license required)  |
| Board Teaching        | Opens the board teaching view to create new master data files (license required)             |
| Clip positioning      | Opens the configuration view for the image layout of the Operator Windows in the ViewClient  |
| Matrix Configuration  | Opens the configuration view for the matrix layout of the Operator Windows in the ViewClient |
| Settings              | Opens the settings of the MasterDataManager  |
| About                 | Information about the product and licensing can be found in the above view.                  |
| Exit                  | Administrator privileges are required to close the PMS.Cliptest (configurable).ViewClient.   |

### 5.2.2 Overview View

This is the initial view of **PMS.Cliptest.MasterDataManager**. All information about the file and database master data can be seen here. The left side shows the file master data which can be modified with the user interface or by importing data from csv-files. All changes are temporary and will be not submitted automatically. On the right side is visualized in the database.



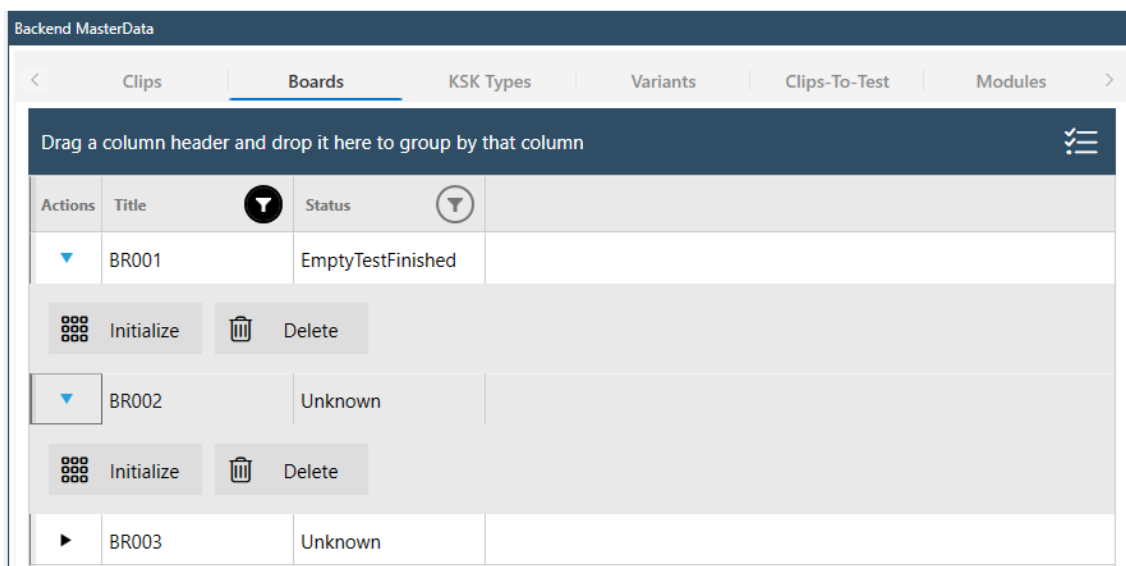
The user interface enables editing the coordinates and the force test flag for the clips. It is also possible to delete unused data (delete key on the keyboard). Some elements of the master data can only be deleted, if the depending element is also deleted. Before an element is deleted, a message is displayed showing how many depending elements are impacted by this operation. All changes are temporary and will be not saved automatically. The save master data feature can be found in the menu option Master Data. Attention: the save process cannot be undone.

Before the save operation is performed, a backup for the existing csv-files is created by the MasterDataManager in the import folders. After finishing the save operation, a new csv-file with the current data from the database is generated. The following table describes the generated files for each file type (**auto-save mechanism**).

| File Type   | Result   |
|-------------|--|
| Cliplist    | a separate file for each variant                           |
| Coordinates | a separate file for each board                             |
| PinStatus   | a separate file for each board                             |
| Variants    | one file which contains all variants with their parameters |
| Modules     | one file which contains all modules                        |

### 5.2.2.1 Board operations

The database master data enables removal of unused boards at the tab page for the board. It is important to note that this operation cannot be undone. After new master data is imported, all clips on a board are initialized as unknown because the current state is not available. If an attempt is made to run an Empty Test for a new imported board, it will fail because all clips are unknown. To solve this problem, the **Initialize Board** option can be pressed (otherwise every clip must be pressed at least once for it to be initialized). The initialize process can only be performed if no test is running on the selected board. These operations are only possible when the file master data is empty.



### 5.2.3 Settings View

The configuration can be opened by using the settings menu option. The settings view has five main configuration sections which are organized in tab pages.

#### 5.2.3.1 General

On tab page **General**, the connection of the service can be initialized and the authorization and the logging behavior configured. Only the hostname (or IP-Address) and the port are required for the Service connection. If the user is not sure about what to setup, the **Discover** button, which searches for all **PMS.ClipTest.Service** applications in the network can be used. The input can be tested with the **Test** button.

The log- / debug-presets can be configured in the Logging group box on this tab page. Select the Log Path and the Log Min Priority.

| Log Debug        | Settings   |
|------------------|--|
| Log path         | Enter a Path where PMS.ClipTest.MasterDataManager should save Log-Files. The default value is "{InstallationPath}/Logs". |
| Min Log Priority | Select the priority of Log-Events to be shown in Log-window and will be written to the log file.                         |

#### 5.2.3.2 File Paths

The folders containing the master data files can be configured on the second tab page (**File Paths**). A separate folder must be configured for each file type (Cliplist, Coordinates, PinStatus, Variants, Modules). All files in the configured folder will be used for importing the master data. Files in subfolders will be not considered. These folders will be also used for exporting the saved master data.

#### 5.2.3.3 CSV Settings

The tab page **CSV Settings** can be used to configure the file separators and comment tokens for importing and exporting master data. The configured values in the group **Import** will be used for importing master data. Multiple separators can be selected if more than one separator is needed. The comment token can be used for comments in the csv-files (these lines will be ignored)

The values in the group **Export** will be used for the files which will be generated by the auto-save mechanism and the Board Teaching view.

#### 5.2.3.4 Teaching

The page **Teaching** is only relevant for the **PMS.Cliptest.MasterDataManager** Extensions. These four configurable filters can be used to get the filtered clip press events. The option **Min Press Time** defines how long a clip must be pressed to be displayed in the **Board Teaching** mode (0 means no Min Press Time). The option **Max Press Time** can be used to get only clip which are no longer pressed than the configured value. The value Press Timespan describes the timespan in which the number of press events (**Min Press Count**) have to be recognized.

#### 5.2.3.5 Scanner

In the Clip Exchange process, clips can be selected by scanning a barcode. The scanner can be configured at the tab page **Scanner**.

The following table describes all settings of the connected RS232 scanner.

| Setting        | Description   |
|----------------|---|
| Enable scanner | A unique name which describes the scanner.  |
| Baudrate       | Select the baudrate the scanner works with. The default value is 9600.  |
| Parity         | Enter the parity the scanner works with. The default value is NONE.   |
| Stopbits       | Enter the count of stopbits the scanner works with. The default value is one.   |
| Databits       | Enter the count of databits the scanner works with. The default value is eight data bits.   |
| Prefix Length  | Number of characters which are used for the address stamping. The default value is 0 (no address stamping enabled).   |
| New Line       | Scanners are sending the scanned value with a terminator in data stream. This terminator marks the end of the scanned data. This terminator must be set to make <i>PMS.Cliptest.ViewClient</i> correctly evaluate scans of the connected scanner. Normal scanners do use a CRLF (Carriage Return Line Feed) as terminator so this is the default value. Since special chars like CR (carriage return) cannot be entered with a keyboard, the ASCII-list can be used. Enter the decimal values of the chars for use as a terminator, e.g. "13" for CR or "10" for LF. The ASCII-table can be used if the user is not sure. The default value is CRLF corresponding to ASCII{13} + ASCII{10}. |

#### 5.2.4 Manage Master Data

The menu option **Master Data** can be used to manage the currently used data.

After the import process, the user is asked to save the imported data. If the prompt to perform the saving process is declined, it can be performed manually with the **Save Changes** button. The **Discard changes** button allows one to discard all changes since the last save action. With the **Create/Restore Backup** button, the backup mechanism of **PMS.Cliptest** (for further information see 5.2.4.1 Backup) can be used. Corrupt or old master data can be removed with the **Delete** button (*Attention: the complete master data will be deleted – this process should be not performed if any test is running – create a backup of the database before running this operation*).

##### 5.2.4.1 Backup

Back ups can be created or restored by using the backup feature. The components for the master data that should be included in the backup can also be selected. The following table describes each option.

| Option            | Description   |
|-------------------|---|
| Board data        | The board data is composed of the data which was imported from the csv files and additionally the configured clip positions from image layout.  |
| Configurations    | The configurations can only be added to the backup if the board data is selected. This backup option saves the settings of all ViewClients and the configured matrix- and image-layouts.  |
| Traceability data | The traceability data option adds all data which are required for the clip reporting and the traceability of harnesses to the backup. Attention: If this option is enabled, the backup process may takes a long period of time and takes a lot of disc space. |

For the backup to be created or restored, the components which should be included and press **Create Backup** or **Restore Backup** button. After clicking on the create button a new backup archive will be created in the destination folder. To avoid data loss, the backup archive must be copied to another device. Created backups will be not removed automatically. Performing a manual cleanup of the configured destination folder is essential. The latest backup will be restored in the restore case.

The restore operation can only be performed if the database is empty (deleting only the master data is not enough).

Create Backup

Please select the components which should be included in the backup.

Please keep in mind that the traceability data can contains a lot of items.

☐ Board data  
☐ Configurations  
☐ Traceability data

Create Backup

Restore Backup

Please select the components which should be restored from the last backup. If a selected component doesn't exists in the backup, all other components will be not affected. Please check the log of the Service if some data was not restored.

☐ Board data  
☐ Configurations  
☐ Traceability data

Restore Backup

During the backup process, **PMS.Cliptest** might need more free disc space than the created backup archive. The directory for the backups is configurable in the **PMS.Cliptest.ServiceConfigurator** (see chapter [5.3.2.1 General](#)).

#### 5.2.4.1.1 AUTOMATIC BACKUP

During the shut down process, the service application tries to create a backup of the **Board data** and the **Configuration**. The **traceability data** will be not included in the automatic backup. The backup is only created if the backup path is configured. Further information for the backup can be found in chapter [5.3.2.1 General](#). After creating a new automatic backup, the previous automatic backup will be deleted to avoid increasing disc usage.

#### 5.2.5 Import Master Data

The menu option **Load from csv-files** can be used to import new master data from csv-files use (The requirements for these files are described in section 4 Program Master Data Requirements). By clicking this menu option, the import process from the files is started. After this process is finished, the imported data can be saved to the database master data. There are two options for saving the outstanding changes; by using the question dialog after importing or by checking the imported data using the save changes action in the submenu of Master Data. New imported clips are initialized as unknown because the current state cannot be determined. All other clips whose state is known will remain in their state.



After saving the imported data to the database, a backup is created by the MasterDataManager. If invalid data is added, the backup can be restored and the import process run again. These backup files remain in the import folder until deleted by the user. It is important to Cleanup the directory manually if master data is imported multiple times.

The import process is performed per board. If a board doesn't exist in the database master data, all information from the csv files will be imported into the database. If a board already exists in the database, the information from the csv-files will be used. That means if the csv-files contain more or less clips than the database the information from the file will be used. Boards which are in the database but not in the files will be not changed.

If a new board is added to the master data, all clips are marked as unknown. Each clip must be pressed at least one time to set it to the right state. To avoid this process, the initialize board feature can be used. This operation sets all clip states as unpressed. It is important to use this option only if you are sure that all clips are unpressed, otherwise you invalid test results will be obtained. For further information see chapter [5.2.2.1 Board operations](#)

## 5.2.6 SAP Import

The SAP Import feature can be used to import harnesses from SAP and run a clip- or empty-test for the specific harness without a connection to a production server. The required information for the tests is exchanged via csv files. The traceability data for successful tests is exported as csv-files (see chapter [5.2.6.3 Traceability data export](#) Traceability data export). This data has to be imported via other tools into the corresponding tables in the production server. This feature can only be used if there are no unsaved changes in the master data.

### 5.2.6.1 Import files

Three files are required for the import. All files and their required columns are described in the following chapters. To be independent for the column order, the first line in each file must be the column headers.

#### ZCREA FILE

The ZCrea file contains entries for each single harness. Usually the export file from SAP contains a couple of columns, but only two columns are relevant for the import.

| Column name | Column type | Comment  |
|-------------|-------------|--|
| KSK         | String      | This column is used to identify a single harness.  |
| LIUMF       | String      | This column describes the KSK Type of the harness. |

#### ZCREA MODULE FILE

In the ZCrea Module file are all required modules for a harness specified. This file contains all modules for all harnesses of the ZCrea file. The KSK column is used to assign the modules to a harness.

| Column name | Column type | Comment  |
|-------------|-------------|--|
| KSK         | String      | This column is used to assign a module to a single harness.  |
| MODULE      | String      | This column describes the modules which are required for the harness. The module is assigned to a specific clip with the OGC Master Data file. |

#### OGC MASTER DATA FILE

The OGC Master Data file describes the relationship between the module and the LED number of a clip. It is possible that the same LED is used for multiple modules

| Column name | Column type | Comment |
|-------------|-------------|---------|
|-------------|-------------|---------|

|        |        |  |
|--------|--------|--|
| LED    | Short  | This column describes the LED of a clip.                   |
| Module | String | This column is used to assign the modules to a LED number. |

#### 5.2.6.2 Workflow

The following section describes the general workflow of the import process. Firstly, the required data is exported from SAP as a csv files. After that, the import process can be started by selecting the files and separators.

The file can be selected by typing the path or using the file selection dialog behind the input (Button with caption "..."). By default, the separator for the files is a semicolon. If another separator is used, it must be configured to continue. The path selection checks the specified file and tries to find the required columns depending on the type of the file (ZCrea, ZCrea Module, OGC Master Data). If the file validation fails, the import process is interrupted. It is important to check the error messages in the path input fields, to find out what the reason for the error is. If an error is displayed stating that a column is missing, the first line of the document doesn't contain the column names or the columns name are not valid. The default values for the column headers are described in the Import files section.

After the files are selected, the board for generating the list of Clips-To-Test must be selected. If running the test for the specified harness is required, the data for each board must be selected separately.

The application checks the data during the process of reading the data from the files. If there are any errors, a desktop alert is sent and detailed messages for the modules are then found in the section for the import data.

Two options are provided for saving the imported data to the database master data. The first option is to open the details section of a single harness and press the import button or select all harnesses that should be imported and use Import selected button below the data grid (selected harnesses which are not valid will be not imported). The import process will create a Variant for each harness with a KSK Type for the LIUMF. The list of Clips-To-Test will be generated by searching all clips on the selected board which have a LED from the import data.

Remarks:

- If there are any clips in the database which are only available for a specific KSK Type and their LED numbers is used during the SAP Import, an error is sent out because the clip cannot be assigned.
- If an LED is used multiple times on the selected board, all clips with these LEDs have to be pressed to complete a cliptest.

#### 5.2.6.3 Traceability data export

To ensure the traceability of completed tests, data can be exported into csv-files. By default, exporting these files is not enabled. This option can be enabled in the **PMS.Cliptest.ServiceConfigurator** (ref.: chapter [5.3.2.1 General](#)).

Remarks:

- Please ensure that the generated files will be included in a backup.
- If you enable the export option over a long time, ensure that you have enough free disc space for the files.

#### 5.2.7 Module Management

There are three different types to decide which clips should be tested during a cliptest. The configured file master data can be used with a variant, a higher-level production system or a barcode scan, which contains the cliplist. One might need additional master data for the last two options. This data can be managed with the MasterDataManager or by an import via modules master data files. The module management can be opened with the menu option.

A list of all configured modules is available for use. Each module has a list of LED numbers in the details section. The details can be seen by selecting a module. The number of the module can be changed by selecting the module and clicking in the number column. The column will change to an editor field. It is important to consider that each module number must be unique, otherwise the changes cannot be saved. All changes will be performed temporarily,

press the Save changes button to submit the changes to the database or Discard changes to reject the changes.  
Attention: the actions cannot be undone.

If a new module must be added, the **Add** button in the bottom left corner can be used. A module, which is not used anymore, can be deleted by using the **Delete Select** button. Opening the details section will enable the LED numbers of a module. New LED numbers can be added with the **Add** button and removed by selecting a LED number and click on the **Delete selected** button. Similar to the module number you can change the LED number with a double click on the value in the LED column.

The column validation provides information if the configuration is not valid or there are some remarks. Remarks will be shown as orange exclamation marks with a tooltip message. There will be also red exclamation marks; these are displayed if the configuration is not valid. In that case, the configuration cannot be saved. All errors are fixed to save the configuration.

### 5.2.8 Clip Exchange View

This feature is only available if the **MasterDataManagerExtensions** are licensed.

The address of an existing clip can be changed by using a wizard. In the first step of the wizard, the user is asked to select the modes, how to select the clips or addresses that must be replaced. The following table describes the available clip selection modes.

| Mode                 | Old Clip   | New Clip   |
|----------------------|--|--|
| Manual Input         | Type the address of the clip, which already exists in the MasterData.  | Type the address of new the clip. Addresses can be selected, which are already in use or not match with the physical address, but this could cause unintended behaviors. |
| Press Selection      | The clip, which already exists in the MasterData, can be pressed and selected from the list. (To configure the press selection, the tab page Teaching in the settings can be used) | Press the new clip. Clips which are already in use can be pressed, but this could cause unintended behaviors.  |
| Scan Selection       | The address of the clip which already exists in the MasterData can be scanned.   | The address of the clip can be scanned. Clips which are already in use can be scanned, but this could cause unintended behaviors.  |
| MasterData Selection | A clip from the list of the MasterData can be selected.  | A clip can be selected from the list of the MasterData.  |

After selecting a clip, the exchange mode can be chosen. Press can be used if a (not damaged) clip must be replaced or a Manual used for a complete clip exchange. If Press is selected, the next step is activating the listening process and pressing the new clip. The clips which fulfill the configured values in the teaching section of the settings will be displayed in a list. Stop the listening process and select an address to proceed. Before the clip exchange is executed, the selected clip can be checked. The exchange process can be started by pushing the Exchange button.

### 5.2.9 Board Teaching View

This feature is only available if the **MasterDataManager** Extensions are licensed.

To start the **Board Teaching** process, one must enter a **Board** and activate it with the **Change** Button. If these requirements are fulfilled, the teaching process can be started with the **Start** button. Similar to the board you can

change the **KSK** Type. The configured KSK Type will only be used if the option **Use current KSK Type** is checked. Otherwise the KSK Type will be empty.

Information about the last clip will be displayed in the center of the screen. The coordinates and the LED of the next clip can be set below to the last clip. A list of the current taught clips can be found on the bottom of this view. The coordinates and cliplist files will be generated and saved in the configured input folders by clicking the **Save** button. The **Reset** button resets all settings to the default.

### 5.2.10 Board Designer View

To configure the board layouts, the menu option **Board Designer** can be used. In the submenu of board designer, a selection can be made between modes.

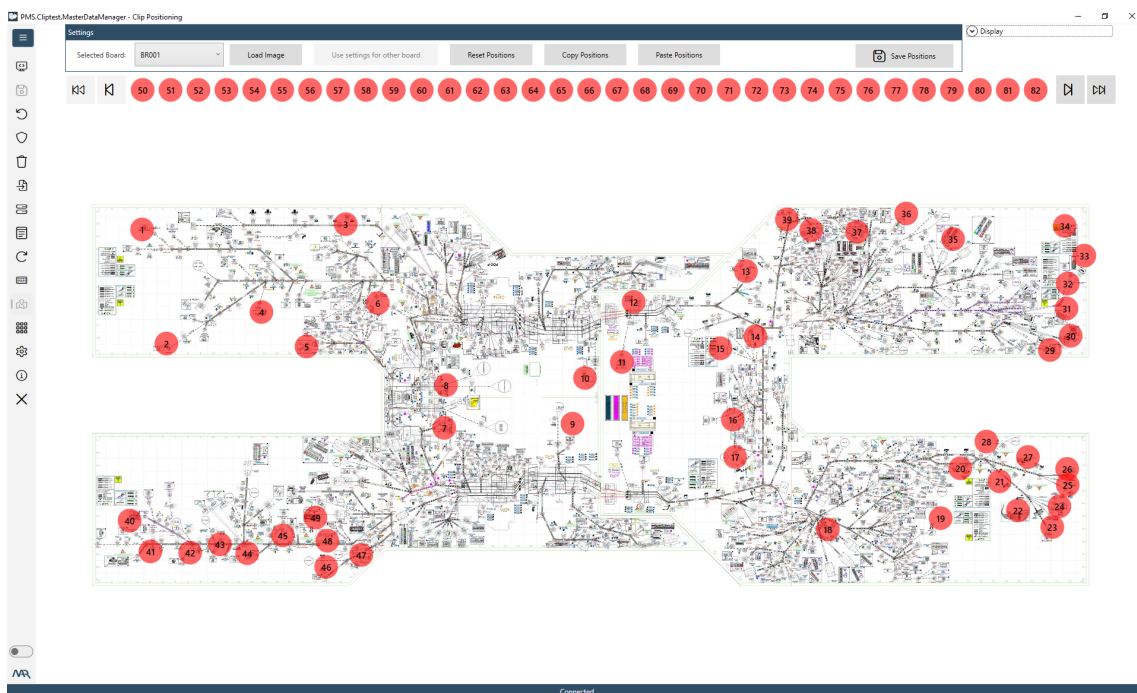
The first mode is **Clip Positioning**. This can be used to configure the image layouts. The other mode is **Matrix Configuration**. It enables the configuration of the matrix layout.

#### 5.2.10.1 Clip Positioning

Three steps are required to configure the image layout. The board for configuration must first be selected. Then the background image loaded, if the current image doesn't match. After selecting the board layout, all clips can be placed by drag and drop. If the accuracy of the positions must be increased, the zoom control and the diameter settings can be used. To display or hide the zoom control, the background image can be clicked with the middle mouse button. Until the clips are not positioned, they can be placed via quick positioning. A double click set the next available clip to the position of the click. To complete the positioning process save the positions.

Each clip can have a different icon. The icon can be change via the context menu or during the initial positioning process. The values for the initial process can be configured in the settings area and have to be enabled with a toggle switch. By default this option is disabled.

If another board has the same positions and same LEDs, they can be cloned by using **use settings for the other board**. The clone feature will only work if the source and destination board have the same LED numbers and each LED number is unique on the boards



### 5.2.10.2 Matrix Configuration

Three steps are required to configure the matrix layout. The board for configuration must first be selected. After selecting the board, columns and rows for sections where no clips are placed must be added. Adding columns and rows is required if the area of physical coordinates is greater than the area of the master data coordinates.

For example, a board has physical coordinates for the columns from 1 to 25 and for the rows from A to J. The imported master data contains only clips with column coordinates between 5 and 25. In this case one can add the missing columns by clicking the items in the corners.

The green cells contain clips and the gray cells are empty. If some cells are not on the board, they can be excluded by clicking. Excluded cells are displayed with a white brush. Excluding cells is not required for using this test panel layout. It is only a feature used to visualize the board with its physical dimensions.

**Important:** If the user must configure the layout for a specific board, the coordinates of these board have to fulfill the following two requirements.

- All coordinates are configured
- The coordinates for each direction (X or Y) are of the same type
  - a. only letters (between A and Z) or
  - b. only numbers (between 0 and 99)

## 5.3 PMS.Cliptest.ServiceConfigurator

The user is able to modify the settings of the **PMS.Cliptest.Service** on the machine where the **ServiceConfigurator** is running with the **PMS.Cliptest.ServiceConfigurator** (short: ServiceConfigurator). If the security option is enabled, the start screen of **ServiceConfigurator** is the **Login** view. The configured password must be entered for one to proceed. The password protection of the service configuration can be changed via the **Service Connection** menu option.

### 5.3.1 ServiceConfigurator options

The ServiceConfigurator has a few options. To configure them, the options view of the ServiceConfigurator must be opened by activating the menu in the left top corner and selecting **Service Connection**.

In the first section of the options view, the service notification can be enabled or disabled. If the feature is enabled, the server configurator notifies the service, that the settings were changed. By default this option is enabled. The configuration of host and port is similar to the other applications. Please note that the settings of a service on which the service configurator is running can only be configured.

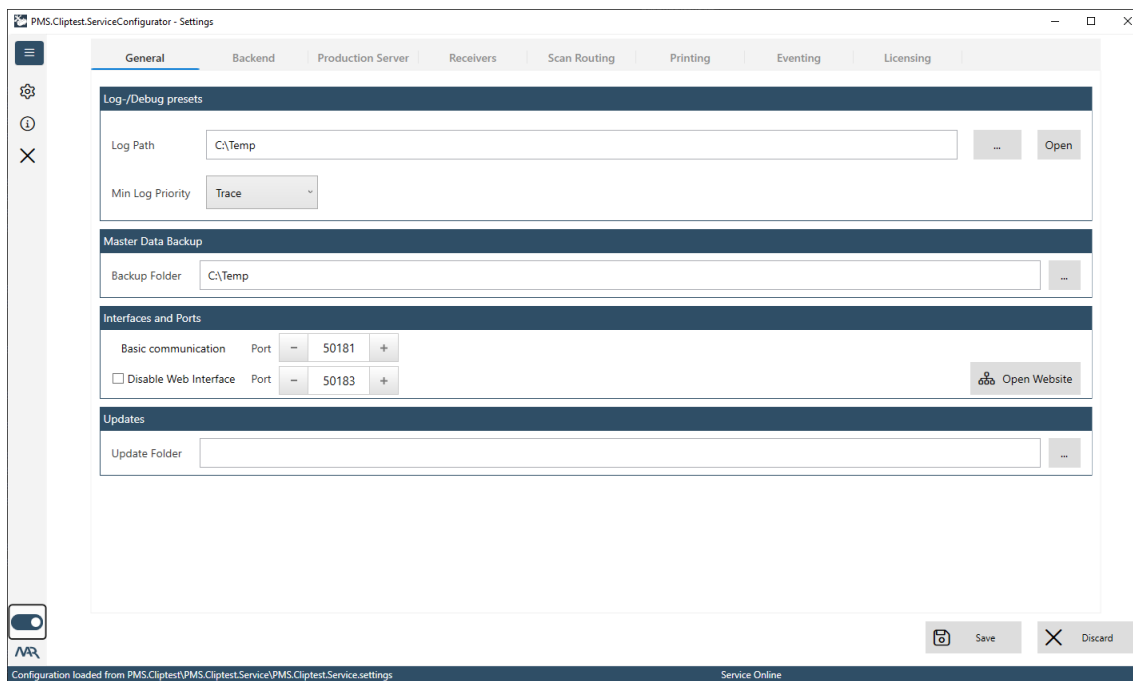
The authorization can be activated in the options of the ServiceConfigurator. If this option is checked, a password for changing the settings can be entered. The password, which is required to open the configuration, can be configured here.

### 5.3.2 Configuration

The configuration view is divided into several tab pages. The following chapter describes all pages. By default, not all tab pages are displayed (simple mode). The expert mode with all configuration options can be activated in the **Menu** with a toggle switch on the right side. All options, which are maybe not displayed in the simple mode are described in the documentation. When the user is logged in to modify the settings, the service connection settings can be accessed via the menu. There the authorization feature can be changed.

#### 5.3.2.1 General

At the tab page **General**, the **log- / debug presets**, the **traceability data export**, the master data backup and the communication settings (**Interfaces and Ports**) can be configured.



The log- / debug-presets can be configured in the first section in this view. The Log Path and the Log Min Priority can be selected here. The log files will be automatically moved into an archive subfolder in the configured path, if the application must be restarted or the file is bigger than 1 megabyte. The archive folder contains the last 30 log files.

| Log Settings     | Comment   |
|------------------|---|
| Log Path         | Enter a Path where PMS.Cliptest.Service should save Log-Files. The default value is "InstallationPath)/Logs". |
| Min Log Priority | Select the Priority of Log-Events to be shown in Log-window.  |

The option **Export Traceability Data** is available in the ViewClient configuration.

A folder must be selected in the group box **Master Data Backup** where the backups of the master data should be stored. Ensure that there is enough free disc space, the service has write access and the backups are safe in case of data loss.

In the last group box of the general tab page, the port configuration can be expanded for the **PMS.Cliptest** Service. The default port for the WCF communication is 45181. This port is required for the basic communication between the client applications and the service. You can enabled or disable and configure the used port. A unique port which will be not used by other applications must be used for each option. It is important to ensure that this port will not be blocked by the firewall.

A folder, which will be used for minor version updates can be configured in the group box **Updates**. This option can only be used if one is sure that the update will not affect the existing installation.

### 5.3.2.2 Backend

**The PMS.Cliptest.Service** needs a backend for its own data (masterdata storage). To configure it, the tab page **Backend** must be used. There are different types of backends, which can be used to store the master data.

Depending on the selected backend, different configuration options are available. The available backends and their corresponding configuration will be explained in the following chapters.

**PMS.Cliptest** writes data to its backend, which will be not cleaned automatically, e.g. the table *COM\_OGC.ClipLog*. If these tables must be cleaned, the cleaning process can be cleaned with the **Purge Configuration**. The setting **Interval** is only used if the start condition is configured to Run purge periodically. The duration between two purge processes can be set in minutes. By default, the value is 60 minutes. Only entries, which are older than the configured duration in days (default value 100 days) can be deleted by the purge process. The tables for performing the purge process should be performed in the list of the **Selected Tables**.

**Attention:** The purge process can only be performed while no tests are active to avoid inconsistent data.

There are different start conditions for the purge process. The purge process is disabled by default. The following table describes the available types.

| Start Condition          | Behavior  |
|--------------------------|---|
| Disabled                 | The purge process will be not performed.  |
| Run purge on next start  | The purge process will be performed one time at the next start of the <b>PMS.Cliptest.Service</b> . |
| Run purge on every start | The purge process will be performed on every start of the <b>PMS.Cliptest.Service</b> .             |
| Run purge on every close | The purge process will be performed on every close of the <b>PMS.Cliptest.Service</b> .             |
| Run purge periodical     | The purge process will be performed periodical.   |

#### 5.3.2.2.1 SQL BACKEND

This backend is based on a SQL server database. The server address and the database can easily be set. For the authentication, the **Windows Integrated Security** or a combination of username and password can be used. (**Attention:** if Windows Integrated Security is chosen, one must ensure that the local system account has access to the database or the user account which starts the service must be modified). To verify the input, the Test Connection feature can be used. If the database does not exist, it can be created at the configured SQL server.

#### 5.3.2.2.2 FILE BACKEND / DEMO BACKEND

These backends are based on files. The file and demo backends have similar behaviors regarding functionality. The demo backend has additional example data for testing scenarios.

In the default case, no configuration is required for this backend. If the storage path for these backends must be changed, the option **Use default path** and select the path must be unchecked. The option **Write Runtime Data** configures the interval about how often the runtime data will be saved from memory to the files.

#### 5.3.2.3 Production Server

**PMS.Cliptest** has the ability to be controlled by a higher level manufacturing execution system (e.g. MES). If **PMS.Cliptest** must be used in combination with a production server, the communication interface on the **Production Server** tab page must be selected. Depending on the interface of the production server, different settings must be entered. By default the production server NONE, which will be used for the standalone mode is selected.

Please contact [support@cliptest.de](mailto:support@cliptest.de) for detailed information about available interfaces

#### 5.3.2.4 Receivers

If your receivers are directly connected to the **Service**, their parameters can be configured at the **Receivers** tab page. A receiver can be selected from the top list and configured as below.

| Setting           | Description   |
|-------------------|---|
| Enable Receiver   | Enables the receiver.   |
| Name              | The Name is used to identify the receiver.  |
| COM port          | Enter the COM-Port you have connected your radio receiver to.   |
| Baudrate          | Select the baudrate the receiver works with. Short protocol receivers (TRB10_Short) should work with 57600 baud, long protocol receivers (TRB10_Long) should work with 9600 baud. The default value is 57600 baud.                |
| Parity            | Select the parity the receiver works with. Short protocol receivers as well as long protocol receivers should work with Parity NONE. The default value is NONE.   |
| Stopbits          | Select the count of stopbits the receiver works with. Short protocol receivers as well as long protocol receivers should work with one Stopbit. The default value is one.   |
| Databits          | Select the count of databits the receiver works with. Short protocol receivers as well as long protocol receivers should work with eight data bits. The default value is eight data bits.   |
| Protocol          | Select the protocol the receiver works with. Short protocol receivers are working with TRB10_Short. Long protocol receivers are working with TRB10_Long. The default value is TRB10_Short.  |
| Invert Clip Stats | Some radio receivers may invert all received clip states. Use this option to reinvert the clip states for correct processing. Use Clip debug window to obtain if this option must be activated. The default value is deactivated. |

If the receiver configuration must be checked, the test view next to the configuration can be used. The current setting can be activated by pressing **Start Receiving**. **Attention:** the test mechanism works only if the selected COM-port is not used. For example, the receiver is connected to the COM-port COM1 and is enabled, one cannot test the settings from port COM1. The service must be stopped or the receiver disabled, the settings saved and thereafter the receiver settings can be tested.

#### 5.3.2.5 Scan Routing

The ability to use multiple barcode scanners with only one cradle is provided by the **PMS.Cliptest.ViewClient**. A prefix for identification is provided by each scanner. The Address Stamp Mapping tab page is used to assign the scanner to the correct **ViewClient** (Ref.: chapter [7.1 Service Configuration](#)). Using one scanner for multiple clients is also enabled by the Address Stamp Mapping. A mapping between the source (from where the scan is performed) and one or many destinations (clients which use the scan information). This setting is also required if scanners are connected directly to the service.

##### 5.3.2.5.1 EXAMPLE CONFIGURATIONS

- Example 1
  - In the first example, one scanner is directly connected to the Service. Only one mapping must be configured to assign the scanner to a specific ViewClient. The prefix of this mapping can be optionally configured, if the scanner uses a prefix. If the scanner is connected to the Service, the option **From Client** and constant **Service**(case sensitive) must be used. The option **To Client** is



used for the **ViewClient** which should be responsible for the scans. The configured **ClientId** of the responsible **ViewClient** (attention: the **ClientId** is case sensitive) must be entered. The following table shows an example configuration.

- Configuration for ClientId ASS01\_01

| Setting     | Value    |
|-------------|----------|
| Prefix      |          |
| From Client | Service  |
| To Client   | ASS01_01 |

- Example 2

- The second example describes a configuration, which has two scanners directly connected to the Service. To identify the scanners prefixes are used. The first scanner has the prefix **A** and the other the prefix **B**. To assign the scanners to a specific workplace, two mappings must be configured. Caused by the directly to the Service connected scanners, the option **From Client** the constant **Service** (case sensitive) must be used. The following table shows an example configuration.

- Configuration for ClientId ASS01\_01

| Setting     | Value    |
|-------------|----------|
| Prefix      | A        |
| From Client | Service  |
| To Client   | ASS01_01 |

- Configuration for ClientId ASS01\_02

| Setting     | Value    |
|-------------|----------|
| Prefix      | B        |
| From Client | Service  |
| To Client   | ASS01_02 |

- Example 3

- In this example, two scanners are connected to the **ViewClient** with the **ClientId** ASS01\*01 and the ViewClient with the ClientId ASS01\*02 has no connected scanner. Prefixes are used to identify the scanners. The first scanner has the prefix **\*A\*** and the other the prefix **B**. To assign the scanners to a specific workplace, two mappings must be configured. The scanner with the prefix A is used for ClientId ASS01\*01 and the other with the prefix B for the **\*ClientId ASS01\_02\_**. The following table shows an example configuration.

- Configuration for ClientId ASS01\_01

| Setting     | Value    |
|-------------|----------|
| Prefix      | A        |
| From Client | ASS01_01 |

|           |          |
|-----------|----------|
| To Client | ASS01_01 |
|-----------|----------|

- Configuration for ClientId ASS01\_02

| Setting     | Value    |
|-------------|----------|
| Prefix      | B        |
| From Client | ASS01_01 |
| To Client   | ASS01_02 |

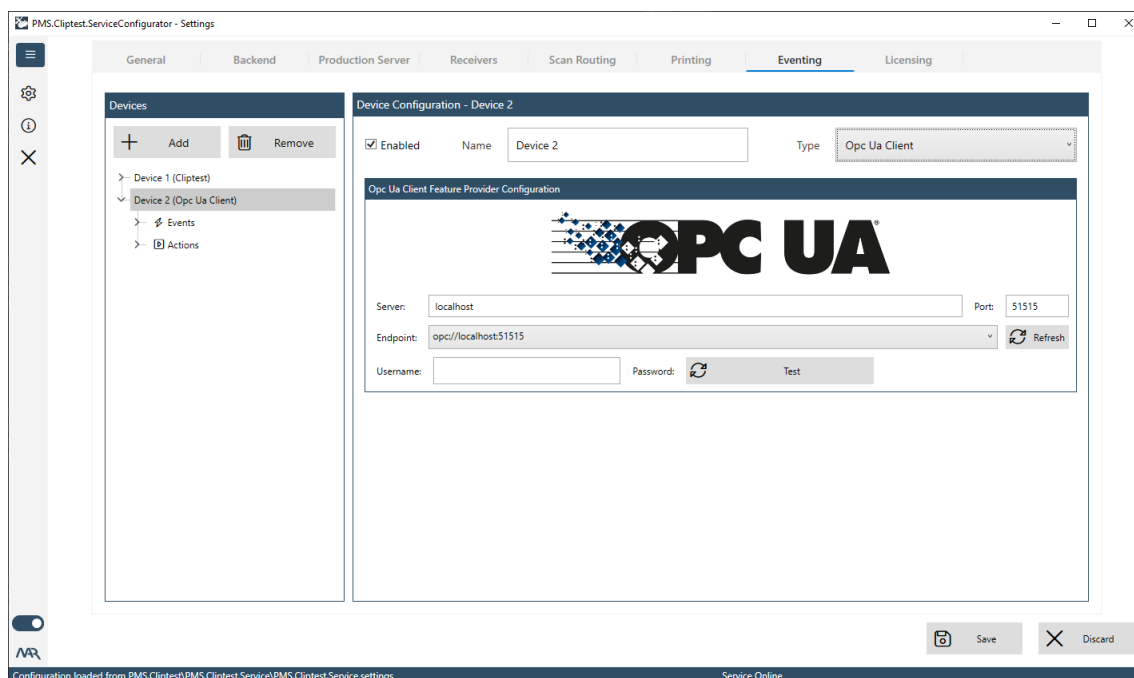
### 5.3.2.6 Printing

The **Printing** tab is used to manage the PMS.PrintProcessor instances. One can define a multiple instance and enable or disable each separately. The **PrintProcessor Name** and the **Host** and **Port** combination have to be unique in the list.

### 5.3.2.7 Eventing

A **PMS.Cliptest.Eventing** license is required to use this feature. The configuration of the **Eventing** is possible, but it will be not processed while **PMS.Cliptest** is running.

The **Eventing** feature enables to the user many possibilities to automatize workflows or manage interactions with other applications / devices. Bunches of possibilities are grouped into a device that has actions and events. All devices have a main configuration, which will be used to enable or disable a device, set up a name to identify it and the type of the device (all available device types are explained in chapter [6 PMS.Cliptest.Eventing](#)). Some devices have an additional specific configuration, e.g. connection information for a PLC, etc. Each action or event can have its own configuration, which will be used to raise the event or handle the action.



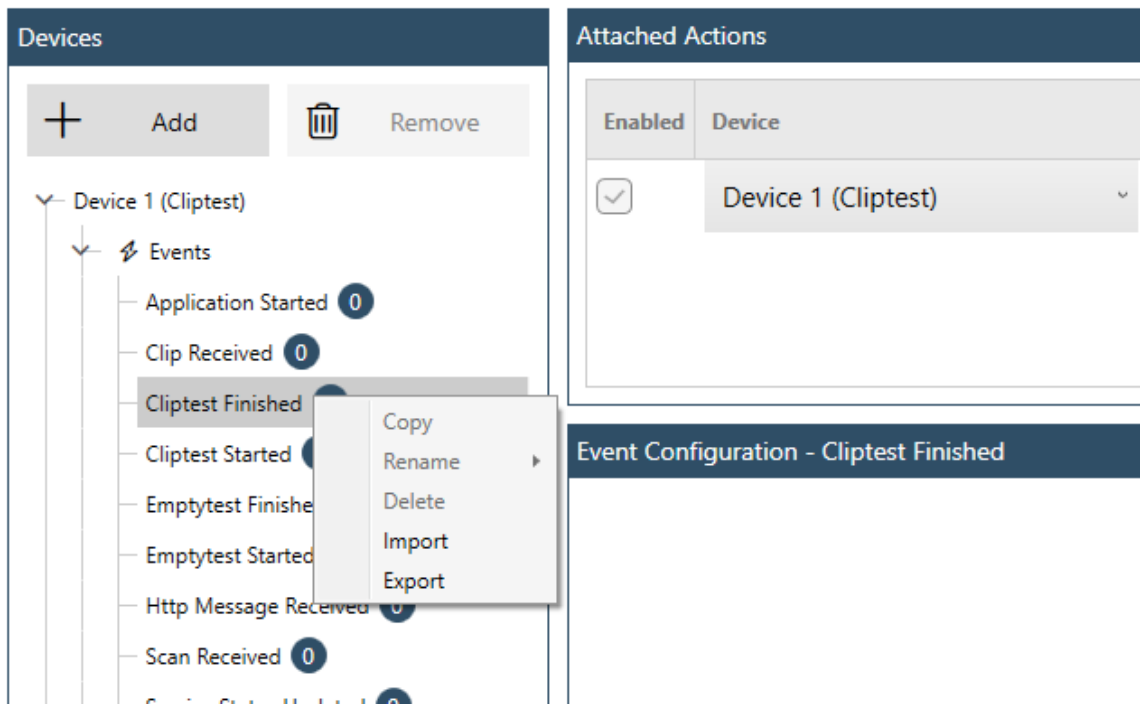
On the left side, the configured devices are displayed as a tree view. To manage the devices, the add- and remove-buttons must be used. The device configuration can be opened via selecting the device node in the tree view. A device node has up to two sub nodes, one for the implemented events and one for the possible actions. To access

the event or action configuration, the request node in the tree view must be selected. For further information about the configuration of actions and events, check the explanations for each type (see chapter [6 PMS.Cliptest.Eventing](#)).

Multiple actions can be executed asynchronously by each event. There is no guarantee that the actions of an event will be handled in the configured order. To attach an action to an event, the event node can be selected and the **Attached Actions** list used. The configuration of an attachment can be changed in the grid line. The device, which should handle the event can be selected and after that, the action of the device. It is not possible to handle each event with all actions caused by the specific data format from an event. The incompatible actions will be displayed in a gray style. If an incompatible action is attached to an event, the actions will be not executed and the issue will be logged.

5.3.2.7.1 DEVICE CONTEXT MENU

The device tree view has a context menu which enables several options to the user. The available options depend on the selected node in the tree view. Some of the events and actions can be duplicated, to have the same items with different configurations. For example it is possible to duplicate the Test Changed event to have single events, for changes in a cliptest and changes in an emptytest.



| Option | Description  |
|--------|--|
| Copy   | This button duplicates the selected item. This option is not available for all actions and events of a device.   |
| Rename | This button renames the selected item. This option is only available if the copy feature is enabled for an item.   |
| Delete | This button deletes the selected item. This option is only available if the copy feature is enabled for an item. The last remaining copy of an item cannot be deleted.   |
| Import | This button imports the eventing configuration from a created export. Therefore the existing configuration will be dropped and the data from the import will be applied. |

|        |   |
|--------|---|
| Export | This button creates an export of the complete eventing configuration. With this option the user can distribute the eventing configuration between different PMS.Cliptest instances or create backups. |
|--------|---|

#### 5.3.2.8 Licensing

The **Licensing** tab is used to manage the currently used licenses. It lists all active licenses and their types. If the license is stored on an USB dongle, the **Type** column shows a small icon. To add a new license, the license wizard can be started by clicking **Add New Licenses**.

### 5.4 PMS.Cliptest.Web

Starting with version 23.2.1.0 PMS.Cliptest brings it's own website. This is done by a self-hosting blazor server side web application. The default port is 45183 and can be configured in PMS.Cliptest.ServiceConfigurator. To open the web-interface the button "Open Website" in PMS.Cliptest.ServiceConfigurator can be used or by navigating in the browser to "http://[Hostname]:[Port]" where Hostname is the name of the machine where PMS.Cliptest.Service is running. On the local machine the url "<http://localhost:45183>" should work by default.

#### 5.4.1 Requirements

- PMS.Cliptest Version 23.2.1.0 or newer
- A modern Web-Browser. Internet Explorer is not supported. Microsoft Edge, Google Chrome, Firefox (> Version 88.0) and Safari are tested.
- In case of remote access - the configured port (default 45183) not being blocked by IT Security

#### 5.4.2 Options

In the top left corner the hamburger menu can be expanded to see the list of all available options. The list contains more elements if an administrator is signed in.

##### 5.4.2.1 Status

This is the home screen where all relevant modules of PMS.Cliptest are visualized to quickly find out if there are any problems. The dashboard shows all required parts of PMS.Cliptest. A red header color of any part indicates that there is a problem which needs to be solved. For example the license part will get red if the license dongle is detached from the machine.

##### 5.4.2.2 Logs

The logs component shows the latest log entries from newest to oldest. Only the latest 1000 entries are shown to quickly find out what causes a current problem. It is built of two tab-pages. The first tab page shows only warnings and errors where the second tab page shows all log entries.

##### 5.4.2.3 Reporting

There are build-in reports implemented. Each report can be accessed by a different tab page, then selecting the time-range and finally pressing the "Generate Report" button. It is required to have a PMS.Cliptest.Reporting license to be able to generate report data. Report data is stored in the PMS.Cliptest backend and is mandatory for reports to be generated.

##### 5.4.2.3.1 TEST STATISTIC

The test statistic report shows how many tests were performed on each board in the selected time-range. The data can be optionally filtered by workplace, test-type and order-type. The results can be exported to a csv file. The report shows how many tests passed and how many tests failed per board by stacked bars.

*This option is available with a PMS.Cliptest.Reporting license.*

##### 5.4.2.3.2 RANKING OF CLIP EVENTS

This report shows the number of clip triggers over all tests in the selected time range. The clips are ordered by the most frequent triggered clip per test to easily detect clips which are pressed unusually too much. The report can be used to detect physical problems with clips in production and furthermore to predict which clips might break sooner than others.

#### 5.4.2.4 Tests

The tests overview shows a list of current and recently (last 10) finished tests. Users can quickly see if tests are running and how their progress is. By using the "Details" button a popup-window appears with all details about the selected test. Information about clips, scans, critical events and additional notes can be reviewed. This might be helpful to select the correct bookmarks for label printing or to verify why any bookmark has no data.

#### 5.4.2.5 Grid Layout

*This is not tested in production environment yet. This view is still in beta experimental phase. Use it on your own risk. As this view contains input possibilities any actions can influence production processes.*

The Grid Layout page is able to control and visualize tests. There is an usb scanner input field to perform scans to start and stop tests. The test is visualized as a list of clips. The current login-name is used as the client-id to identify the viewclient to act for. When opening this page a subscription is done to the PMS.Cliptest.Service and therefore a corresponding license is blocked. The subscription is removed when leaving the page. If the same client-id is already connected this page can be used to mirror the visualization of the same test as in the PMS.Cliptest.ViewClient. Next to the scan input field there is a button which can be used to switch to (and back) from a full-screen view. This hides the header and the menu bar of the page. If the full-screen mode of the browser is activated as well this button enables the full-screen view of the test.

#### 5.4.2.6 Image Layout

*This is not tested in production environment yet. This view is still in beta experimental phase. Use it on your own risk. As this view contains input possibilities any actions can influence production processes.*

The Image Layout page is able to control and visualize tests. The behaviour is similar to the description for the Grid Layout except a different visualization. The image and the corresponding clip positions are presented which were configured in PMS.Cliptest.MasterdataManager.

#### 5.4.2.7 Printing

The printing page can be used to trigger label printouts. It is supposed to be used during the printing setup or to investigate any problems in the label design. After selecting the correct PrintProcessor at the configured IP and Port the printer and label needs to be selected. The printout is triggered by the "Print" button. A default automatically generated Test will be used as input for the label.

#### 5.4.2.8 Receiver

This page provides an overview about connected receivers and received signals. By pressing the "Start" button of a receiver the signals are update immediately on the website. It can be checked if signals are received like expected as well as if there are very much signals from neighbour lines which are not relevant for this PMS.Cliptest.Service and corresponding masterdata. If the amount of not interesting signals is very high it is recommended to consider to change the position of the receiver, for example more away from other lines and closer to the relevant line and boards.

#### 5.4.2.9 Eventing

The Eventing page provides an overview about the established connections. It is used to find problems in case of automations are not working like expected. Furthermore it can be used to see if there are connections which are triggered unexpected frequently which might decrease overall performance of the system. It can also help to see that there are connections which are not used and can be removed. It is required to use a PMS.Cliptest.Eventing license in order to see data on this page. Some of the eventing devices provide an additional user interface (for example OPC

UA Server) which can be opened by click on the providers name (if user interface is available the provider name becomes a button which opens a popup window).

#### 5.4.2.10 Counters

There are several counters inside PMS.Cliptest. This pages shows an overview about all counters and their values. Users with administrative priviledges have the possibility to reset each counter using the "Reset" button.

#### 5.4.2.11 Administration

The Administration page is used to manager users and groups. By default there is a build-in user named "root". At the first login a password for user "root" needs to be configured. The password for "root" can only be changed by user root and no other administrative users. Therefore ensure to not loose the password for user "root", keep it at a save place. More users can be added in this page by "Add User" button and existing users can be edited by "Edit" button. Users can be assigned to group "User" or "Admin". Users of group "Admin" can use advanced functions in PMS.Cliptest.Web which can influence production. Users of group "User" are not allowed to use any functions which can lead to production processes.

#### 5.4.2.12 Login/Logout

This option is used to login as a user or if already logged in to logout the current user.

## 6 PMS.Cliptest.Eventing

PMS.Cliptest.Eventing is a powerful feature to automate various tasks to help operators, enhance product quality and reduce testing time. It is based on logical or physical devices that provide information as events and consume information to perform an action. Available devices can be combined by attaching one or more actions to an event. Furthermore, each action can consider one or more conditions which need to be fulfilled to execute an action. Conditions are to be used to check any case and result in true or false. Only when all configured conditions result in true then this action will be executed

This feature is implemented as part of the **PMS.Cliptest.Service** and can be configured in **PMS.Cliptest.ServiceConfigurator**. It was introduced in version 20.2.1.0 and is continuously extended with more devices.

This section describes all currently available devices with their events and actions. The events have some example actions which can be triggered from the event. This list will be ongoing extended by more devices and can be easily added to existing installations.

### 6.1 Data Types

The following tables describes the available **Data Types**, which will be provided from the events. The description of an event contains the provided type and the description of an action contains the accepted types.

| Data Type        | Description  |
|------------------|--|
| None             | No specific data will be provided                          |
| Clip Status      | information about a clip with its address and status       |
| Test Information | All information about a test                               |
| Text             | Raw text data  |
| Scanner Message  | Scan data from a view client                               |
| All              | All Data Types are allowed and will be process if possible |

## 6.2 Cliptest

**PMS.Cliptest** acts as a device for the Eventing feature. The following chapter describes all events and actions of **PMS.Cliptest**

### 6.2.1 Events

#### 6.2.1.1 Application Started

|               |  |
|---------------|--|
|               |  |
| Trigger       | No specific data will be provided                    |
| Data Type     | information about a clip with its address and status |
| Configuration | All information about a test                         |

#### 6.2.1.2 Clip Received

|                 |  |
|-----------------|--|
|                 |  |
| Trigger         | No specific data will be provided  |
| Data Type       | information about a clip with its address and status   |
| Configuration   | filter options for <b>Clip Address</b> and <b>Clip State</b> for which clip the event should be raised |
| Example actions | - Start Cliptest on pressing a specific clip   |
|                 | - Start Emptytest on releasing a specific clip   |

#### 6.2.1.3 Cliptest Finished

|               |  |
|---------------|--|
|               |  |
| Trigger       | A Cliptest was finished (independent from the test result) |
| Data Type     | Test information   |
| Configuration | No specific configuration                                  |

#### 6.2.1.4 Cliptest Started

|               |                           |
|---------------|---------------------------|
|               |                           |
| Trigger       | A Cliptest was started    |
| Data Type     | Test information          |
| Configuration | No specific configuration |

#### 6.2.1.5 Emptytest Finished

|               |  |
|---------------|--|
|               |  |
| Trigger       | An Emptytest was finished (independent from the test result) |
| Data Type     | Test information   |
| Configuration | No specific configuration                                    |

#### 6.2.1.6 Emptytest Started

|         |                          |
|---------|--------------------------|
|         |                          |
| Trigger | An Emptytest was started |

|               |                           |
|---------------|---------------------------|
| Data Type     | Test information          |
| Configuration | No specific configuration |

#### 6.2.1.7 Http Message Received

|               |   |
|---------------|---|
| Trigger       | Clptest receives a message from the web interface |
| Data Type     | Text  |
| Configuration | No specific configuration                         |

#### 6.2.1.8 Scan Received

|               |  |
|---------------|--|
| Trigger       | A <b>ViewClient</b> sends data to the service                        |
| Data Type     | Scanner Message  |
| Configuration |  |
|               | - filter for a workplace which sends the scanned data (optional)     |
|               | - filter for the type of the scanner message                         |
|               | - filter for the scanned data with regex pattern matching (optional) |

#### 6.2.1.9 Service Status Updated

|               |   |
|---------------|---|
| Trigger       | Occurs when the status of a service is changed changed. |
| Data Type     | Service Status Message                                  |
| Configuration | No specific configuration                               |

#### 6.2.1.10 Test Changed

|               |   |
|---------------|---|
| Trigger       | Occurs when the state of a test changed.        |
| Data Type     | Test Information                                |
| Configuration |   |
|               | <b>Test Type:</b> optional test type filter     |
|               | <b>Test Status:</b> optional test status filter |
|               | <b>Test Result:</b> optional test result filter |
|               | <b>Workplace:</b> optional Workplace filter     |

### 6.2.2 Actions

#### 6.2.2.1 Change Clip Status

|  |  |
|--|--|
|  |  |
|--|--|



|                 |  |
|-----------------|--|
| Description     | Updates the status of a clip, e.g. triggered by a hardware device. Sets the clip status based on the configuration of the ClipStatus of the event. If there is no address configured and a scan message triggers the action, the value will be used for the address. |
| Data Types      | All  |
| Configuration   | filter options for <b>Clip Address</b> and <b>Clip State</b> for which clip the event should be raised   |
| Example actions | - Start Cliptest on pressing a specific clip   |
|                 | - Start Emptytest on releasing a specific clip   |

#### 6.2.2.2 Handle Scan

|               |   |
|---------------|---|
| Description   | Handles the provided data as a scan       |
| Data Types    | Text, Scanner Message                     |
| Configuration | This action has no specific configuration |

#### 6.2.2.3 Hide Message

|               |  |
|---------------|--|
| Description   | Hides displayed messages on the ViewClient |
| Data Types    | All  |
| Configuration | This action has no specific configuration  |

#### 6.2.2.4 Import Master Data

|               |  |
|---------------|--|
| Description   | Cliptest tries to start the import of the master data from the configured path.  |
| Data Types    | All  |
| Configuration |  |
|               | - Import folder where the master data files are located  |
|               | - Option to enable the Variants import (requires csv files with the header <b>#VariantName; KSKType; Comment; Parameters</b> ) |
|               | - Option to enable the Cliplist import (requires csv files with the header <b>#VariantName; Board; LED; Comment</b> )          |

#### 6.2.2.5 Log Event data

|               |  |
|---------------|--|
| Description   | Writes the provides event data to the eventing log file with log level <b>Info</b> |
| Data Types    | All  |
| Configuration | This action has no specific configuration  |

#### 6.2.2.6 Print

|  |  |
|--|--|
|  |  |
|--|--|

|               |  |
|---------------|--|
| Description   |  |
| Data Types    |  |
| Configuration |  |

#### 6.2.2.7 Request Service Status

|               |  |
|---------------|--|
|               |  |
| Description   |  |
| Data Types    |  |
| Configuration |  |

#### 6.2.2.8 Reset Counter

|               |  |
|---------------|--|
|               |  |
| Description   |  |
| Data Types    |  |
| Configuration |  |

#### 6.2.2.9 Show Message

|               |   |
|---------------|---|
|               |   |
| Description   | Shows a popup on all ViewClients with the provided data |
| Data Types    | Text or Scanner Message                                 |
| Configuration | This action has no specific configuration               |

#### 6.2.2.10 Start Cliptest

|               |   |
|---------------|---|
|               |   |
| Description   | Starts a Cliptest   |
| Data Types    | A Test or Clip Status   |
| Configuration |   |
|               | - The workplace (for the <b>ViewClient</b> ) to start the test for. Only required in case Data Type is Clip Status. |
|               | - The Variant to be tested.   |

#### 6.2.2.11 Start Emptytest

|               |   |
|---------------|---|
|               |   |
| Description   | Starts an Emptytest   |
| Data Types    | Test or Clip Status   |
| Configuration |   |
|               | - The workplace (for the <b>ViewClient</b> ) to start the test for. Only required in case Data Type is Clip Status. |

#### 6.2.2.12 Stop Test

|  |  |
|--|--|
|  |  |
|--|--|

|               |   |
|---------------|---|
| Description   | Stops a test  |
| Data Types    | Test or Clip Status   |
| Configuration |   |
|               | - The workplace (for the <b>ViewClient</b> ) to start the test for. If this value is configured, the event data will be not used. |

### 6.2.3 Conditions

#### 6.2.3.1 Is Test Running For Board

|               |  |
|---------------|--|
| Description   | Checks if a test is running for a provided board |
| Data Types    | Test, All (value from static configuration)      |
| Configuration |  |

#### 6.2.3.1 Is Test not Skipped

|               |   |
|---------------|---|
| Description   | Checks whether previous the test was skipped                |
| Data Types    | Text, ScannerMessage, All (value from static configuration) |
| Configuration |   |

## 6.3 Timer

### 6.3.1 Events

#### 6.3.1.1 Scheduler

|               |  |
|---------------|--|
| Trigger       | Triggered by configured schedule. Can be used to perform cyclic tasks at defined times like for example every day at midnight.   |
| Data Types    | None   |
| Configuration |  |
|               | - Interval is the time between events  |
|               | - "Start at" defines the date and time which is used as first occurrence of cyclic events. If the date or time is in the past, all further events will be relative to the start date and time. |

#### 6.3.1.2 Timer Elapsed

|               |  |
|---------------|--|
| Trigger       | Triggered by configured schedule. Can be used to perform cyclic tasks at defined times like for example every day at midnight. |
| Data Types    | None   |
| Configuration |  |
|               | - Configure the interval between the events of this timer  |

|  |   |
|--|---|
|  | - Check "Run continuously" if events should be raised cyclic, if unchecked the event is raised once |
|--|---|

### 6.3.2 Actions

#### 6.3.2.1 Start Timer

|               |   |
|---------------|---|
|               |   |
| Description   | Starts the timer                          |
| Data Types    | All                                       |
| Configuration | This action has no specific configuration |

#### 6.3.2.2 Stop Timer

|               |   |
|---------------|---|
|               |   |
| Description   | Stops the timer and therefore the occurrence of events of this device |
| Data Types    | All   |
| Configuration | This action has no specific configuration                             |

## 6.4 Audio Player

### 6.4.1 Actions

#### 6.4.1.1 Play File

|               |   |
|---------------|---|
|               |   |
| Description   | Play a wave file  |
| Required data | All   |
| Configuration |   |
|               | - Select your *.wav file which should be played                 |
|               | - Play immediately and test your output using the Play - Button |

#### 6.4.1.2 Play System Sound

|               |                            |
|---------------|----------------------------|
|               |                            |
| Description   | Plays random system sounds |
| Required data | All                        |
| Configuration | no specific configuration  |

#### 6.4.1.3 Text To Speech\*

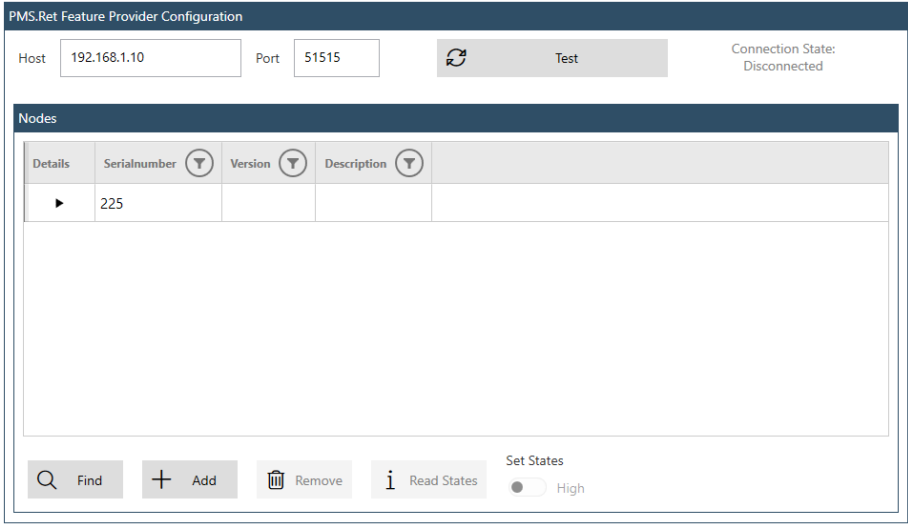
|               |   |
|---------------|---|
|               |   |
| Description   | Speaks a provided text using the default output device (windows default audio device) |
| Required data | Text, Scanner Message or All (configured static text will be used)                    |
| Configuration |   |

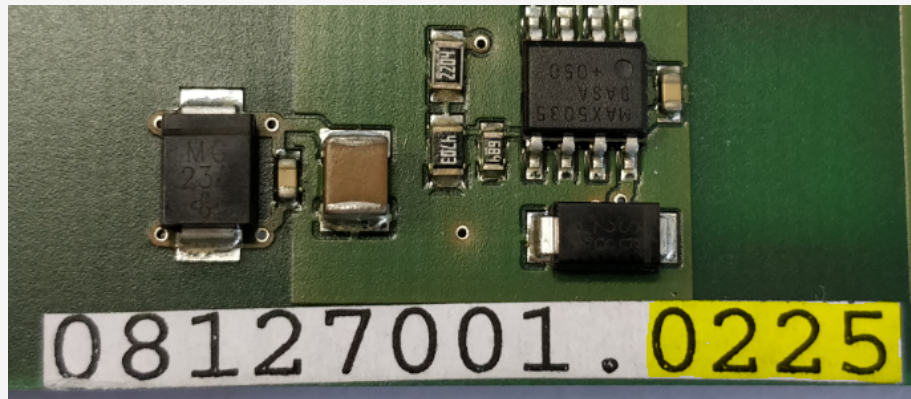
|  |   |
|--|---|
|  | - Text will be used when configured. Leave empty if the text is provided by the corresponding event |
|  | - Select the preferred voice (needs to be installed in Windows)                                     |
|  | - Use the Test – Button to try the output   |

### 6.5 PMS.Ret

The PMS.Ret device is to be used to connect hardware that is able to communicate over CAN bus. The implementation assumes that **PMS.ClipTest** is not connected to the CAN bus directly, instead an Ethernet-CAN Gateway is to be used.

The following configuration is required:

|               |  |
|---------------|--|
| Configuration |   |
|               | - Specify the IP address and port of the Ethernet-CAN gateway. Use the Test-Button to verify the connection to the gateway.  |
|               | - Use the "Find"-Button to scan the CAN bus for devices. One can also use Add and Remove buttons to specify nodes and their serial numbers. The "Read States" button can be used to read the current channel states and the firmware version of the selected node. |
|               | - The "Set States" toggle switch button is used to set all output channels to high or low signal level state.  |
|               | - You can find the serial numbers on the hardware (marked yellow):   |



- Configure the IO-Channels of each CAN bus device by pressing the down-arrow in the first column:

Nodes

Details
Serialnumber
Version
Description

225

Available Channels

| Channel | Is Input                 | Led | Invert                   | Current State | Description |
|---------|--------------------------|-----|--------------------------|---------------|-------------|
| 0       | <input type="checkbox"/> | 1   | <input type="checkbox"/> | Low           |             |
| 1       | <input type="checkbox"/> | 2   | <input type="checkbox"/> | Low           |             |
| 2       | <input type="checkbox"/> | 3   | <input type="checkbox"/> | Low           |             |
| 3       | <input type="checkbox"/> | 4   | <input type="checkbox"/> | Low           |             |

Find
Add
Remove
Read States
Set States
High

- Configure for each of the 16 IO channels if it is output or input, assign the corresponding Led number from PMS.Cliptest Masterdata and specify the Output-Signal level using the "Invert Output" option. The default is outputs are low active. The current state column shows the result of the "Read States" button.

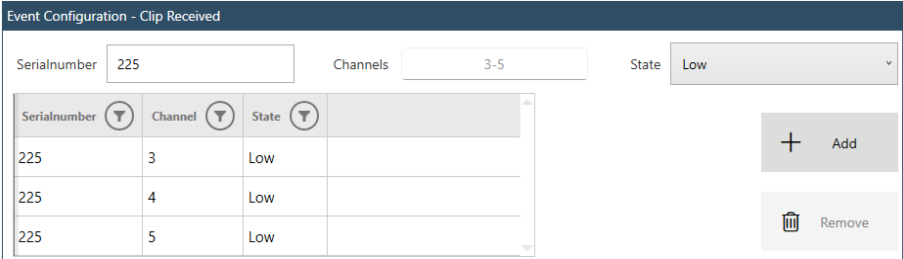
## 6.5.1 Events

### 6.5.1.1 Clip Received


|               |  |
|---------------|--|
| Trigger       | Changes of input signals will be treated as changed clips. The changed channel is being mapped to the led by PMS.Ret device configuration. |
| Data Type     | Clip Status  |
| Configuration |  |
|               | - Optional filter for input changes on serialnumber, specified channels and input state. Add and Delete buttons modify the filter table.   |
|               | - If no filter is used (default) all input changes from all nodes will raise this event  |

## 6.5.2 Actions

### 6.5.2.1 Set Output

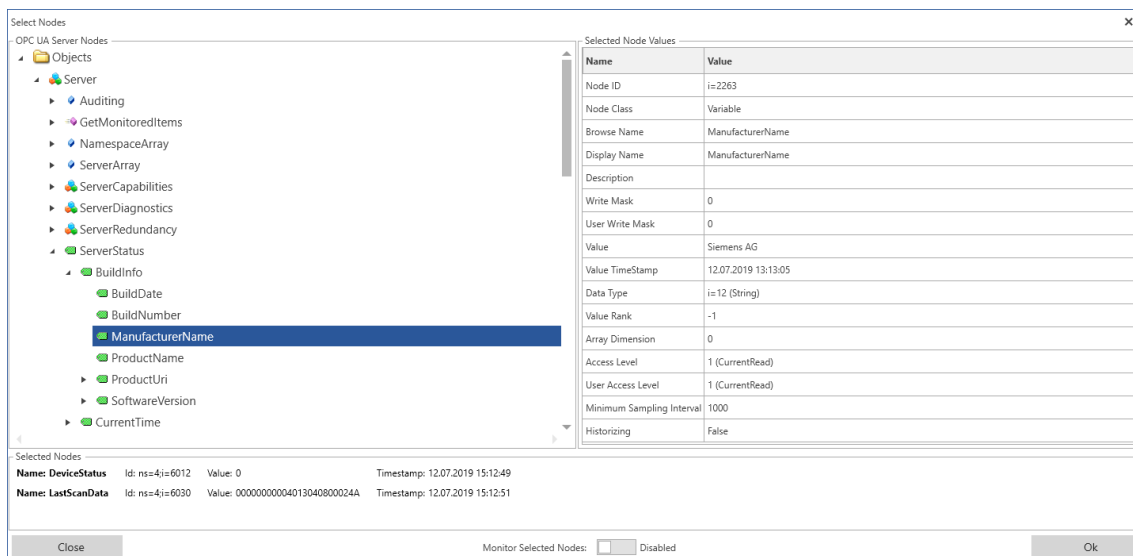
|               |  |
|---------------|--|
| Description   | Changes of input signals will be treated as changed clips. The changed channel is being mapped to the led by PMS.Ret device configuration. |
| Required Data | Test, Clip Status, Text, None (configuration is required)  |
| Configuration |    |
|               | - Optional definition of the nodes, channels and states to be set. Add and Delete buttons modify the filter table.                         |
|               | - If no filter is used (default) all output channels from all nodes will be changed corresponding to the connected event's datatype        |

## 6.6 Opc Ua Client

|               |   |
|---------------|---|
| Configuration |   |
|               | - Specify the IP address and port of the Opc Ua Server  |
|               | - Use the Refresh-Button to discover the endpoints of the configured server. If there are multiple entries for the same server, the entries have different authentication types and use different communication encryption. "#None#None" means no authentication and no encryption. |

### 6.6.1 Opc Ua Server Browser

In the configuration of the actions and events it is possible to specify Opc Ua Items. To reduce the risk of invalid item paths, the item can be selected via the browse button next to the item input fields.



The top left area "OPC UA Server Nodes" shows the structure of the server as a tree view. When an item is selected, it loads its sub-elements and expands automatically if there are sub items. There are different icons for different types of nodes.

The type and other details of the selected node are visualized on the top right area "Selected Node Values". This area contains information about the selected node like the data type, the current value and the access permissions. If the selected item is writeable, you can change the value directly in the corresponding cell, which contains the current value.

The area below "Selected Nodes" holds the list of already selected nodes. To select a node drag and drop a node from the "OPC UA Server Nodes" to this area "Selected Nodes". A row for the dropped item will be added to the list with its current value. To remove items from this list just select the item and press the delete-key on the keyboard. You can also just close the browser and open again to clear all selections. If you need to select only one node then the first node (top) of this list will be returned by this browser dialog.

The switch-button below "Monitor Selected Nodes" enables a live-view of the current values of all items, which are in the selected nodes list. It might be very helpful to verify node value changes already during the configuration of this eventing device.

## 6.6.2 Events

### 6.6.2.1 Item Changed

|               |   |
|---------------|---|
| Trigger       | Occurs when a subscribed item changed.  |
| Data Type     | Text  |
| Configuration |   |
|               | Item: Path to the OPC UA item which should be monitored.  |
|               | Value: optional filter for a specific value. Use regular expressions (see chapter <a href="#">12 Appendix III – RegEx Examples</a> for examples or help). |

### 6.6.2.2 Item Read

|  |  |
|--|--|
|  |  |
|--|--|



|               |  |
|---------------|--|
| Trigger       | Occurs when action Read Value completes.                                       |
| Data Type     | Text   |
| Configuration |  |
|               | Item: optional path to the OPC UA item for which the read action was executed. |
|               | Value: optional filter for a specific value.                                   |

### 6.6.3 Actions


#### 6.6.3.1 Read Value

|               |   |
|---------------|---|
|               |   |
| Description   | Reads a value to a specified item.                  |
| Required data | All (configuration will be used)                    |
| Configuration |   |
|               | Item: path to the OPC UA item which should be read. |

#### 6.6.3.2 Write Value

|               |   |
|---------------|---|
|               |   |
| Description   | Writes a value to a specified item.   |
| Required data | Text, ScannerMessage, All (configuration will be used)  |
| Configuration |   |
|               | Item: path to the OPC UA item which should be written.  |
|               | Value: optional field for writing a static value if the required data is not a Text or a ScannerMessage |

### 6.7 Brain Box

|               |  |
|---------------|--|
|               |  |
| Configuration |                            |
|               | - Specify the IP address and port of the brainbox. Use the Test-Button to verify the connection to the device. |

#### 6.7.1 Events

##### 6.7.1.1 Input Changed

|         |  |
|---------|--|
|         |  |
| Trigger | Change the signal level of the channels. |

|               |  |
|---------------|--|
| Data Type     | Clip Status  |
| Configuration |  |
|               | <b>Channel:</b> number of the channel (required information)   |
|               | <b>State:</b> Configures the condition when the event will be raised (input changed from LowToHigh, HighToLow or both) |

## 6.7.2 Actions

### 6.7.2.1 Pulse Output

|               |  |
|---------------|--|
| Description   | Writes an impulse to the output.   |
| Required data | All (configuration will be used)   |
| Configuration |  |
|               | <b>Channel:</b> number of the channel to which the value should be written |
|               | <b>Level:</b> Signal level of the impulse. (High or Low)                   |
|               | <b>Duration:</b> Duration of the impulse in milliseconds.                  |

### 6.7.2.2 Set Output

|               |   |
|---------------|---|
| Description   | Change the signal level of an output channel. The level can be inverted by configuration. |
| Required data | All (configuration will be used)  |
| Configuration |   |
|               | <b>Channel:</b> number of the channel to which the value should be written                |
|               | <b>State:</b> value which should be written to the output. (High, Low or Toggle)          |

## 6.8 HellermannTyton Cpk

The HellermannTyton Cpk eventing device connects cable tie pliers (Autotool 2000 CPK) to PMS.ClipTest over a serial port connection. It is used to test the correct amount of cable ties additional to clips. The cpk hybrid devices from HellermannTyton provide information about each cable tie immediately after it was applied to the product. The data contains information about a general result, possible error codes, used amount of current (tightness of cable tie) the timestamp and some more. This information can be saved into statistical data and also be printed. To combine the test of clips alongside with cable ties it is required to configure virtual clips in masterdata which have a special (configured) prefix as their address. If a cable tie was applied PMS.ClipTest looks for the next cable tie clip according to the address prefix which was not pressed yet. If there are no more cable tie clips left the event Binding Too Much will be raised which indicates a process error. The action Reset Binding Clips is used to reset all cable tie clips to NotPressed. A common scenario is to use action Set Binding Clips on every cliptest start to introduce the virtual cable tie clips to this eventing device. Action Reset Binding Clips can be called after the cliptest has finished for example to reset all virtual clips to NotPressed. Event Binding Clip Received is supposed to be connected to action Change Clip Status in the ClipTest eventing device to set the virtual clips to Pressed.

|  |  |
|--|--|
|  |  |
|--|--|

|               |  |
|---------------|--|
| Configuration | <div><div>HellermannTyton CPK Configuration</div><div><div><div>HellermannTyton</div></div><div><div>Communication</div><div>COM PortCOM1Baudrate57600ParityNoneStopbitsOneDatabits8</div></div><div><div>Processing</div><div><div><input checked="" type="checkbox"/> Check Binding Count</div><div>Clip Address PrefixCABLETIE</div></div></div><div><div>Test Configuration</div><div><div><div>Start Receiving</div><div>Stop Receiving</div><div>Send Command</div></div><div>Data</div></div></div></div></div>   |
|               | <p>- It is required to specify the COM port to which the hardware is connected. The default baudrate is 256000 when the usb port of the device is used. The option "Check Binding Count" is used to ensure the correct amount of cable ties. Can be deactivated on demand. The Clip Address Prefix needs to match the beginning of the addresses of the virtual clips in masterdata which represent the cable ties. The prefix is used to identify the cable tie clips and are then ordered by their address. The first cable tie will trigger the first virtual clip according to their sorted addresses. The configured prefix can be suffixed with numbers to ensure the correct press order according to the real process. Virtual addresses like "CABLETIE01", "CABLETIE02" are good examples. Finally there is a "Test Configuration" area where the communication can be tested. If the COM port is correct data should arrive in the data grid after each cable tie trigger. It is also possible to send commands to the device by the input field and the "Send Command" button. Valid commands are "VERN:" to receive the tools version information, "STAT:" to receive current state information or "ABBD:" to trigger an automated binding. Refer to the device documentation for the full list of commands.</p> |

6.8.1 Events

6.8.1.1 Binding Clip Received

|               |                                   |
|---------------|-----------------------------------|
| Trigger       | After each binding by the device. |
| Data Type     | Clip Status                       |
| Configuration | no specific configuration         |

6.8.1.2 Binding Faulted

|               |  |
|---------------|--|
| Trigger       | After a binding was done but with an error code. |
| Data Type     | Text   |
| Configuration | no specific configuration                        |

6.8.1.3 Binding Too Much

|  |  |
|--|--|
|  |  |
|--|--|

|               |  |
|---------------|--|
| Trigger       | A valid binding was received but all virtual binding clips were already triggered. |
| Data Type     | None   |
| Configuration | no specific configuration  |

#### 6.8.1.4 Data Received

|               |  |
|---------------|--|
| Trigger       | Is raised whenever data is received. After each binding as well as any other message from the device (after command was sent). |
| Data Type     | Text   |
| Configuration | no specific configuration  |

### 6.8.2 Actions

#### 6.8.2.1 Set Binding Clips

|               |  |
|---------------|--|
| Description   | Introduces the list of virtual binding clips to this event device. The list is required to enable the event Binding Clip Received. The list is extracted from all clips which address match the configured prefix (default is CABLETIE) and is sorted by the address |
| Required data | Test   |
| Configuration | no specific configuration  |

#### 6.8.2.2 Reset Binding Clips

|               |   |
|---------------|---|
| Description   | Used to reset the list of virtual binding clips to state NotPressed. Requires a previous call of action "Set Binding Clips" |
| Required data | All   |
| Configuration | no specific configuration   |

#### 6.8.2.3 Send Command

|               |  |
|---------------|--|
| Description   | Sends a command to the device. Event "Data Received" can be used to process the answer.                          |
| Required data | All (configuration will be used)   |
| Configuration | <div> <div>Action Configuration - Send Command</div> <div> <div>Command</div> <input type="text"/> </div> </div> |

|  |
|--|
| <b>Command:</b> the command to be sent. Valid commands are "VERN:" to receive the tools version information, "STAT:" to receive current state information or "ABBD:" to trigger an automated binding. Refer to the device documentation for the full list of commands. |
|--|

## 7 First Program Start

Before the first use of **PMS.Cliptest**, the **service** must be configured, the master data imported and the **ViewClient** configured.

### 7.1 Service Configuration

Before **PMS.Cliptest** can be used, licenses must be added with the ServiceConfigurator or USB dongles. For further information see 1.2 License Activation. **PMS.Cliptest** needs its own local database to store the imported master data, the test data and the current clip states. Configure the connection settings and test it. If no tables exist in the selected database, the **PMS.Cliptest.Service** creates the required tables at the first start.

The radio receivers for the clip events have to be directly connected to the Service. **PMS.Cliptest** enables one to connect multiple scanners to one **ViewClient**. The **ViewClient** transmits the scans to the Service and there the requested action is performed. To assign the scan from a source **ViewClient** to the destination **ViewClient**, a scan routing can be configured between the prefix of the scanner, the source **ViewClient** and the destination **ViewClient**. If only one scanner is connected to each **ViewClient** this setting is not relevant.

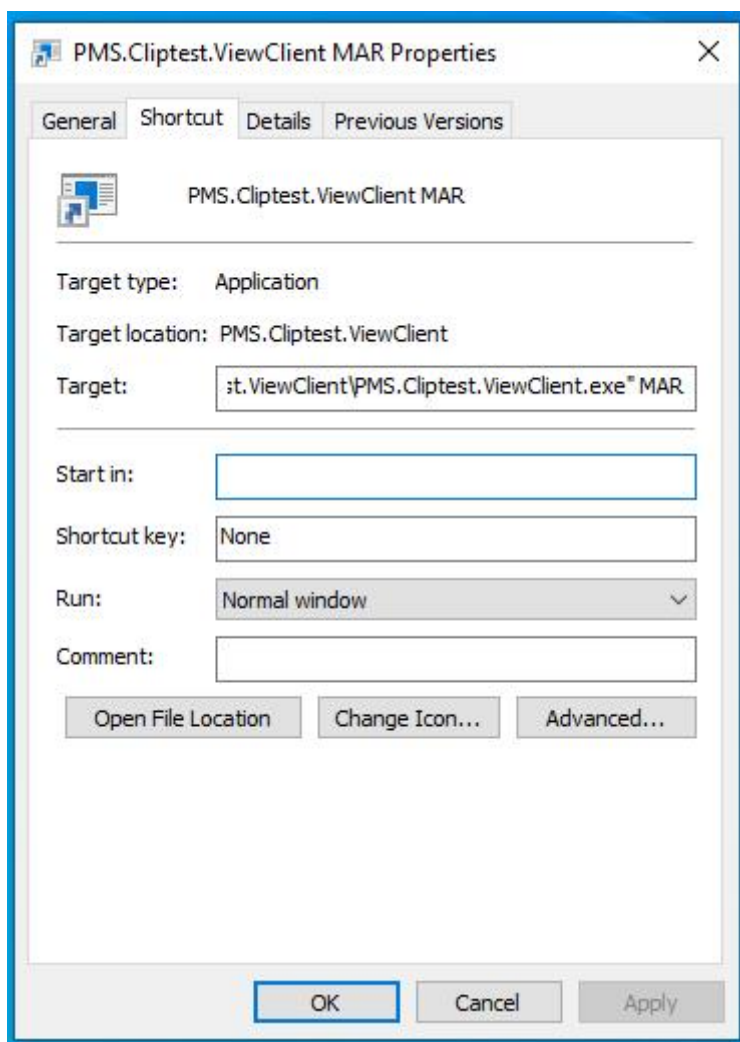
The **PMS.Cliptest.Service** is responsible for handling the defined print events. Print events can be configured in the **ViewClient**. On demand the **Service** transmits the print jobs to the **PMS.PrintProcessor.Service**. At least one **PMS.PrintProcessor.Service** must be configured for the printing feature to be used. By default the communication URI to PMS.PrintProcessor is **localhost** on port **55002** where localhost is the machine on which the PMS.PrintProcessor is installed.

### 7.2 Master Data Import

The first step is to configure the host and port to enable the communication with the **PMS.Cliptest.Service**. After that the paths for the csv files are set. Each file type needs its dedicated folder. If one does not use the default separators and the default comment token, they must be configured before starting the import.

### 7.3 ViewClient

Each **ViewClient** application has a unique id to differentiate between the clients. By default for this id, the machine name will be used. If starting two instances of the **ViewClient** application on the same computer is required, different client ids must be used. The client id with a startup parameter for the **ViewClient** application can be set up. A new shortcut can be added for the **ViewClient** application and open the properties. The id for the second client can be added on the tab page **Shortcut** add in the **Target** field. The id can be added after the last quote separated with a whitespace.



If different computers are used for the **Service** and the **ViewClient**, it is necessary to configure the host (and port) in the **ViewClient** settings. By default, the host is **localhost**. To change the host, the machine name or IP-address where the **PMS.ClipTest.Service** is running must be set. If the port of the service application must be changed, the same value must be configured. For further information, see chapter [5.1.3.1 General](#).

## 8 Troubleshooting

Common issues with possible solutions are described in this section. In the **ViewClient** application, one can get further information about the service status on the Overview page in the main window. The status of the connected clients, the backend, the production server and the receivers will be displayed. The latest log information of the service can also be seen.

| Application      | Issue                                   | Possible solution  |
|------------------|---|--|
| All applications | Save settings is not possible           | Check all tab pages for errors. If there is an error on any configuration section you will be not able to save the settings. |
| ViewClient       | <b>ViewClient</b> cannot connect to the | Check service logs of categories higher than warning   |

|                                 |   |   |
|---------------------------------|---|---|
|                                 | service                                   |   |
|                                 |   | Check the firewall settings. The required communication ports are described in chapter 0  |
|                                 |   | Check that the PMS.ClipTest has a valid license   |
| ViewClient                      | Test immediately finished with result NIO | Check service logs for the message "Clip (LED=XXX) is not configured for board (BoardTitle)"  |
|                                 |   | If the production server interface requests Clips which are not configured in the master data you will get these error. These Clips have the state UNCNF            |
| ViewClient<br>MasterDataManager | Demo Backend selected                     | Check the connection settings to the database in the ServiceConfigurator.   |
| Printing                        | Multiple pages printed                    | Check the printing condition in the template. If there is no printing condition is configured, for each entry of the list bookmarks a single page will be generated |

## 9 Appendix I - Printing Interface

The printing interface of **PMS.ClipTest** raises an Event on each change of internal process state (e.g. Empty test started or Clip test completed). If a corresponding event can be found in events configuration the specified label is printed on a printer via the **PMS.PrintProcessor.Service**. The following data fields (Key-Values) are provided with this interface and can be used in the TFORMer as bookmarks

The printing data contains two main parts. The first part is the general data, which provides information of a test. These data items are grouped by the TableName **General**. The second part is the list data. Each group of the list data has its own table name. The available groups with their corresponding bookmarks will be described in chapter [9.2 List Data](#). For further information about the printing interface, use the corresponding documentation.

### 9.1 General Data

#### 9.1.1 Test Data

| Bookmark  | Description  |
|-----------|--|
| TableName | Has the value General to identify this group   |
| Order     | Providing the name of the order / order type proceeded with the actual test (processed data)     |
| OrderType | Providing the type of the order / order type proceeded with the actual test                      |
| Board     | Providing the name of the assembly board on which the actual test was performed (processed data) |
| Steering  | Providing the Steering of the order / order type proceeded with the actual test                  |
| Workplace | Providing the internal configured name of the instance of PMS.ClipTest                           |
| ANLIE     | Providing the ANLIE of the order / order type proceeded with the actual test                     |

|                  |   |
|------------------|---|
| ProcessState     | Providing the process state of the actual test in clear text (e.g. Finished)              |
| TestType         | Providing the type of the actual test in clear text (ClipTest or EmptyTest)               |
| Result           | Providing the result of the actual test (e.g. IO or NIO)                                  |
| ResultBinary     | Providing the result of the actual test (e.g. 1 or 0)                                     |
| ID               | Providing a unique ID for the test with a format of a GUID                                |
| _NOW_            | Providing the current date and time   |
| CtYear           | Providing the current year (format YYYY)  |
| CtShortYear      | Providing the current year (format YY)  |
| CtMonth          | Providing the current month (value between 1 and 12)                                      |
| CtDay            | Providing the current day   |
| CtHour           | Providing the current hour (value between 0 and 23)                                       |
| CtMinute         | Providing the current minute  |
| CtSecond         | Providing the current second  |
| VariantComment   | Providing the Comment of the selected Order variant, which was configured in the Cliplist |
| VariantParameter | See chapter <a href="#">9.2.1 Variant Parameters</a>                                      |
| Operator         | Providing the actual logged in operator performing the actual printed test                |
| Scan             | See chapter <a href="#">9.2.2 Scans</a>   |
| StartTime        | Providing the timestamp of the test start (local time)                                    |
| FinishTime       | Providing the timestamp of the test finish (local time)                                   |
| ClipCountAll     | Providing the total number of clips   |
| ClipCountNio     | Providing the number of clips which are in the status NIO                                 |
| ClipCountIo      | Providing the number of clips which are in the status IO                                  |
| Login            | See Operator  |
| ElapsedTime      | Providing the duration since the test start   |
| RemainingTime    | Providing the remaining time until the test timeout                                       |
| Canceled         | Providing the status whether the test was canceled (true or false)                        |
| CanceledBinary   | Providing the status whether the test was canceled (1 or 0)                               |
| Skipped          | Providing the status whether the test was skipped (true or false)                         |
| SkippedBinary    | Providing the status whether the test was skipped (1 or 0)                                |

### 9.1.2 Scan Data



The Scan Data bookmarks can be used when there is no test but values of scans need to be referenced.

| Bookmark  | Description   |
|-----------|---|
| Order     | Contains the order scan after optional transform pattern was applied                            |
| OrderRaw  | The original value of the order scan without transform pattern                                  |
| Board     | The board scan after optional transform pattern was applied                                     |
| BoardRaw  | The original value of the board scan without transform pattern                                  |
| Workplace | The configured workplace of the ViewClient which has scanned                                    |
| Login     | The value of the login scan after optional transform pattern was applied                        |
| LoginRaw  | The original value of the login scan without transform pattern                                  |
| TestType  | Contains the test type (cliptest, emptytest) based on the scan sequence where the scan was done |
| Scan1     | Provides the value of the first custom scan - index is taken from order in scan sequence.       |
| Scan1Raw  | The value of the first custom scan without transform pattern applied                            |
| Scan2     | Provides the value of the second custom scan - index is taken from order in scan sequence.      |
| Scan2Raw  | The value of the second custom scan without transform pattern applied                           |
| Scan3     | Provides the value of the third custom scan - index is taken from order in scan sequence.       |
| Scan3Raw  | The value of the third custom scan without transform pattern applied                            |
| Scan4     | Provides the value of the fourth custom scan - index is taken from order in scan sequence.      |
| Scan4Raw  | The value of the fourth custom scan without transform pattern applied                           |
| Scan5     | Provides the value of the fifth custom scan - index is taken from order in scan sequence.       |
| Scan5Raw  | The value of the fifth custom scan without transform pattern applied                            |

### 9.1.3 Status Detail Data

The following bookarks can be used in Eventing when Action "Request Service Status" was executed. The corresponding event "Service Status Updated" provides status information where the following bookmarks can be used.

*Please note that the name of the client is appended to the bookmarks which begin with "Client" to be able to differentiate between the clients or workplaces. For client "MyWorkplace" the usage is for example:  
[ClientEmptyTestsFinishedIoMyWorkplace]*

| Bookmark         | Description                                     |
|------------------|---|
| ClientsCreated   | Number of internally created Clients            |
| ClientsConnected | Number of currently connected clients           |
| ServiceVersion   | Version information of the PMS.Cliptest.Service |

|  |  |
|--|--|
| ServiceUptime                              | Time passed since the PMS.Cliptest.Service was started                       |
| ServiceLicenseViewClients                  | Number of licensed/allowed ViewClients                                       |
| ServiceLicenseDemo                         | True if a Demo license is in use otherwise false                             |
| ServiceLicenseDemoValidUntil               | If Demo license is used, the Timestamp until the license is valid            |
| ServiceLicenseReporting                    | True if a Reporting license is found otherwise false                         |
| ServiceLicenseEventing                     | True if a Eventing license is found otherwise false                          |
| ServiceLicenseDashboards                   | True if a Dashboards license is found otherwise false                        |
| ServiceClipLicenseCount                    | Number of clips which can be handled in masterdata with current license      |
| ServiceLicenseService                      | True if a Service license is found otherwise false                           |
| ServiceLicenseMasterdataManagerExtensions  | True if a Masterdata Extensions license is found otherwise false             |
| ServiceLatestErrors                        | Last 10 errors which occurred in PMS.Cliptest.Service                        |
| ClientTestsFinished + ClientId             | Number of finished tests for this client                                     |
| ClientTestsFinishedIo + ClientId           | Number of finished tests with result IO for this client                      |
| ClientTestsFinishedNio + ClientId          | Number of finished tests with result NIO for this client                     |
| ClientEmptyTestsFinished + ClientId        | Number of finished emptytests for this client                                |
| ClientEmptyTestsFinishedIo + ClientId      | Number of finished emptytests with result IO for this client                 |
| ClientEmptyTestsFinishedNio + ClientId     | Number of finished emptytests with result IO for this client                 |
| ClientClipTestsFinished + ClientId         | Number of finished cliptests for this client                                 |
| ClientClipTestsFinishedIo + ClientId       | Number of finished cliptests with result IO for this client                  |
| ClientClipTestsFinishedNio + ClientId      | Number of finished cliptests with result NIO for this client                 |
| ClientUserTestsFinished + ClientId         | Number of finished tests for the current user and client                     |
| ClientUserTestsFinishedIo + ClientId       | Number of finished tests with result IO for the current user and client      |
| ClientUserTestsFinishedNio + ClientId      | Number of finished tests with result NIO for the current user and client     |
| ClientUserEmptyTestsFinished + ClientId    | Number of finished emptytests for the current user and client                |
| ClientUserEmptyTestsFinishedIo + ClientId  | Number of finished emptytests with result IO for the current user and client |
| ClientUserEmptyTestsFinishedNio + ClientId | Number of finished emptytests with result IO for the current user and client |

|   |   |
|---|---|
| ClientUserClipTestsFinished + ClientId        | Number of finished cliptests for the current user and client                                |
| ClientUserClipTestsFinishedIo + ClientId      | Number of finished cliptests with result IO for the current user and client                 |
| ClientUserClipTestsFinishedNio + ClientId     | Number of finished cliptests with result NIO for the current user and client                |
| ClientSessionTestsFinished + ClientId         | Number of finished tests since the ViewClient connected for this client                     |
| ClientSessionTestsFinishedIo + ClientId       | Number of finished tests since the ViewClient connected with result IO for this client      |
| ClientSessionTestsFinishedNio + ClientId      | Number of finished tests since the ViewClient connected with result NIO for this client     |
| ClientSessionEmptyTestsFinished + ClientId    | Number of finished emptytests since the ViewClient connected for this client                |
| ClientSessionEmptyTestsFinishedIo + ClientId  | Number of finished emptytests with result IO since the ViewClient connected for this client |
| ClientSessionEmptyTestsFinishedNio + ClientId | Number of finished emptytests with result IO since the ViewClient connected for this client |
| ClientSessionClipTestsFinished + ClientId     | Number of finished cliptests since the ViewClient connected for this client                 |
| ClientSessionClipTestsFinishedIo + ClientId   | Number of finished cliptests with result IO since the ViewClient connected for this client  |
| ClientSessionClipTestsFinishedNio + ClientId  | Number of finished cliptests with result NIO since the ViewClient connected for this client |
| ClientSessionCount + ClientId                 | Number of currently connected sessions with this client-Id                                  |
| ClientOperator + ClientId                     | Contains the name of the logged in operator at this client                                  |
| BackendMasterdata                             | Providing information about the existing masterdata in PMS.Cliptest.Service                 |
| BackendConnectionString                       | Provides information about the connection the the backend of PMS.Cliptest.Service           |
| BackendError                                  | If backend is in error state contains information about the error                           |
| Exception                                     | If there was an exception, the information can be retrieved with this bookmark              |
| ReceiverReceivingPorts                        | Names of currently used COM-Ports for working and connected receiver(s)                     |
| ReceiverFaultedPorts                          | Names of receiver COM-Ports which are configured but can't be opened                        |

|                                     |   |
|-------------------------------------|---|
| ProductionInterfaceClients          | List of Client-Ids which are present in the production interface    |
| ProductionInterfaceWorkplaces       | List of workplaces which are registered in the production interface |
| ProductionInterfaceInfo1            | Custom information of each production interface                     |
| ProductionInterfaceConnectionString | Currently used connection information in production interface       |

## 9.2 List Data

### 9.2.1 Variant Parameters

| Bookmark         | Description  |
|------------------|--|
| TableName        | Has the value <b>VariantParameters</b> to identify this group.   |
| VariantParameter | The parameters of a variant are accessible via a table. The data of each row is also available as a bookmark in the general data. e.g. <b>{VariantParameter3}</b> for the third parameter. |

### 9.2.2 Scans

| Bookmark  | Description   |
|-----------|---|
| TableName | Has the value Scans to identify this group.   |
| Scan      | The custom scans of a test are available via this table. The data of each row is also available as a bookmark in the general data. e.g. [Scan3] for the third scan. |

### 9.2.3 Critical Events

| Bookmark                     | Description   |
|------------------------------|---|
| TableName                    | Has the value <b>CriticalEvents</b> to identify this group. |
| CriticalEventMessage         | Automatic description of the event                          |
| CriticalEventComment         | Comment of the operator                                     |
| CriticalEventsAcknowledged   | Status of the acknowledgement                               |
| CriticalEventOperator        | Operator who has acknowledged the event                     |
| CriticalEventTimestamp       | Timestamp of event  |
| CriticalEventAckTimestamp    | Timestamp of acknowledgement                                |
| CriticalEventClipAddress     | Address of the clip which has triggered the event           |
| CriticalEventClipLED         | LED of the clip which has triggered the event               |
| CriticalEventClipDescription | Description of the clip which has triggered the event       |

|                      |   |
|----------------------|---|
| CriticalEventClipX   | X coordinate of the clip which has triggered the event      |
| CriticalEventClipY   | Y coordinate of the clip which has triggered the event      |
| CriticalEventTrigger | Trigger of the event (e.g. Timeout or Needless not pressed) |

#### 9.2.4 Clips

| Bookmark        | Description   |
|-----------------|---|
| TableName       | Has the value <b><i>Clips</i></b> to identify this group. |
| ClipLED         | LED number  |
| ClipState       | Final clip state e.g. IO, NIO, ...                        |
| ClipX           | X coordinate  |
| ClipY           | Y coordinate  |
| ClipAddress     | Clip address  |
| ClipPressCount  | Count of clip press events                                |
| ClipPressTime   | Timestamp of the last clip press event                    |
| ClipReleaseTime | Timestamp of the last clip release event                  |
| ClipDescription | Clip description  |
| ClipName        | Clip name   |
| ClipGroup       | Name of the group of the clip (optional)                  |

## 10 Appendix II - Available Clip States

There are seven clipstates available in **PMS.Cliptest**. They have different meanings depending what kind of test is running.

#### Clip Test

| Clipstate | Description   |
|-----------|---|
| IO        | This state is shown for all <b><i>pressed</i></b> clips which are requested in Cliplist table to be pressed for the order / order type and board.   |
| NIO       | This state is shown for all <b><i>unpressed</i></b> clips which are requested in Cliplist table to be pressed for the order / order type and board. |
| NLIO      | This state is shown for all <b><i>unpressed</i></b> clips which are NOT requested in Cliplist table.  |
| NLNIO     | This state is shown for all <b><i>pressed</i></b> clips which are NOT requested in Cliplist table.  |
| NLUKN     | This state is shown for all <b><i>unknown</i></b> clips which are NOT requested in Cliplist table.  |
| UNCNF     | This state is shown for all clips which are requested in Cliplist table to be pressed for the order /   |

|       |   |
|-------|---|
|       | order type and board but cannot be found in PinStatus table.  |
| GRPIO | If all clips of a clip group are in the state IO, the clips switches to this state and will not change to NIO on any changes during the test. |

## Empty Test

| Clipstate | Description   |
|-----------|---|
| IO        | This state is shown for all pressed clips which are requested in Cliplist table to be pressed for the order / order type and board.           |
| NIO       | This state is shown for all pressed clips defined for the requested Board in PinStatus table.   |
| NLIO      | This state is not relevant for an empty test.   |
| NLUKN     | This state is not relevant for an empty test.   |
| UNCNF     | This state is not relevant for an empty test.   |
| GRPIO     | If all clips of a clip group are in the state IO, the clips switches to this state and will not change to NIO on any changes during the test. |

## 11 Appendix III – Localization

Xliff files (version 2.0) are used for the localization in **PMS.Cliptest**. Xliff is an XML-based standardized format for the localization process. The files are located in the directory C:\ProgramData\MAR

GmbH\PMS.Cliptest\Localization\Translations. A separate file is provided for each application and language. On the application start, a file with the default language will be generated. If the application has new elements which are not in the existing files, the missing elements will be added with the default language in each language file.

### 11.1 Adding a new language

If a new language should be added, the directory with the translations must be opened and a copy of the default language file created. The file must be renamed to the new language, the translations updated and the target language updated during the update process. After an application restart, the new language will be available.

### 11.2 Xliff tools

The Language **PMS.Cliptest** files in can be edited with tools, which supports the xliff standard in version 2.0.

## 12 Appendix IIII – RegEx Examples

At several steps during configuration there are options to define regular expressions to filter, extract or detect values. The following table contains useful examples of commonly used expressions.

| RegEx          | meaning   |
|----------------|---|
| (?<=^..)*\$    | Skip the first two characters or digits               |
| ([0-9]{10})\w+ | Extract one or more numbers with minimum of 10 digits |

If there is no matching example the following online-tool is very powerful and explains the usage of regular expressions: <https://regexr.com/>.

The **ViewClient** has an integrated help dialog to test regular expressions. Use the "?"-button to open the help dialog:

Regular expressions are used to check data against a specified pattern. Important features:

- ^ Use this option to configure how the scanned data has to start. For example use the pattern '^ (BR)' to configure that the message has to start with 'BR'.
- \$ Use this option to configure how the scanned data has to end. For example use the pattern '(BR)\$' to configure that the message has to end with 'BR'.
- (, ) Use this option to specify how often the item has to occur. For example use the pattern '[3]' to ensure that the item occurs three times. You can also use '[2,5]' to specify a range (minimum 2 and maximum 5).
- + Use this option to mark an item as required. For example use '(ABC)+' for a message which has to contain at least one time the text 'ABC'
- [ ] Use this option to specify the allowed characters. For example use '[A-Za-z]' to allow only small and big letters. You can also use '[A|B|C]' to allow only the characters 'A', 'B' or 'C'.

Pattern Test

Match pattern

(? <= ^, ., \*)\$

Replace pattern

Input

LEBR001

Result

BR001

OK